Goal Babbling for an Efficient Bootstrapping of Inverse Models in High Dimensions

by
Matthias Rolf

Dissertation

Faculty of Technology
Bielefeld University

Bielefeld, August 2012

Printed on permanent paper according to ♾ ISO 9706.
Gedruckt auf alterungsbeständigem Papier ♾ ISO 9706.

Abstract

Learning to coordinate high-dimensional motor systems is a fundamental task for humans as well as robots. Traditional approaches to the computational learning of coordination skills rely on an exhaustive exploration of possible actions, which is not feasible in high dimensions. This thesis investigates *reaching* as a prototypical coordination problem and introduces the concept of *goal babbling* for the learning of reaching skills in high-dimensional domains. Goal babbling is inspired by studies about infant development that show that already newborns attempt goal-directed movements, even if they can not perform them successfully. This thesis develops methods that bootstrap reaching skills by mimicking such early goal-directed movements, and demonstrates their success in high dimensions.

The methods developed in this thesis implement goal babbling for the learning of *inverse models* as a direct mean to solve coordination problems. Theoretical results show how such inverse models can be learned by means of goal babbling. This thesis introduces the first algorithm that can learn inverse models by fitting observed data even when the coordination problem contains solution sets that are not convex, which has been a severe limitation of previous algorithms. It is shown that the approach allows for a bootstrapping that scales almost constantly with respect to the dimension of the action space, which is opposed to the exponential cost of exhaustive exploration. Experiments demonstrate that goal babbling constitutes a positive feedback loop between exploration and learning during the initial bootstrapping of skills. In an online learning scenario this is shown to permit substantial speedups of learning and to allow for human-level learning speed. Reaching with a bionic robot trunk is investigated as a practical scenario that is very hard to solve without learning due to the lack of analytical models and non-stationary system behavior. Extensive real-world experiments demonstrate the practical feasibility and usefulness of the goal babbling approach on this challenging platform.

Contents

List of Figures

Chapter 1

Introduction

> "From the motor chauvinist's point of view the entire purpose of the human brain is to produce movement. Movement is the only way we have of interacting with the world. All communication, including speech, sign language, gestures and writing, is mediated via the motor system. All sensory and cognitive processes may be viewed as inputs that determine future motor outputs." [Wolpert et al., 2001]

1.1 Motivation

The human body possesses more than 600 skeletal muscles [Welsh and Llins, 1997]. Performing purposeful actions to achieve some behavioral goal requires a high degree of coordination of these many degrees of freedom. Yet, human infants are born without the most basic coordination skills like reaching for an object [Konczak et al., 1997], which poses the *learning* of sensorimotor coordination as a fundamental problem in human development. The ability to learn sensorimotor coordination from scratch also allows to master the change induced by varying environments or body growth, and to learn more complex tasks like writing or riding a bicycle [Wolpert et al., 2001]. Understanding this ability to learn, and utilizing it for modern robotics systems is one of the major goals of the research fields of *cognitive* [Kopp and Steil, 2011] and *developmental robotics* [Lungarella et al., 2003, Asada et al., 2009].

This thesis investigates the learning of reaching skills, as an exemplary coordination skill, from a perspective of robotics and machine learning. The problem of reaching is to find motor commands (e.g. joint angles of a robot arm) that move the robot's end-effector (e.g. the gripper) towards some desired position in space. This problem setup is not only illustrative, but very *prototypical* for other problems of sensorimotor coordination: it asks the very general question of *how* to achieve some behavioral goals by means of actions. The skill of reaching itself is also *fundamental* for both robots and humans, since the positioning in space is necessary for any use of the robot's gripper or the human's hand. Already standard robots with well known geometry and mass distribution largely benefit from learning for the purpose of accurate and agile motor coordination [Nguyen-Tuong and Peters, 2011]. Learning is even more important for new generations of robots that combine mechanical flexibility, elastic material, and lightweight actuation like pneumatics. Such robots are often inspired by biological actuators like octopus arms [Laschi et al., 2009], elephant trunks [Korane, 2010] (see

Figure 1.1: The *Bionic Handling Assistant* mimics an elephant trunk.

figure 1.1), or human biomechanics [Hosoda et al., 2012], and provide enormous potential for the physical interaction between the robot and the world, and in particular between robots and humans. The downside of their biologically inspired design is that analytic models for their control are hardly available and difficult to design. This qualifies learning as an essential tool for their successful application.

Successful reaching skills can be well understood with the notion of *internal models* [Wolpert et al., 1998], whereas forward models predict the outcome of an action and inverse models suggest actions in order to achieve a desired outcome. The bootstrapping of internal models without explicit prior-knowledge requires experience that has to be generated by *exploration*. Machine learning approaches thereby traditionally rely on an exhaustive exploration of all possible motor commands, frequently generated by means of an entire random procedure, which is referred to as "motor babbling" [Bullock et al., 1993, Demiris and Dearden, 2005]. After the data generation phase, learning and coordination can be phrased in a variety of ways [D'Souza et al., 2001, Sun and Scassellati, 2005, Reinhart and Steil, 2011]. Yet, exhaustive exploration can not be achieved on high-dimensional motor systems such as modern humanoid robots, bionic elephant trunks, or the human body. The sheer number of combinations of commands for different actuators is too large to be explored in the lifetime of any learning agent. Successful application of exploration and learning on robots like the Bionic Handling Assistant demands approaches that yield useful results even without fully exploring the space of possible motor commands. Therefore,

> *the overarching goal of this thesis is to develop concepts and methods that allow for an efficient bootstrapping of reaching skills in high dimensions.*

The central inspiration to solve this challenge comes from studies on infant development. Infants display an enormous efficiency when bootstrapping their repertoire of

sensorimotor skills: they display rudimentary reaching skills already four months after birth [Thelen et al., 1996], which are successively refined during the first year of life. Although this does not involve all of the more than 600 muscles, it does at least involve arm, head [Thelen and Spencer, 1998], and torso [Rochat, 1992], which is still clearly too high-dimensional to be fully explored within four months of life. Mimicking this efficient bootstrapping requires insight into the exploratory movements performed by infants. The methods in this thesis draw inspiration from [von Hofsten, 1982] in order to organize exploration in an efficient manner. Von Hofsten showed that already newborns attempt goal-directed reaching movements, even if they can not perform them successfully. This thesis investigates such early goal-directed actions as mechanism for exploratory bootstrapping of coordination skills. Therefore the new concept of "goal babbling" is introduced, implemented, analyzed, and used to solve the coordination of the Bionic Handling Assistant.

1.2 Outline

Chapter 2 provides a general formalism of coordination problems (2.1) and basic terminology of how such problems can be solved with *internal models* (2.2). The sections 2.3 and 2.4 discuss standard methods of *learning* such internal models.

Chapter 3 introduces the approach and methodology of this thesis. Existing approaches for the learning of internal models are reviewed and criticized in the light of infant developmental studies. The concept of *goal babbling* is introduced as an approach to exploratory learning based on early goal-directed movements. The chapter discusses which particular methods should be used to study this concept and lays out distinct and detailed research goals for this thesis.

Following these goals, the chapters 4 and 5 investigate how inverse models can be learned by means of goal babbling. Chapter 4 presents a theoretical analysis of learning in purely linear domains, which shows the necessity of exploratory noise (as opposed to previous approaches [Sanger, 2004]) in order to find suitable actions for the behavioral goals. Chapter 5 investigates learning in non-linear domains in which learning from arbitrary exploratory data can fail due to inconsistent solutions [Jordan and Rumelhart, 1992], and shows how goal-directed exploration allows to solve this problem. The experiments provided in this chapter show first evidence for the efficiency of the approach to scale to high-dimensional systems.

The chapters 6 and 7 consider the *practicability* of the approach in real-world scenarios. Chapter 6 concerns the absolute *speed* of learning by investigating the dynamics of online-learning during goal babbling. Empirical results show that reaching skills can be bootstrapped within few hundred movements even in high-dimensional domains, which is feasible on a real robot and competitive with human learning [Sailer et al., 2005]. Finally, chapter 7 demonstrates the practical use of the developed method to learn reaching with the *Bionic Handling Assistant* (see figure 1.1).

Chapter 8 summarizes the findings on a conceptual and methodological level and discusses current limitations, as well as newly emerging research questions.

Chapter 2

Autonomous Learning of Coordination Skills

This chapter introduces the basic problem formulation of coordination that is used in this thesis and gives an overview of standard methods to solve it by means of learning.

2.1 The Coordination Problem

The present work considers an agent that can execute actions $q \in \mathbf{Q}$, where $\mathbf{Q}$ is the *action space* that subsumes all possible actions of the agent. Each action causes an outcome $x \in \mathbf{X}$ in some *observation space*. The unique causal relation between both variables is formally defined by some forward function f that describes the functioning of the agent's body or generally the world in which the agent is behaving:

$$f : \mathbf{Q} \to \mathbf{X}, \quad f(q) = x \tag{2.1}$$

Thereby actions q and observations x are considered to be multi-dimensional variables in a continuous space:

$$\mathbf{Q} \subseteq \mathbb{R}^m, \quad \mathbf{X} \subseteq \mathbb{R}^n \tag{2.2}$$

The dimension of the actions m is usually referred to as the number of *degrees of freedom* (DOF), and n is the dimension of the task. A general assumption is that *any* outcome $x \in \mathbf{X}$ can be achieved by some action $q \in \mathbf{Q}$, which can be formulated for any f and $\mathbf{Q}$ by defining $\mathbf{X}$ as the image of f: $\mathbf{X} = f(\mathbf{Q})$. Hence, f must be a surjective function with $n \leq m$. Furthermore, f is assumed to be continuous throughout this thesis, which is realistic for reaching problems and other examples discussed in this section. This very general relation of an action and observation space is illustrated in figure 2.1. If there are multiple actions $q_1 \neq q_2$ that cause the same outcome $f(q_1) = f(q_2)$ this is referred to as *redundancy*: the domain provides more different actions than necessary to achieve any possible outcome. Domains with $n < m$ generally have outcomes x that can be achieved by an infinite number of different actions. However, also domains with $n = m$ can contain a discrete number of solutions. The scope of this thesis are domains in which n is rather low-dimensional (such as $n = 3$), but m can be very high-dimensional, which is realistic in many real-world scenarios.

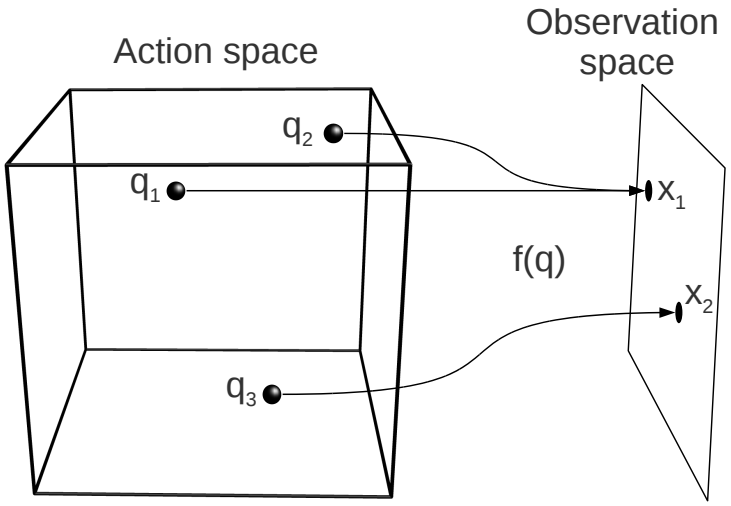


Figure 2.1: Action and observation space are connected by a forward function f that maps actions to their causal outcome. Since the action space is considered to have at least the dimension of the observation space, several actions can have the same outcome.

The *coordination problem* arises when the agent is asked to cause some desired outcome, or *goal*[1] $x^* \in \mathbf{X}^*$ out of a set $\mathbf{X}^* \subseteq \mathbf{X}$. The agent can not cause that outcome directly, but it has to estimate an appropriate action $\hat{q}$ that results in the observation of x^*, such that $f(\hat{q}) = x^*$. Hence, it has to know *how* to achieve a goal. In the most abstract way, the agent's *skill* to solve that problem for all goals in $\mathbf{X}^*$ can be denoted by some mechanism Ω that receives a goal as input and returns an appropriate action. This mechanism is not necessarily a mathematical function of x^*, but may have an internal state τ. The agent solves the coordination problem when the actions suggested by Ω always lead to the observation of x^*:

$$f(\Omega(x^*, \tau)) = x^* \quad \forall\, x^* \in \mathbf{X}^* \quad \forall\, \tau\ . \tag{2.3}$$

For the learning of such coordination skills, the agent does not know the underlying forward function f. The only elementary mechanism to probe knowledge is to query the forward function by choosing some exploratory action q, performing it and observing the outcome x. It is generally not possible to probe the reverse direction: there is no direct way to probe a correct solution q^* that solves a given goal x^*. The sections 2.2 to 2.4 provide an overview of standard approaches to the learning of coordination skills Ω based on this problem formulation.

[1] Throughout this thesis the star (*) of variables or sets indicates some *desired* state. Corresponding variables without star indicate possible or actual states. In contrast, the hat notion ($\hat{}$) refers to variables that are *estimated values* of something.

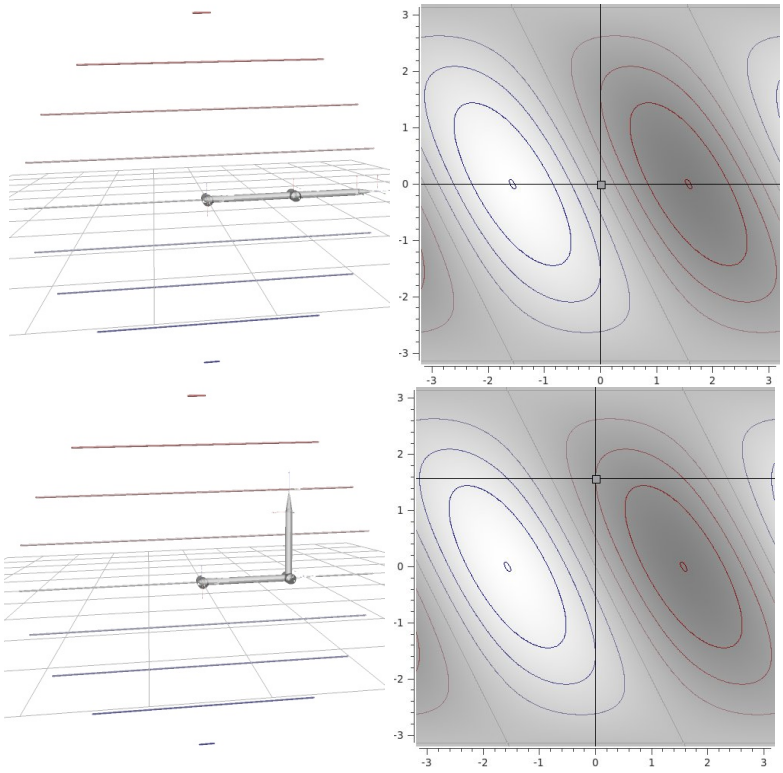


Figure 2.2: A robot arm with two joints shown in a stretched (top) and bent (bottom) posture. The left side shows the arms configuration, while the right side illustrates the action (joint) space. The marker in the joint space shows the current posture. Coordinating the height x of the effector can be done with a variety of joint angles q. Blue and red contours show redundancy manifolds, i.e. actions q that lead to the same observation x, as indicated by correspondingly colored lines on the left side.

Examples

A minimal example for a coordination problem with redundancy is shown in figure 2.2: a robot arm with two revolute joints ($m = 2$) shall be used for reaching. For a minimal scenario, the reaching only concerns the height of the effector ($n = 1$). Hence, the action space $\mathbf{Q}$ (here also "joint-space") comprises all possible combinations to position the joints inside $\mathbf{Q} = [-\pi; \pi]^2$. Assuming that each of the two links of the robot has a length of $0.5m$, the observation space of all possible effector heights is $\mathbf{X} = [-1m; 1m]$. Left and right movements of the effector are ignored in this minimal example. The redundancy appears in form of manifolds through the 2-DOF joint space, on which all joint angles apply the same effector height. Some of these manifolds are visualized by colored contours (see figure 2.2, right). The geometry of the arm defines the forward

function $f(q)$ as

$$f(q) = 0.5 \cdot sin(q_{(1)}) + 0.5 \cdot sin(q_{(1)} + q_{(2)}), \tag{2.4}$$

where $q_{(1)}$ and $q_{(2)}$ are the first and second component of q. The coordination problem here is to find and select joint angles q that causes the observation of some desired effector height x^*. Since q and x follow a geometric relation, this scenario is also known as *kinematics*. The experiments in this thesis follow this basic setup. However, realistic scenarios comprise a substantially larger number of degrees of freedom. Also, they comprise more task dimensions than $n = 1$, but usually not more than three (spatial position of the effector) or six (including the 3D orientation of the effector). An instance of kinematics that is not based on joint-angles, but on the length of various effectors, is investigated in chapter 7.

Another coordination problem that has been formulated in this way is *quadruped walking*: Baranes *et al.* set up a walking mechanism by defining some parametrized movement pattern that moves the four legs in a rhythmic manner [Baranes and Oudeyer, 2011]. The $m = 24$ parameters are *interpreted* as actions q. The causal outcome when applying such a movement for a certain time is that the robot has moved to some position (u, v) with orientation ϕ on the floor, which gives $x = (u, v, \phi)$ with $n = 3$. Hence, the forward function is a conjunction of the movement model that converts parameters to motor movement, and the physical way the movements interact with each other and with the ground. The coordination problem is to find appropriate movement parameters q whenever the robot is asked to move to some desired position x^*.

Hypothetical examples along these lines go much further: for instance Sanger discussed learning how to *play golf* [Sanger, 2004]. Like walking, this concerns dynamic movements, which could be encoded by a fixed-length sequence of intermediate steps or with a generic parametrization. An outcome could be the two-dimensional stopping-position of the golf ball on the green.

Wu *et al.* investigated the generation of *facial expressions* on a robot head, such that the expression is perceived to display various emotions [Wu et al., 2009]. Actions in that case are positions for $m = 27$ servo-motors that move an elastic skin on the robot's face. The resulting facial expression is evaluated by measuring $n = 12$ "facial action units" [Ekman and Friesen, 1978] that describe features of various emotional expressions. The forward function in this domain is the deformation physics of the skin, plus someone's perception (or an artificial recognition) of its expression. The coordination problem is to provoke some desired expression by means of motor commands.

Related Problem Formulations

The formulation of the world behavior as described in equation (2.1) neglects cases in which the outcome x of an action q depends on some *state* s of the world or the own body. For instance the interplay of forces or torques with the movement of rigid-

body systems can only be described in a state-dependent manner. In that scenario a forward function could, for instance, describe the acceleration of the body caused by some joint-torque, based on the state comprising current geometry and velocities of the different body-parts [Featherstone and Orin, 2007]. Coordination problems then arise when a torque is needed that results in some desired acceleration [Peters and Schaal, 2007, 2008]. A state-dependency is synthetically introduced in *instantaneous* kinematics formulations for robot control [Waldron and Schmiedeler, 2007, D'Souza et al., 2001]. This formulation investigates the relation between derivatives $\dot{q}$ and $\dot{x}$ in kinematic domains. The relation of these variables depends on the state $s = q$. This state-dependency is straight-forward to derive mathematically, and yet synthetical because it is based on the existence of a state-less forward function f as described in equation (2.1).

State-dependent coordination problems are a clear escalation of the state-less problem investigated in this thesis. In reverse, however, the state-less scenario is fundamental to the understanding of coordination problems with state, and exposes substantial challenges as described in the remainder of this thesis.

Another problem domain that is concerned with the choice of actions is *reinforcement learning*, in which the world's feedback to the agent does not consist of a multi-dimensional result x, but a scalar reward that needs to be maximized [Sutton and Barto, 1998]. Typical setups do not contain multiple goals, but one desired behavior that is encoded in a reward function (e.g. [Theodorou et al., 2010]). The *contextual bandits* problem [Langford and Zhang, 2008] is a sub-problem in reinforcement learning that has a similar structure to the coordination problem investigated in this thesis. In such a problem, an agent receives some context, which can be interpreted as a goal. The agent chooses an action and receives a reward based on context and action.

The crucial difference between the coordination problem in this thesis and the general reinforcement learning scenario is the rich feedback of an outcome x versus the sparse feedback of a reward. While rewards certainly describe the more general setup, many problems *do* provide a rich outcome-feedback, such as the examples discussed in this section.

2.2 Internal Models for Coordination

The mastery of coordination skills as defined in the last section can be well understood with the notion of *internal models* [Wolpert et al., 1998]. Internal models are functions that are available to the agent and describe relations between actions and their outcomes. A *forward model* $\hat{f}$ approximates the world's forward function f and predicts the outcome of an action:

$$\hat{f}(q) = \hat{x} \ . \tag{2.5}$$

An *inverse model* g suggests an action necessary to achieve a desired outcome

$$g(x^*) = \hat{q} \ . \tag{2.6}$$

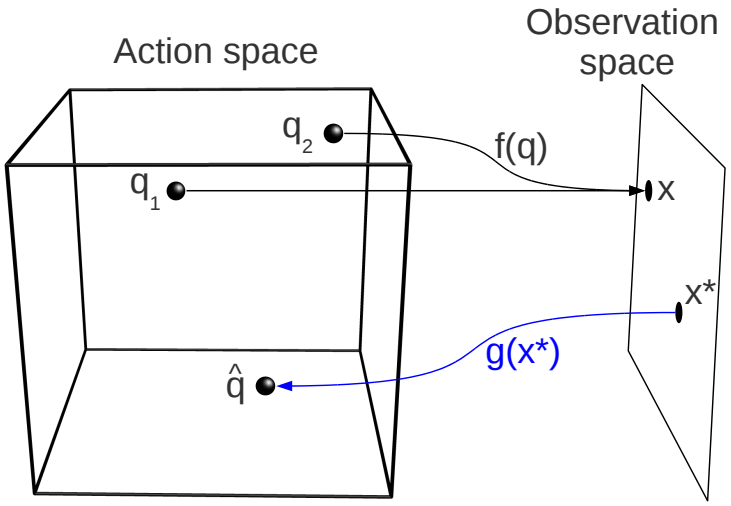


Figure 2.3: An inverse model g suggests an action $\hat{q}$ necessary to achieve a goal x^* and thereby directly solves the coordination problem.

Internal models are widely believed to have a central role in human sensorimotor coordination, and are believed to be located in the cerebellum [Flanagan and Wing, 1997, Wolpert et al., 1998, Kawato, 1999]. Computational models of such cerebellar functioning assume that combinations of forward and inverse models are stored and used for a variety of different tasks [Haruno et al., 1999, 2001, Wolpert and Kawato, 1998]. Internal models are not only argued to be important for sensorimotor coordination, but also hypothesized to be systematically involved in higher cognitive processes [Ito, 2008].

Coordination with Inverse Models

Internal models can be used in a variety of ways to solve different coordination problems [Jordan, 1996, Nguyen-Tuong and Peters, 2011]. The most straightforward way is to directly use an inverse model for the coordination (see figure 2.3):

$$\Omega(x^*, \tau) = g(x^*) = \hat{q} \ . \tag{2.7}$$

The inverse model implements a *direct* functional relation from goal to action and thereby selects exactly one action $\hat{q}$ for a given goal x^*. Therefore, the state variable τ is empty, which largely simplifies equation (2.3) that describes the mastery of a coordination problem:

$$f(g(x^*)) = x^* \quad \forall \ x^* \in \mathbf{X}^* \ . \tag{2.8}$$

Hence, if g solves the coordination problem, it must be a *right-inverse function* of f on the set of goals $\mathbf{X}^*$.

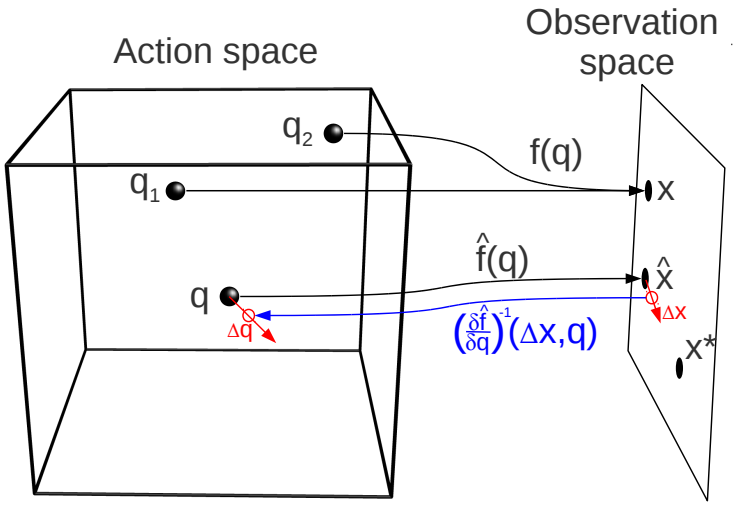


Figure 2.4: A forward model $\hat{f}$ predicts the outcome $\hat{x}$ of some action q. The coordination problem can be solved only indirectly with such a model. A standard scheme is to compute corrective actions Δq based on a desired change Δx of the outcome. This requires to analytically differentiate the forward model and to analytically invert the derivative locally.

Coordination with Forward Models

Forward models are predictors that can be used to predict the outcome of some hypothetical action q, without actually performing the action. Forward models can *not* solve the coordination problem *directly*. Indirect mechanisms to use them for coordination are, however, widely used, and define a process that dynamically *searches* for an appropriate action q by using the known output and shape of $\hat{f}$. A variety of numerical search algorithms can be used in this domain, see [Waldron and Schmiedeler, 2007].

Many standard robots with analytically known kinematic forward functions are actuated by search mechanisms based on the *inverse Jacobian matrix* of the forward model [Liegeois, 1977, Baillieul, 1985, Gienger et al., 2005]. The approach starts from some initial action q. For the kinematic control of robots this is typically the current physical joint configuration. The current outcome x is observed, that can generally differ from the goal x^*, and a corrective movement Δx is attempted towards the goal (see figure 2.4). Obtaining the necessary corrective action Δq requires a local analytic inversion of the forward function. For local movements it holds that

$$\Delta q = J(q)^{-1} \cdot \Delta x \ , \ \text{with} \ \ J(q) = \left(\frac{\delta f_i(q)}{\delta q_j} \right)_{i,j} \in \mathbb{R}^{n \times m} \ . \tag{2.9}$$

Hence, the forward model must be analytically differentiated in order to obtain the

Jacobian matrix $J(q)$. Then, the Jacobian matrix needs to be inverted. In redundant domains with $n < m$ there are infinitely many solutions to that inversion. The selection of one particular action depends on the way the matrix is inverted, in contrast to the inverse model approach which selects a solution directly from the model.

When the corrective action Δq is known, it is integrated on the initial action q and the new action $q + \Delta q$ is executed. New corrections Δx are iteratively applied until the goal x^* is reached. Within the notion introduced in the last section, the overall coordination skill Ω is an iterative search process that is based on the forward model f. In contrast to coordination with inverse models (see equation 2.7), this approach has a state $\tau = q$. It combines some mechanism C that utilizes the forward model f to compute corrective actions based on the current state q and the goal x^*, and a mechanism I that integrates these corrective actions on the state q and executes them:

$$\Omega(x^*, \tau) = \Omega(x^*, q) = I\left(C(f, q, x^*), q\right) \ . \tag{2.10}$$

This scheme makes very indirect use of the internal forward model[2], which appears deeply nested into analytical mechanisms inside equation (2.10). The coordination skill is not solved by the forward model, but solutions are selected by the mechanisms I and C. Consequently, the evaluation how well the scheme solves the coordination skill is substantially harder than evaluating an inverse model for coordination (see equation 2.8). Often "[...] the evaluation is based on the accuracy of the model itself rather than on its control capabilities"[Salaün et al., 2010], whereas a full evaluation would require iterative solution attempts from all possible states $\tau = q$.

Yet, the scheme is important for many studies on the learning of sensorimotor coordination, which either propose to learn a forward model in order to apply such coordination schemes and mimic traditional ways of robotic motor control (see section 2.3), or to use corrective actions as learning signal (see section 2.4).

Feedforward vs. Feedback Control

The notion of forward and inverse models is not to be confused with the control theory terms of feedforward control and feedback control [Jordan, 1996]. *Feedback control* describes coordination mechanisms that incorporate world feedback like the currently observed outcome x into the selection of the next action. It is also possible that the action q can not be perfectly executed such as a joint configuration of a robot that can not be applied, or has not yet been applied due to timing constraints, which can be fed back into the coordination mechanism. *Feedforward control* describes a coordination mechanism that operates without such feedback.

The advantage of feedback controllers is that they typically yield high accuracies in matching x and x^*, because initial errors can be iteratively corrected. Their drawback is that they can not be used if feedback is not available, and that noisy or delayed

[2] In kinematic robotics domains the entire skill Ω [Ulbrich et al., 2012], or the inversion mechanism C [Gienger et al., 2005] is often referred to as "inverse kinematics". This notion is not to be confused with an "inverse model" g of kinematics as discussed in this chapter, which is not present in this scheme.

feedback can cause unstable behavior [Xu et al., 2002, Jordan, 1996]. The opposite holds for feedforward controllers. They typically result in residual deviations between x and x^*, but are insensitive to missing, inaccurate, or delayed feedback.

The typical use of inverse models as described in this section is a feedforward control scheme. The coordination with forward models based on inverse-Jacobians is a feedback control scheme. However, this association of model and control type is not mandatory: Forward models can also be used for feedforward controllers [Pattacini et al., 2010]. Chapter 7 makes use of a scheme that extends the inverse-model-based feedforward scheme with a feedback controller.

2.3 Exhaustive Learning of Forward Models

The learning of forward models for sensorimotor coordination is a heavily investigated and widely used method in motor learning literature. Learning forward models allows to resemble coordination mechanisms that are typically used for the control of robots with analytically known forward functions, and the actual learning appears to be a standard *regression* problem:

- There is a ground truth functional relation f that is to be approximated by the learned forward model $\hat{f}$.
- For any input q of the model, the correct output x (or in stochastic domain the output distribution $P(x|q)$) can be queried by executing the forward function.

Hence, it is possible to collect a data set $D = \{(q_0, x_0), \ldots, (q_{L-1}, x_{L-1})\}$ and learn the forward model, parameterized with some adaptable parameters θ, by reducing the *prediction error* E^P on the data set

$$E^P(D, \theta) = \frac{1}{2L} \sum_{l=0}^{L-1} ||\hat{f}(q_l, \theta) - x_l||^2 \approx \int_q ||\hat{f}(q, \theta) - f(q)||^2 P(q) dq \ , \tag{2.11}$$

which approximates the expected prediction error based on the input distribution $P(q)$ of the actions. This view of the input distribution exposes a central difference between forward model learning for coordination and standard regression problems: $P(q)$ usually corresponds to some (at least empirically) known real world distribution that expresses how likely, and thus relevant, certain inputs to the learner are. For instance in digit recognition [LeCun and Cortes, 1998], the inputs are images with several hundred pixels. Yet, not all possible pixel images are equally likely or even possible within that task.

For the learning of a coordination skill, it is usually not known which actions are relevant to the solution of the coordination problem, and in fact it largely depends on the search mechanism used on top of the forward model which action *will* be used. Since no knowledge on the "true" distribution $P(q)$ during coordination is available, the standard approach is to assume a uniform distribution of all possible actions $q \in \mathbf{Q}$, which corresponds to an *exhaustive* sampling in the action space.

The most frequently used approach to realize this is to sample actions in an entirely random manner [Sun and Scassellati, 2004, 2005, Dearden and Demiris, 2005, Nori et al., 2007, Sturm et al., 2008, Salaün et al., 2010], which is often referred to as "motor babbling" [Bullock et al., 1993, Demiris and Dearden, 2005]. After a distinct phase of data generation, the second step is to learn the forward model based on the generated data, which can then, in a third distinct phase, be used to solve the coordination problem.

The exhaustive sampling of actions can usually be done in low-dimensional domains. Yet, it does not provide a feasible method when $\mathbf{Q}$ is high-dimensional. Generally, the cost of such exploration mechanisms must be assumed to be *exponential* in the action dimension m, because each dimension of q needs to be explored with all combinations of values of the other dimensions. Exploring five different values per dimension in a $m = 5$ dimensional domain results is $5^5 = 3125$ actions to be explored, which is typically feasible. Applying the same pattern in $m = 20$ dimensions already gives $5^{20} \approx 9,5 \cdot 10^{13}$ actions, which is clearly too much to be explored within the lifetime of any agent.

Several approaches have been suggested to improve the feasibility of learning forward models for coordination. One is concerned with the incorporation of *prior knowledge*. For the reaching coordination of standard robots with revolute joints, it is indeed possible to exploit that there are only revolute joints. An approach to identify the analytic parameters of such kinematic chains was presented in [Hersch et al., 2008]. Related to that idea, Ulbrich *et al.* presented a method that exploits the fact that revolute joints only produce circular movements when one joint is moved. Their approach allows to make an exact match $\hat{f} = f$ for revolute joint robots if the training data does not contain noise. Within their formulation it is possible to exactly pinpoint the number of examples needed to 3^m [Ulbrich et al., 2012]. It is reasonable to assume that approaches without prior knowledge need substantially more examples in order to learn an accurate forward model for all actions. Still, 3^m is not applicable in high dimensions, and results in more than three billion exploratory actions in $m = 20$.

Another approach utilizes the exploration concept of *active learning*. This approach intertwines data generation and learning and attempts to iteratively generate examples that are maximally informative for learning [Settles, 2010]. Several studies have shown that this concept allows to reduce the absolute number of examples necessary to learn accurate forward models [Baranes and Oudeyer, 2009, Martinez-Cantin et al., 2010]. While active learning can avoid the generation of uninformative examples (for instance for inputs where the forward model is already accurate), it can not avoid the exhaustive character of the exploration. Active learning still aims at the error reduction over the entire input distribution $P(q)$ of the learner [Cohn et al., 1996].

Related Approach: Associative Models

An approach that exhibits a similar overall organization of exploration, learning, and coordination uses *associative* methods in order to represent and learn a coordination skill. Instead of learning internal models as described in this chapter, they attempt

to learn possible combinations of q and x in an associative memory. When these combinations are retrieved, a dynamical search process [Walter et al., 2000, Lopes and Damas, 2007] allows to query various relations, in particular $q \rightarrow x$ and $x \rightarrow q$, but also combinations in which only particular dimensions are given. In recurrent neural network implementations this selection is done by the inherent dynamics of the network [Butz et al., 2007, Reinhart and Steil, 2008, 2011].

These models *contain* a forward model, since the functional relation $q \rightarrow x$ can be queried. An inverse model as a function is typically not included, since the resulting action q for a target x^* can depend of the internal state of the search process or dynamical system. The overall organization of the existing studies follows the same three phases of the exhaustive forward model learning: exhaustive data generation, then learning of all possible q/x combinations, then exploitation for the specific coordination problem.

2.4 Learning of Inverse Models

Inverse models correspond to a direct solution of the coordination problem. It is straightforward to solve coordination with an inverse model (equation 2.7) and to evaluate how well it performs during learning: equation (2.8) denotes that an inverse model must be a right-inverse function of f in order to solve the coordination problem. The condition compares the goals x^* with the outcomes $x = f(g(x^*))$ in the observation space. For a finite set of goals $\mathbf{X}^* = \{x^*_0, \ldots, x^*_{K-1}\}$ this directly leads to the *performance error* E^X, which measures how close an inverse estimate[3] with parameters θ is to a solution:

$$E^X(\mathbf{X}^*, \theta) = \frac{1}{2K} \sum_{k=0}^{K-1} ||f(g(x^*_k, \theta)) - x^*_k||^2 \approx \int_{x^*} ||f(g(x^*, \theta)) - x^*||^2 P(x^*) dx^* \ . \quad (2.12)$$

In this case, the input distribution $P(x^*)$ of the learner is typically known at least as an empirical set. When $\mathbf{X}^*$ is not a finite, but continuous set, it is reasonable to assume a uniform distribution: in contrast to forward model learning the input space of the model is typically low-dimensional, so that it can be effectively sampled. However, optimizing this error functional is *clearly not* a regression problem, since the learner's output is not compared to a ground truth value, and in particular it is *not* directly possible to query a ground truth output.

Error-based Learning in the Observation Space

Error-based approaches repeatedly perform a *goal-directed* exploration step in order to measure the actual position x when trying to reach for a target x^*:

$$q = g(x^*, \theta) \ , \quad x = f(q) \ . \quad (2.13)$$

[3] In this thesis the term "inverse estimate" is used, instead of "inverse model", to indicate that some $g(\cdot, \theta)$ is not readily trained, but learning is currently in progress.

Then, a parameter-change is attempted that causes a direct correction of the observation along $\Delta x = (x^* - x)$.

A frequently used approach is to realize this by gradient descent on the performance error E^X. Differentiating E^X with respect to the parameters gives the *performance gradient* for the single goal x^*:

$$\frac{\partial E^X}{\partial \theta} = \frac{\partial g(x^*, \theta)}{\partial \theta}^T \frac{\partial f(q)}{\partial q}^T (x - x^*) = \frac{\partial g(x^*, \theta)}{\partial \theta}^T J(q)^T (x - x^*) \tag{2.14}$$

The first term $\frac{\partial g(x^*, \theta)}{\partial \theta}^T$ is specific to the learner and known for any function approximation scheme. The last term $(x - x^*)$ is known from the exploration step. Yet, the scheme requires to know the Jacobian matrix $J(q)$ of the forward function, which is not directly accessible in the coordination problem. Computing this exact gradient requires analytic knowledge about the forward function.

A gradient descent step on this error with some step-width η can be written as:

$$\begin{aligned} \Delta\theta &= -\eta \frac{\partial E^X}{\partial \theta} = -\eta \frac{\partial g(x^*, \theta)}{\partial \theta}^T J(q)^T (x - x^*) \\ &= \eta \frac{\partial g(x^*, \theta)}{\partial \theta}^T J(q)^T (x^* - x) = \eta \frac{\partial g(x^*, \theta)}{\partial \theta}^T \underbrace{J(q)^T \Delta x}_{\Delta q} , \end{aligned}$$

Hence, the scheme tries to achieve some corrective outcome Δx by means of some corrective action[4] Δq. This Δq has a tight relation to the feedback control schemes based on the differentiation of forward models. While these feedback control schemes are typically driven with the pseudo-inverse of the Jacobian matrix (see equation 2.9), it is also possible to use the transpose of the Jacobian [Wolovich and Elliot, 1984, Baillieul, 1985], as it is present in the performance gradient. Hence, the knowledge of the term $J(q)^T \Delta x$ could directly solve the coordination problem by means of feedback control.

In *feedback-error learning* [Kawato, 1990] it is simply assumed that a mechanism to derive this corrective action, and thus a feedback controller, is already given, which leads to successful and stable results for learning [Miyamura and Kimura, 2002]. Learning with *distal teacher* [Jordan and Rumelhart, 1992] avoids a pre-existing controller, but requires to first learn a forward model $\hat{f}$. The corrective action Δq can then be approximated by analytically differentiating the forward model. However, this scheme directly inherits the scalability problems of forward model learning as discussed in the last section, since it requires an exhaustive exploration before the inverse model can be learned.

An alternative approach has been developed in [Porrill et al., 2004, Porrill and Dean, 2007], that operates directly on the desired observation change Δx, without deriving a

[4] In literature on physiological motor learning this term is often called "motor error" [Kawato and Gomi, 1992]. Yet, this wording is not used throughout this thesis in order to avoid confusion with other error measures.

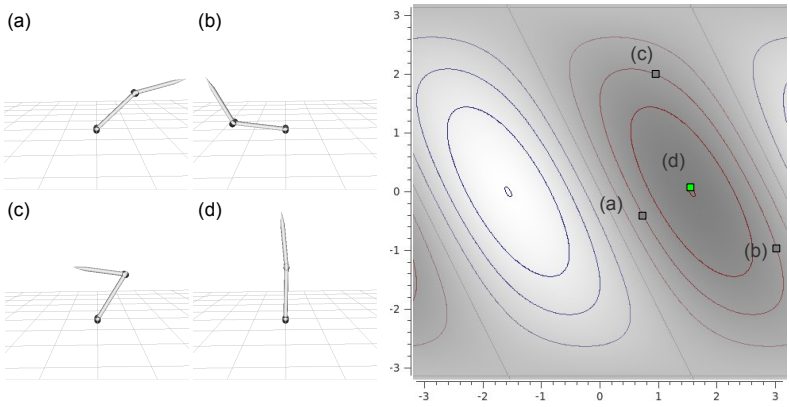


Figure 2.5: Multiple actions (a-c) for the same effector height x are located on a non-convex solution set. The average action (d) results in a different height.

corrective action Δq. Instead, the approach starts with an initially inaccurate inverse model and learns to virtually shift the goals x^* along Δx. However, that scheme only works if the initial inverse estimate is already very close to a solution and can not be used for bootstrapping.

The critical aspect of these approaches is that the learning of the actual inverse model requires prior-knowledge. Feedback-error learning and the distal teacher approach require to derive a corrective action. This mechanism alone could solve the coordination problem. More importantly, it requires knowledge about the forward function f over the entire set of actions $\mathbf{Q}$, since the learning schemes require that a corrective action can be queried for any q. This dependency is most visible in learning with distal teacher, because it requires an explicit, exhaustive pre-exploration in order to learn the forward model. Accordingly, this approach is not suitable for a bootstrapping of inverse models in high dimensions, because either prior knowledge, or an inefficient exhaustive pre-training is required. The approach to shift inputs to the inverse model is likewise unable to bootstrap a coordination skill since it requires a good initial solution.

Example-based Learning in the Action Space

Another approach is to learn inverse models directly from examples collected by exploration. Similar to the learning of forward models, a data set $D = \{(q_0, x_0), \ldots, (q_{L-1}, x_{L-1})\}$ is collected by starting exploration in the action space, and observing the outcomes of the selected actions. Then the learning of the inverse model is *treated as* regression problem by *fitting* the inverse model to the exploratory data. Here the *action error*

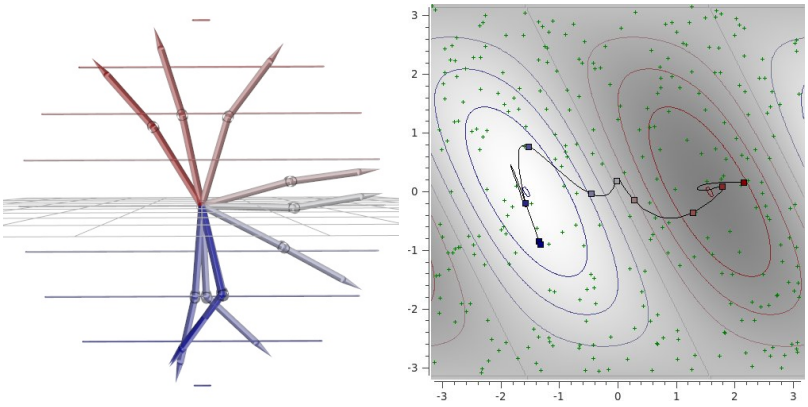


Figure 2.6: Non-convex solution sets forbid learning an inverse model from random data-sets (green points). The suggested solutions (colored postures left, and correspondingly colored markers right) do not reach the desired height.

E^Q measures in action space how well the inverse estimate already describes D:

$$E^Q(D,\theta) = \frac{1}{2L}\sum_{l=0}^{L-1} ||g(x_l,\theta) - q_l||^2 \ . \tag{2.15}$$

If the coordination problem does not contain redundancy, there is an implicit ground-truth function f^{-1} to which g is fitted. In this case, a valid inverse model can be learned from arbitrary data sets D. For low-dimensional n and $m = n$ this is typically done with exhaustive motor babbling [Kuperstein, 1988].

Problems arise in the case of redundancy. In the data set it holds that $x_l = f(q_l)$, so that the action error can be rewritten as:

$$E^Q(D,\theta) = \frac{1}{2L}\sum_{l=0}^{L-1} ||g(f(q_l),\theta) - q_l||^2 \ . \tag{2.16}$$

Compared to the performance error (equation 2.12) which corresponds to learning a right-inverse function as necessary for coordination, this error corresponds to learning a *left inverse* function such that $g(f(q)) = q$. While right-inverses are an ill-posed problem in redundant domains, such left-inverse functions *do not exist*: when different actions q_i evaluate to the same outcome $f(q_i) = x$, there is no function g that could reconstruct the original action. When multiple examples with $q_i \neq q_j$ and $x_i = x_j$ are used for learning, this leads to averaging. In non-linear redundant domains the sets of redundant solutions can generally have a *non-convex* shape, such that averaging leads to invalid solutions [Jordan and Rumelhart, 1992]. This effect is illustrated in figure 2.5 based on the toy-example of inverse kinematics introduced in section 2.1. The redundancy manifolds, illustrated as colored lines in the action space (right),

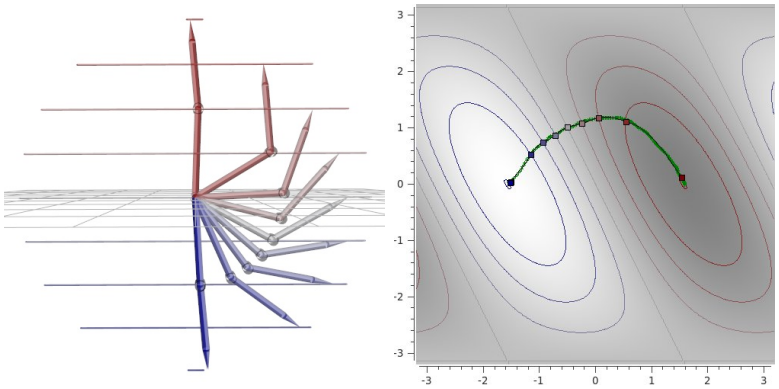


Figure 2.7: An inverse model is trained with an expert-generated data set that contains exactly one valid action for any desired outcome.

have an oval, non-convex shape. If several actions with the same effector height are averaged, this leads to an action with a different effector height. The effect on an entire inverse model is shown in figure 2.6. In this domain an inverse model can be visualized by a $n = 1$ dimensional manifold through the action space. Suggested actions $q = g(x^*)$ for several color-coded values of x^* are shown as markers in the action space and corresponding postures are visualized on the left side. The inverse model solves the coordination problem if each of these suggested actions lies exactly on the manifold with the corresponding color. This is clearly not the case in the example, which has been trained with 250 random actions, shown as green dots. Arbitrary or random data sets can not be used to learn inverse models when non-convex solution sets exist. Special solutions have been proposed for cases with only a discrete number of solutions and $n = m$: a very dense sampling in such domains can be used to segment the action space and then learn one inverse estimate for each segment [Demers and Kreutz-Delgado, 1992].

Learning inverse models in general redundant domains can not be done with exhaustive sampling. Yet, there is no principle need to know the entire action space. In redundant domains an inverse model *selects* one valid action for each goal. Hence, only one solution needs to be known. Figure 2.7 illustrates the learning of an inverse model from a data set containing exactly one possible path of solutions. The training data (green dots) passes each of the redundancy manifolds exactly one time, and thus contains no inconsistent solutions with $q_i \neq q_j$ and $x_i = x_j$. On such a data set, it is possible to represent a *partial left inverse* function that is only a left-inverse of f on the training data, such that $g(f(q_l)) = q_l$. Then, it is straightforward to show that the inverse model is also a right-inverse on the training data:

$$g(f(q_l)) = q_l \quad \Rightarrow \quad f(g(f(q_l))) = f(q_l) \quad \Leftrightarrow \quad f(g(x_l)) = x_l \ . \tag{2.17}$$

Hence, the coordination problem can be solved if the observed outcomes x_l also span

the set of goals $\mathbf{X}^*$. A potential advantage, if an exploration strategy to generate such data autonomously could be devised, is that the training data only covers an n-dimensional subspace within the m-dimensional set of actions. Since n is typically small, such spaces can be sampled *efficiently* which provides a direct account to *scalability* to many degrees of freedom.

Generating such an exploratory data set that (i) contains all goals in $\mathbf{X}^*$ as outcomes x_l and (ii) does not contain inconsistent actions is, however, far from trivial. Prior to the work described in this thesis, the only known way was to let an expert generate the data set [Rolf et al., 2009], which reflects deep prior knowledge about the particular coordination problem.

Related Approach: Differential Inverse Models

Problem formulations that typically allow to learn inverse models from arbitrary data sets are differential formulations like differential kinematics [Mel, 1991, D'Souza et al., 2001] or inverse dynamics [Peters and Schaal, 2008, Nguyen-Tuong et al., 2008]. An inverse model in a differential kinematics formulation represents corrective actions directly with a model $g(\Delta x, q) = \Delta q$. When Δx and Δq correspond to small movements the non-convexity can typically be neglected, since the redundancy manifolds of differentiable functions are locally linear, and thus locally convex [D'Souza et al., 2001]. This scheme is highly related to the feedback control based on learned forward models. The idea is to directly learn the, otherwise computed, corrective action for feedback control. It therefore inherits the problem that the entire action space (which is also the coordination skill's space of states $\tau = q$) must be known in order to fully solve the coordination problem. Consequently, the approaches for learning such models start with an exhaustive motor babbling [D'Souza et al., 2001, Peters and Schaal, 2008], after which the model can be refined while performing goal-directed actions.

Chapter 3

A Framework for Goal Babbling

> "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's?" [Turing, 1950]

3.1 Inspiration from Infant Development

The standard models for the learning of coordination skills discussed in the last chapter demand either an exhaustive exploration of all actions, or prior knowledge about the action space and forward function. Therefore the acquisition of the coordination skill is divided into separate stages of (random) exploration, learning, and exploitation of the learned mechanisms. However, exhaustive exploration does neither provide an explanation of infants' efficiency in sensorimotor development, nor does it provide a feasible approach for artificial agents to learn in high-dimensional domains.

Nevertheless, the generation of random actions by means of motor babbling has been repeatedly motivated [Kuperstein, 1988, Bullock et al., 1993, Caligiore et al., 2008] by Piaget's view on infant development [Piaget, 1953]. Piaget suggested that development is organized in distinct stages and that, at first, infants do not perform purposeful actions. "The implication of [Piaget's] proposal is that the early behavior of the neonate is essentially random and insensitive to contextual information. Recent research suggests that some re-thinking of this extreme position is necessary" [Bertenthal, 1996]. Contrary to Piaget's suggestions, and the random motor babbling approach, infant developmental studies over the last three decades have found conclusive evidence for coordinated behavior even in newborns. Examples include orienting towards sounds [Clifton et al., 1981], tracking of visual targets [Bloch and Carchon, 1992], and apparent reflexes that have been re-discovered as goal-directed actions [van der Meer et al., 1995, van der Meer, 1997]. "These behaviors are fragile and inconsistent, which explains why they were overlooked for quite some time" [Bertenthal, 1996].

In the case of reaching, it has been shown that newborns attempt goal-directed movements already few days after birth [von Hofsten, 1982, Rönnqvist and von Hofsten, 1994]. Von Hofsten showed that, when salient objects are in the visual field, infants produce more arm movements towards that object, than movements away from it. This indicates a strong role of "learning by doing" instead of random exploration and that infants learn to reach by trying to reach: "Before infants master reaching, they spend hours and hours trying to get the hand to an object in spite of the fact that

they will fail, at least to begin with" [von Hofsten, 2004]. From a machine learning point of view, these findings motivate to devise methods that closely intertwine exploration, learning, and exploitation, instead of organizing these aspects in distinct and subsequent stages.

Findings of early goal-directed actions are complemented by studies investigating the structure of infants' reaching attempts over the course of development. When infants perform the first *successful* reaching movements around the age of four months, these movements are controlled in an entire *feedforward* manner [Clifton et al., 1993, Out et al., 1998]. This strongly indicates the use of an inverse model as discussed in the last chapter, which selects one solution and applies it without corrections. The importance of feedforward control does not diminish over the course of development, which is well known from prism-glass experiments [Baily, 1972], but the skill is later on augmented by mechanisms that allow for more adaptive movements and error corrections by means of visual feedback [Bushnell, 1985]. Moreover, the earliest reaching movements are rather jerky and suboptimal in the sense that the distribution and timing of muscular forces is more complicated than actually necessary [Konczak et al., 1995, 1997, Berthier and Keen, 2005].

In short, infants appear to follow a very efficient pathway, on which one initial solution is learned, and directly used for goal-directed behavior. Only later on these movements are gradually optimized and become more adaptive. While this pathway is very intuitive, it is orthogonal to the motor-babbling approach which first attempts to gather full knowledge about the sensorimotor space, from which particular solutions can be derived afterwards.

3.2 Concept: Goal Babbling

The general idea that connects early goal-directed movements and initial feedforward control is to take redundancy as an *opportunity* to reduce the demand for exploration, instead of a burden that has to be dealt with. If there are multiple ways to achieve some behavioral goal, there is no inherent need to know all of them. Of course, this requires an exploration mechanism that can generate relevant training data without exhaustive exploration. The hypothesis of this thesis is that early goal-directed movements do not only reflect an early exploitation of knowledge, but that they constitute the very mechanism to *generate* that knowledge by exploration, and therefore enable an efficient learning of valid solutions for the coordination problem. Consequently, the first distinct research goal of this thesis concerns the general mechanism of goal-directed exploration:

> **Research goal 1**: Conceptualize and understand early goal-directed movements as mechanism for the bootstrapping of coordination skills.

As a basis for this investigation, this thesis introduces the notion of "goal babbling" (based on the first mentioning in [Rolf et al., 2010c]):

> **Definition: Goal babbling** is the bootstrapping of a coordination skill by repetitively trying to accomplish multiple goals related to that skill.

A central aspect is, of course, trying to accomplish goals, which corresponds to infants' attempts to perform goal directed movements. Still, several other aspects of this definition need to be highlighted in order to distinguish this concept from other approaches:

- Goal babbling aims at the *bootstrapping* of coordination skills. In contrast, goal-directed exploration has been used in several approaches to sensorimotor learning, but has only been used for fine-tuning of well initialized models [D'Souza et al., 2001, Peters and Schaal, 2007], or requiring other prior knowledge [Kawato, 1990, Jordan and Rumelhart, 1992, Porrill et al., 2004].
- Goal babbling defines this as a *repeated* process, which implies that the skill acquisition is incremental and ongoing, as opposed to stage-like organizations of exploration and learning [Bullock et al., 1993, Demiris and Dearden, 2005].
- Goal babbling applies to domains with multiple related goals. In the coordination problem formulation in section 2.1 this is naturally given by a set of goals situated in a continuous observation space. Even if one goal can not be achieved, the learner can observe the outcome and learn how to achieve that state if desired. This exploration across multiple goals stands in contrast to typical scenarios in reinforcement learning, in which only a single desired behavior is considered [Theodorou et al., 2010], and also algorithms in coordination domains which perform goal-directed exploration in order to achieve a single goal [Schaal and Atkeson, 1994].
- Goal babbling considers the "trying to accomplish" itself as a primary mechanism, in contrast to conceptualizations of intrinsic motivations [Oudeyer and Kaplan, 2008] that consider goals or more general intentions within active learning architectures. The latter one focuses on the role of active learning, while this concept focuses on the distinct impact of goal-directed exploration.

Given this research goal and the definition of goal babbling, several questions need to be asked:

- Is goal babbling *possible* at all, and what are the mechanisms necessary to enable it? Early studies have consequently failed to enable a goal-directed bootstrapping in a reliable manner [Oyama and Tachi, 2000, Sanger, 2004]. This question will be addressed in the chapters 4 and 5.
- Does it actually permit a bootstrapping that is *scalable to high dimensions*? This question will be addressed mainly in the chapters 5 and 6.
- What are observable characteristics of such a bootstrapping process that closely intertwines exploration and learning? Results along this question will be discussed in the chapters 4 to 6.

3.3 Method: Learning Inverse Models from Examples

Goal babbling does not refer to a particular algorithm, but to a concept that can be methodically investigated by various means. An approach that has been proposed in parallel to the work described in this thesis, and is compatible with the concept of goal babbling, has been introduced in [Baranes and Oudeyer, 2010a]. Baranes' model attempts to learn an instance-based associative memory in which a search algorithm can be applied, which can be viewed as the learning of a partial *forward model* [Baranes and Oudeyer, 2013]. Goal babbling then generates a distribution of actions that can be quickly exploited and does not need to sample the entire action space.

In contrast to Baranes' model, this thesis investigates the learning of *inverse models* by means of goal babbling, and therefore focuses on learning the coordination skill directly, without relying on analytical inversion mechanisms. This approach resembles infants' developmental pathway, which serves as an example of efficiency, by acquiring at first one valid solution that can be used for feedforward control.

Focusing on inverse models leaves a choice between error-based and example-based learning. The demand for a bootstrapping mechanism clearly disqualifies error-based methods due to their inherent need for prior knowledge. Hence, this thesis focuses on *example-based learning* of inverse models as a method to investigate goal babbling. Learning inverse models by fitting examples was believed to be impossible due to the non-convex solution sets in non-linear redundant domains [Jordan and Rumelhart, 1992]. Consequently, the second research goal concerns this methodological aspect:

> **Research goal 2**: Enable the learning of inverse models from examples in non-linear and redundant domains.

Finding an exploration scheme that can realize this goal clearly needs to cope with non-convex solution sets. Previous studies have only shown how to deal with non-convexity locally, either by reformulating the problem into a differential one [D'Souza et al., 2001], or by using prior knowledge to start learning from a well-initialized state [Schenck, 2008]. Chapter 5 will introduce a method to deal with non-convexity directly and through the entire bootstrapping process. However, non-convexity is not the only problem to deal with. While non-convexity makes it difficult to handle multiple solutions q for the same outcome x, the initial problem is to *find at least one correct solution* to realize the desired outcomes in $\mathbf{X}^*$ and, hence, to invert the causal relation of the forward function in a reliable manner. This *inversion of causality* is a general problem for exploration schemes, since the direction $x \rightarrow q$ can not be directly queried within the coordination problem. Random motor babbling can theoretically solve the problem because it simply explores all actions, such that the necessary ones are also explored. This, however, is practically not feasible in high-dimensional domains. The inversion of causality has a distinct characteristic in goal-directed exploration schemes which tend to get stuck in only partial solutions of the coordination problem [Atkeson, 1989, Oyama and Tachi, 2000, Sanger, 2004], in which only a subset of $\mathbf{X}^*$ can be successfully realized. The general pattern to solve that problem is to introduce exploratory noise into the process [Peters and Schaal, 2007, Schenck, 2008]. Chapter

4 investigates goal babbling with exploratory noise and provides a general theoretical framework that describes the relation between example-based learning (minimizing E^Q) and error-based learning (minimizing E^X) in linear domains.

After enabling goal babbling, and particularly learning inverse models from examples, the consequential goal is to make this method practically useful in high-dimensional real-world scenarios:

> **Research goal 3**: Devise a practical algorithm for goal babbling that is scalable, fast, and applicable in real-world scenarios.

A first investigation of scalability is provided in chapter 5. However, the question of absolute speed is mainly discussed in chapter 6. For a practical application, the number of examples needed for learning must be small enough to be executed in reasonable time. Chapter 6 shows that the learning can, even in high-dimensional domains, be fast enough if online learning is applied. The experiments point out that goal babbling constitutes a positive feedback loop during bootstrapping, in which exploration and learning reinforce each other. This positive feedback loop is identified as an important conceptual property of goal babbling. Experiments demonstrate that it allows to achieve human-level learning speed.

Chapter 7 finally investigates the practical use of the approach to learn the inverse kinematics of the *Bionic Handling Assistant*. The application of goal babbling on this bionic robot faces several practical problems not investigated in the other chapters like sensory noise, delayed execution of actions, actions that are not executable due to physical limits, and non-stationary system behavior. The experiments show that goal babbling can deal with these challenges and yields accurate inverse models. Further, chapter 7 shows how an additional feedback controller can be used on top of the inverse model, which is otherwise only used for feedforward control. This scheme allows to utilize the insensitivity of feedforward control to noisy and delayed feedback by means of the learned inverse model, plus the ability to fine-tune the movement by feedback control if necessary.

Chapter 4

Inversion of Causality in Linear Domains

Learning inverse models from examples is a barely theoretically investigated topic in machine learning literature. Previous studies only provided negative results by showing the impossibility to learn from non-convex solution sets [Jordan and Rumelhart, 1992] or by showing failure modes of simple goal-directed exploration schemes [Sanger, 2004]. This chapter studies the theoretical basis of inverse model learning by investigating linear domains. Linear domains do not contain non-convex solution sets, and yet allow to study the effect of redundancy. An early theoretical study of the example-based learning of inverse models by means of goal-directed exploration was presented in [Sanger, 2004]. Sanger showed that even in simple non-linear domains, without redundancy, such learning can fail when the model is not well initialized. The investigation in this chapter complements this negative outcome with an analysis of linear redundant cases. Learning inverse models from examples is investigated by comparing its learning gradients, that are shaped by the explored examples, against the performance gradient that serves as a reference direction through the space of parameters of the learner. The analysis first re-investigates the setup of Sanger and shows additional failure modes that are caused by redundancy. Then it is shown that the addition of exploratory noise to goal-directed exploration leads to the discovery of valid solutions, and thereby solves the inversion of causality, which are moreover optimal in a least-squares sense. Parts of the content of this chapter have been published in [Rolf and Steil, 2012b, 2013a].

4.1 Two Spaces and their Gradients in Linear Domains

In a linear domain, the relation between actions $q \in \mathbf{Q} \subseteq \mathbb{R}^m$ and outcomes $x \in \mathbf{X} \subseteq \mathbb{R}^n$ is given by a linear forward function:

$$x = f(q) = M \cdot q \ . \tag{4.1}$$

The real-valued matrix $M \in \mathbb{R}^{n \times m}$ with $n \leq m$ is thereby assumed to have full rank $rank(M) = n$. This implies that f is surjective, i.e. that any outcome x in $\mathbb{R}^n$ can be achieved by some action q in $\mathbb{R}^m$ and therefore expresses that the coordination problem is solvable. For $n = m$ this formulation does not contain redundancy, but there is an exact one-to-one relation between actions q in outcomes x in $\mathbb{R}^n$. For $n < m$ there are infinitely many solutions q for any outcome x, which appear as linear subspaces in $\mathbb{R}^m$.

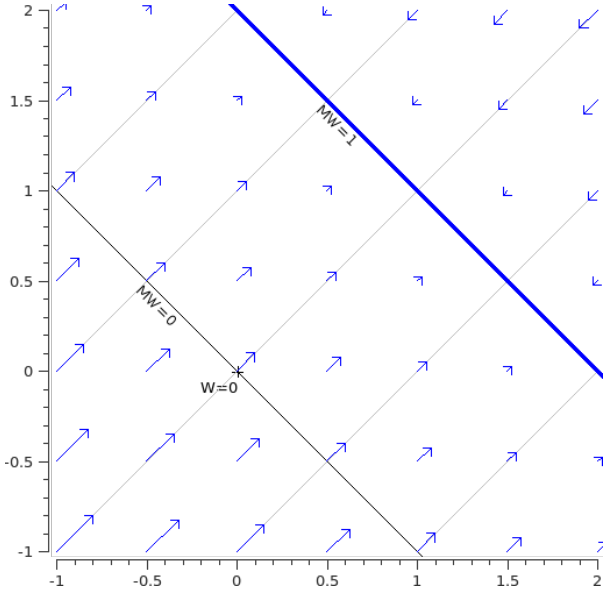


Figure 4.1: The learning gradient $-\frac{\partial E^X(W)}{\partial W}$ is shown in the parameter space of a simple example with $M = (0.5, 0.5) \in \mathbb{R}^{1\times 2}$ and learning parameters $W \in \mathbb{R}^{2\times 1}$. Any value W is straightly pulled towards $MW = 1$.

Such linear redundancy manifolds are obviously convex, so that different solutions can be safely averaged.

In correspondence to a linear forward function, an inverse model can be denoted with a linear representation, using parameters $W \in \mathbb{R}^{m\times n}$ that are adaptable by learning:

$$g(x^*, W) = W \cdot x^* \ . \tag{4.2}$$

Considering some set of goals $\mathbf{X}^* \subseteq \mathbf{X}$, the mastery of coordination can then be rephrased by a simple linear algebra expression. Generally, an inverse model has to be a right-inverse function of f in order to solve the coordination problem (compare equation 2.8):

$$\begin{aligned} & f(g(x^*)) = x^* \ \ \forall\ x^* \in \mathbf{X}^* \\ \Leftrightarrow \quad & MWx^* = x^* \ \ \forall\ x^* \in \mathbf{X}^*. \end{aligned}$$

Assuming that the set of goals $\mathbf{X}^*$ spans the entire space $\mathbf{X}$ further simplifies the condition to:

$$f(g(x^*)) = x^* \ \ \forall\ x^* \in \mathbf{X}^* \quad \Leftrightarrow \quad MW = \mathbb{1}_n. \tag{4.3}$$

Hence, W must be a right inverse matrix of M in order to solve the coordination problem. This equation is exactly solvable in W. For $n < m$ it is ill-posed and multiple solutions W exist.

As introduced in section 2.4, the *performance error* measures how well an inverse model solves the coordination problem, and therefore the deviation from the right-inverse condition (4.3). For a linear forward function and inverse models, and a finite set of goals $\mathbf{X}^* = \{x_0^*, \ldots, x_{K-1}^*\}$, this error functional is:

$$E^X(\mathbf{X}^*, W) = \frac{1}{2K} \sum_{k=0}^{K-1} ||f(g(x_k^*, W)) - x_k^*||^2 = \frac{1}{2K} \sum_{k=0}^{K-1} ||MWx_k^* - x_k^*||^2 \ . \tag{4.4}$$

Computing the derivative with respect to W gives the *performance gradient*:

$$\frac{\partial E^X(\mathbf{X}^*, W)}{\partial W} = \frac{\partial E^X(\mathbb{X}^*, W)}{\partial W} = M^T(MW - \mathbb{1}_n)\mathbb{X}^* \tag{4.5}$$
$$\text{with} \quad \mathbb{X}^* = \frac{1}{K} \sum_{k=0}^{K-1} x_k^* x_k^{*T} \in \mathbb{R}^{n \times n}.$$

Figure 4.1 shows the performance gradient in relation to correct right inverse solutions. An exemplary problem is chosen with a forward matrix $M = (0.5, 0.5) \in \mathbb{R}^{1\times 2}$. The figure shows the parameter space of $W \in \mathbb{R}^{2\times 1}$. Right inverse matrices fulfill $MW = \mathbb{1}_1$ or in scalar notation $MW = 1$. These solutions give $\frac{\partial E^X}{\partial W} = 0$. The performance gradient drives any value of W straight to that solution manifold. As argued in chapter 2, this gradient is not directly accessible during learning and the example-based approach used in this thesis avoids the detour to estimate it. Yet, the performance gradient serves as an important theoretical tool. It expresses the steepest direction in the parameter space to improve the performance of coordination. Consequently, it serves as a reference direction, which must be at least roughly followed by any learning mechanism.

Learning an inverse model from examples generally considers a data set $D = \{(x_l, q_l)\}_l$ which has been generated by performing actions q_l and observing the outcomes $x_l = f(q_l) = Mq_l$. The initial analysis does *not* assume a particular exploration mechanisms to generate the actions q_l, so that the data set can have arbitrary structure. Learning is performed by fitting the inverse model to that data, which is done by reducing the action error (compare equation (2.15)):

$$E^Q(D, W) = \frac{1}{2L} \sum_{l=0}^{L-1} ||g(x_l, W) - q_l||^2 = \frac{1}{2L} \sum_{l=0}^{L-1} ||WMq_l - q_l||^2 \tag{4.6}$$

Using the correlation matrices

$$\mathbb{Q} = \sum_{k=0}^{L-1} q_l {q_l}^T \quad \text{and} \quad \mathbb{X} = \sum_{k=0}^{L-1} x_l {x_l}^T = M\mathbb{Q}M^T$$

the corresponding *action gradient* can be derived, which is utilized to reduce E^Q by

means of gradient descent:

$$\frac{\partial E^Q(D,W)}{\partial W} = \frac{\partial E^Q(\mathbb{Q},W)}{\partial W} = (WM - \mathbb{1}_m)\mathbb{Q}M^T. \tag{4.7}$$

Learning an inverse model from examples can only succeed if following the action gradient (minimizing E^Q) also reduces the performance error E^X. Therefore, the action gradient does not have to be identical with the performance gradient, but must have at least a non-negative angle to the performance gradient, which means that both gradients must not differ by more than 90°. For the general case with non-linear forward functions and arbitrary data sets such positive angles can obviously not be guaranteed, since learning from such data fails in the presence of non-convex solution sets. For the linear case, however, a tight relation between both gradients can be shown:

Theorem 1. *For any data set D, the action gradient is related to the performance gradient on the observed $\{x_l\}$ positions by*

$$M^T M \frac{\partial E^Q(\mathbb{Q},W)}{\partial W} = \frac{\partial E^X(\mathbb{X},W)}{\partial W} \ . \tag{4.8}$$

Proof.

$$\begin{aligned} M^T M \frac{\partial E^Q(\mathbb{Q},W)}{\partial W} &\overset{(4.7)}{=} M^T M (WM - \mathbb{1}_m)\mathbb{Q}M^T \\ = M^T(MWM - M)\mathbb{Q}M^T &= M^T(MW - \mathbb{1}_n)M\mathbb{Q}M^T \\ = M^T(MW - \mathbb{1}_n)\mathbb{X} &\overset{(4.5)}{=} \frac{\partial E^X(\mathbb{X},W)}{\partial W} \end{aligned}$$

□

Both gradients have a non-negative angle since $M^T M$ is a positive semi-definite matrix. Minimizing E^Q by gradient descent will never increase the performance error on the observed positions $\{x_l\}$. For $n{=}m$, $M^T M$ is even positive definite which guarantees a positive angle. Hence, learning a right inverse function is generally possible by minimizing E^Q in linear domains. For $n{<}m$, $M^T M$ becomes singular and the action gradient can project into its nullspace, leaving the performance error unchanged. This makes the redundant case mildly more complicated, but not as difficult as the general non-convex case, for which no angle can be guaranteed for arbitrary data sets. Generally, theorem 1 qualifies example-based learning as a *sound mechanism* for obtaining inverse models in linear domains. Note, however, that this theorem does *not* give a direct relation to the performance on the actual goals $E^X(\mathbb{X}^*, W)$. Whether a right inverse for all goals $\mathbf{X}^*$ can be learned still depends on what data set D is generated by exploration and whether the observations $\{x_l\}$ span the entire space $\mathbf{X}$.

4.2 Fixpoint Analysis for Explorative Learning

How the parameters W are adapted during learning depends on how the data set D, generated by exploration, shapes the action gradient. The general mechanism of learning from examples is to apply gradient descent on E^Q. Starting from some initial parameter value W_0, the parameters are iteratively updated with the learning equation

$$W_{t+1} = W_t - \eta \frac{\partial E^Q(\mathbb{Q}, W)}{\partial W} \ . \tag{4.9}$$

Mastering a coordination skill requires to obtain a right-inverse function, to that the most important question is whether learning converges to a W that satisfies $MW = \mathbb{1}_n$. In order to check for this behavior, the following analysis investigates the fixpoints of the learning equation depending on D. A fixpoint $W_{t+1} = W_t$ is obviously given if and only if the action gradient becomes zero. The following two theorems provide general conditions for which combinations of parameter values W and data sets D the action gradient becomes zero.

Theorem 2 (Sufficient fixpoint condition)**.** *If W is a partial left inverse of M on the explored actions (i.e. $WMq_l = q_l \forall q_l \in D$), then W is a fixpoint of equation (4.9).*

Proof.

$$WMq_l = q_l \forall q_l \in D \Leftrightarrow WM\mathbb{Q} = \mathbb{Q} \Leftrightarrow (WM - \mathbb{1}_m)\mathbb{Q} = 0$$

Right-multiplication with M^T gives:

$$\Rightarrow (WM - \mathbb{1}_m)\mathbb{Q}M^T \overset{(4.7)}{=} \frac{\partial E^Q(\mathbb{Q}, W)}{\partial W} = 0$$

□

Sanger [Sanger, 2004] showed for goal-directed exploration that this condition is also sufficient in the non-linear case with $n = m$. In fact, this condition is very general because it indicates that the action error in equation (4.6) is zero. The learner already fits the data which directly results in a zero gradient. In a linear system with $n = m$, the condition is also necessary because M is a square matrix with full rank. Therefore the right-multiplication with M^T in the proof is reversible and which implies equivalence between partial-left inverse condition and zero gradient. For redundant systems ($n < m$) the condition is not necessary, since left inverse functions do not exist on arbitrary data sets. If, for instance, data is generated entirely within the nullspace of M, different $q_i \neq q_j$ are generated which evaluate to $x_i = x_j = 0$. Such data can not be fitted with any inverse estimate since multiple target-outputs are given for identical input. A more general condition for fixpoints is given by:

Theorem 3 (Necessary fixpoint condition)**.** *If W is a fixpoint of learning equation (4.9), then W is a (partial) right inverse of M on the observed positions x_l, i.e. $MWx_l = x_l \ \forall \ x_l \in D$.*

Proof.

$$
\begin{aligned}
& \frac{\partial E^Q(\mathbb{Q}, W)}{\partial W} = 0 \\
\Rightarrow \quad & M \frac{\partial E^Q(\mathbb{Q}, W)}{\partial W} = 0 \\
\overset{(4.7)}{\Leftrightarrow} \quad & M(WM - \mathbb{1}_m)\mathbb{Q}M^T = (MW - \mathbb{1}_n)M\mathbb{Q}M^T = (MW - \mathbb{1}_n)\mathbb{X} = 0 \\
\Leftrightarrow \quad & MW\mathbb{X} = \mathbb{X} \\
\Leftrightarrow \quad & MWx_l = x_l \ \forall \ x_l \in D
\end{aligned}
$$

□

This theorem states a very important and non-trivial property of example-based learning in linear domains: Learning from examples corresponds to learning a left-inverse function, because the action error evaluates on $g(f(q_l)) - q_l$, but which can not generally succeed because left-inverses do not exist on arbitrary data sets in redundant domains. However, learning is guaranteed to lead to, at least partial, right-inverses for *any* data set, which corresponds to solving the coordination problem. Like theorem 2 this statement becomes an equivalence for $n = m$ (here because the left-multiplication with M is reversible). Both theorems can be summarized by

$$
WMq_l = q_l \forall \ l \quad \Rightarrow \quad \frac{\partial E^Q(W)}{\partial W} = 0 \quad \Rightarrow \quad MWx_l = x_l \forall \ l \ .
$$

Only for $n = m$ these conditions are equivalent. This asymmetry for $n < m$ is the second result on the impact of redundancy, additionally to the gradients losing their strictly positive relation in theorem 1. According to theorem 3, learning from examples will always result in a right inverse solution on the outcomes x_l contained in the data set. If the outcomes do not span the entire space $\mathbf{X}^*$ (if $\mathbb{X}$ does not have full rank), the solution will only be valid in the corresponding subspace.

4.2.1 Plain Goal-Directed Exploration

These fixpoint conditions for general data sets now allow to investigate right inverse learning driven by particular exploration processes. Early approaches to goal-directed exploration have been discussed for the generation of data D in [Sanger, 2004] and [Oyama and Tachi, 2000]. A data set $D_t = \{(x_k^{(t)}, \ q_k^{(t)})\}_k$ is newly generated for each learning step t. The current inverse estimate $g(x^*, W_t)$ is evaluated on $\mathbf{X}^* = \{x_0^*, ..., x_{K-1}^*\}$ to select actions $q_k^{(t)}$:

$$
q_k^{(t)} = g(x_k^*, W_t) = W_t x_k^* \ , \qquad x_k^{(t)} = f(q_k^{(t)}) = MW_t x_k^* \ ,
$$

which corresponds to "trying to reach" by means of the current inverse estimate, and observing the outcome. A learning step according to equation (4.9) is performed

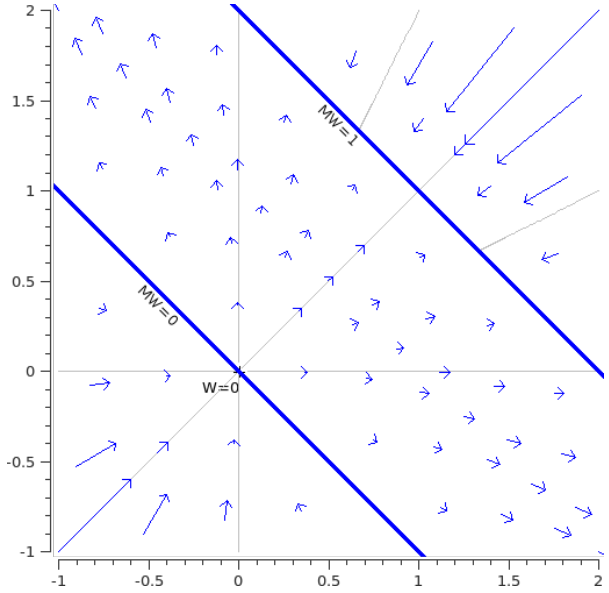


Figure 4.2: The learning gradient $-\frac{\partial \hat{E}_p^Q(W)}{\partial W}$ during plain goal-directed exploration is shown in the parameter space. Right inverse matrices with $MW = 1$ are fixpoints, but the exploration can get stuck in the Nullspace $MW = 0$.

and the process repeats. Using this definition allows to derive the matrices $\mathbb{Q}$ and $\mathbb{X}$ depending on the current value of W:

$$\begin{aligned} \mathbb{Q}[W] &= \sum_{k=0}^{K-1} q_k^{(t)} q_k^{(t)^T} = \sum_{k=0}^{K-1} (W x_k^*)(W x_k^*)^T = W \mathbb{X}^* W^T \ , \quad \text{and} \\ \mathbb{X}[W] &= M \mathbb{Q}[W] M^T \ . \end{aligned}$$

Inserting $\mathbb{Q}$ into the action gradient (equation 4.7) gives the gradient that is followed during plain goal-directed exploration:

$$\frac{\partial \hat{E}_p^Q(W)}{\partial W} = (WM - \mathbb{1}_m)\mathbb{Q}[W]M^T = (WM - \mathbb{1}_m)W\mathbb{X}^* W^T M^T \tag{4.10}$$

Theorem 3 guarantees that fixpoints of the learning equation are partial right inverses. In plain goal-directed exploration the observed matrix $\mathbb{X}$ can generally lose rank, either because W does not have full rank, or because W projects into the nullspace of M:

$$n \geqq rank(W) = rank(\mathbb{Q}[W]) \geqq rank(\mathbb{X}[W])$$

It can be seen directly that all valid right inverses are fixpoints: replacing $MW = \mathbb{1}_n$ in the gradient results in zero. Also, $W = 0$ is a fixpoint in plain goal-directed

exploration, but which is not a valid right inverse of M. This case represents a partial solution in a zero-dimensional subspace as described in theorem 3. Generally, solutions for observed subspaces of $\mathbf{X}$ can occur as fixpoints. Figure 4.2 illustrates learning on this gradient for the example with $M = (0.5, 0.5)$. As predicted by theorem 1, the gradient does never point in the opposite direction of $\partial E^X / \partial W$. In the example, fixpoints lie on $MW = 0$ or $MW = 1$. $MW = 1$ represents the set of correct solutions to the right inverse problem. If the learner is initialized with $MW_0 > 0$, it will converge to such a solution. Otherwise it will stop in $MW = 0$ which is not a solution to the right inverse problem, but only a partial solution for the zero-dimensional subspace $x = 0$. Note that theorem 2 correctly describes the fixpoints $MW = 1$ and $W = 0$. Such values of W will never generate inconsistent data $q_i \neq q_j,\ x_i = x_j$. For the remaining fixpoints, W lies entirely in the Nullspace of M, but is not zero itself. These fixpoints $MW = 0, W \neq 0$, are only described by theorem 3 which gives the necessary condition. Gray lines show exemplary trajectories on which W_t is changed during learning. The directions are entirely concentric, as W_t is moved on straight paths away from $W = 0$ for $0 < MW < 1$, or towards $W = 0$ for $MW < 0$ and $MW > 1$. Hence, W_t never changes its column space which can be shown for the general case by factorizing the gradient:

$$\frac{\partial \hat{E}_p^Q(W)}{\partial W} = W \cdot \underbrace{\left[(MW - \mathbb{1}_n)\mathbb{X}^* W^T M^T\right]}_{P} = W \cdot P \ .$$

The right multiplication with $P \in \mathbb{R}^{n \times n}$ transforms W into the gradient, which means that both have the same column space. Then W has still the same column space after a gradient update.

Plain goal-directed exploration can lead to the discovery of a valid right inverse solution. However, it sticks to only partial solutions if it is not well initialized and can therefore not reliably solve the inversion of causality. The exploration does not allow for an orienting towards new stimuli because it remains in a fixed column space.

4.2.2 Exploratory Noise

Plain goal-directed exploration does not contain exploratory noise, which can result in degenerated data sets within subspaces. The following analysis investigates the impact of such exploratory noise, i.e. exploring actions that do not exactly correspond to the suggestion of the inverse estimate $q = g(x^*, W)$. Exploratory noise can be injected by generating examples with a perturbed variation of inverse estimate[1] g. In the linear case this perturbation can be formulated by choosing actions with some generating matrix W_{gen}, that is a perturbed version of W:

$$q_k^{(t)} = W_{gen}^{(t)} x_k^* \quad \text{with} \quad W_{gen}^{(t)} \sim W_t + \varepsilon \ .$$

[1] The proofs in this section can be derived analogously for additive noise on the actions, such that $q_k^{(t)} = W_t x_k^* + \varepsilon$. The only difference for the analysis is that the term $trace(\mathbb{X}^*)$ in the expected action matrix disappears, but which leaves the full rank and the fixpoints unchanged. The advantage of perturbing g instead will become visible in chapter 5.

The components of the perturbation $\varepsilon \in \mathbb{R}^{m\times n}$ are chosen i.i.d. with zero mean and variance σ^2. Examples for multiple perturbations can be collected and used for one gradient step according to the learning equation (4.9). This analysis assumes that enough data is collected to approximate the learning process by the expectation of this exploration process. The expected action matrix that is generated during such exploration is:

$$\begin{aligned}\mathbb{Q}[W] &= E\left[(W+\varepsilon)\mathbb{X}^*(W+\varepsilon)^T\right]_\varepsilon \\ &= E\left[W\mathbb{X}^*W^T + W\mathbb{X}^*\varepsilon^T + \varepsilon\mathbb{X}^*W^T + \varepsilon\mathbb{X}^*\varepsilon^T\right]_\varepsilon\end{aligned}$$

This gives $E[W\mathbb{X}^*\varepsilon^T]_\varepsilon = E[\varepsilon\mathbb{X}^*W^T]_\varepsilon = 0$ because $E[\varepsilon] = 0$. Expanding the last term gives:

$$E[\varepsilon\mathbb{X}^*\varepsilon^T]_\varepsilon = E\left[\left(\sum_{l=0}^{n-1}(\sum_{k=0}^{n-1}\varepsilon_{i,k}x^*_{k,l})\varepsilon_{j,l}\right)_{i,j}\right]_\varepsilon$$

Since $\varepsilon_{i,k}$ and $\varepsilon_{j,l}$ are by definition un-correlated unless $i=j$ and $k=l$, the expected matrix is diagonal with the components:

$$E\left[\left(\varepsilon\mathbb{X}^*\varepsilon^T\right)_{i,j}\right]_\varepsilon = \begin{cases}\sum_{k=0}^{n-1} x^*_{k,k}\sigma^2 & \text{if } i=j,\\ 0 & \text{else.}\end{cases}$$

Thus, the matrix is scalar with

$$E[\varepsilon\mathbb{X}^*\varepsilon^T]_\varepsilon = trace(\mathbb{X}^*)\sigma^2\mathbb{1}_m\ ,$$

which gives the expected action matrix

$$\mathbb{Q}[W] = W\mathbb{X}^*W^T + trace(\mathbb{X}^*)\sigma^2\mathbb{1}_m\ . \tag{4.11}$$

Unlike the loss of rank in plain goal-directed exploration, this matrix has full rank, which also results in full rank for $\mathbb{X}$:

Proposition 1. *For $\sigma^2 > 0$: $rank(\mathbb{Q}) = m$, $rank(\mathbb{X}) = n$*

Proof. $rank(\mathbb{Q}) = m$: The symmetric form $W\mathbb{X}^*W^T$ in equation 4.11 is positive-semidefinite. The second term is scalar and thus positive-definite for $\sigma^2 > 0$. The sum of a positive-semidefinite and a positive-definite matrix is also positive-definite, which implies full rank.
$rank(\mathbb{X}) = rank(M\mathbb{Q}M^T) = n$ then follows from basic linear algebra. □

The full rank of $\mathbb{X}$ implies that all fixpoints of the learning equation are valid right inverse functions:

Proposition 2. *For $\sigma^2 > 0$, any fixpoint W of the learning equation (4.9) is a right inverse of M.*

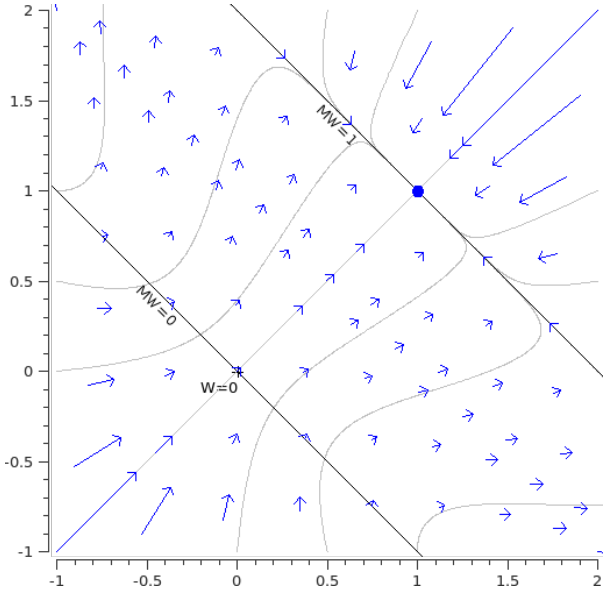


Figure 4.3: Learning gradient $-\frac{\partial \hat{E}^Q}{\partial W}(W)$ for noise $\sigma^2 = 0.2$ The introduction of exploratory noise lets the inverse estimate converge to the Moore-Penrose pseudoinverse.

Proof. Theorem 3 states that $MW\mathbb{X} = \mathbb{X}$ for any fixpoint. Since $\mathbb{X}$ has full rank it is possible to right-multiply with $\mathbb{X}^{-1}$ which gives:

$$MW\mathbb{X} = \mathbb{X} \quad \Rightarrow \quad MW = \mathbb{1}_n$$

□

Hence, the introduction of exploratory noise remedies the flaw of plain goal-directed exploration to result in only partial solutions. For a full analysis, $\mathbb{Q}[W]$ can be inserted into the gradient equation (4.7) in order to obtain the gradient on which learning proceeds:

$$\frac{\partial \hat{E}^Q(W)}{\partial W} = (WM - \mathbb{1}_m)(W\mathbb{X}^*W^T + trace(\mathbb{X}^*)\sigma^2\mathbb{1}_m)M^T.$$

Using this equation, it can be shown that the exploration does not only yield valid right inverse function, but results in a unique fixpoint even in redundant domains:

Theorem 4. *For $\sigma^2 > 0$, the unique fixpoint of the learning equation (4.9) is the Moore-Penrose pseudoinverse: $W = M^{\#} = M^T(MM^T)^{-1}$.*

Proof. The pseudoinverse can be derived from the fixpoint equation by utilizing the

previous result $MW = \mathbb{1}_n$. Expanding the gradient first gives for $\alpha = trace(\mathbb{X}^*)\sigma^2 > 0$:

$$\begin{aligned} 0 = \frac{\partial \hat{E}^Q}{\partial W} &= WMW\mathbb{X}^*W^TM^T + WM\alpha\mathbb{1}_mM^T \\ &\quad - W\mathbb{X}^*W^TM^T - \alpha\mathbb{1}_mM^T \end{aligned}$$

Substituting MW with $\mathbb{1}_n$ gives

$$\begin{aligned} & W\mathbb{X}^* + \alpha WMM^T - W\mathbb{X}^* - \alpha M^T = \alpha WMM^T - \alpha M^T = 0 \\ \Leftrightarrow\ & WMM^T = M^T \\ \Leftrightarrow\ & W = M^T(MM^T)^{-1} \end{aligned}$$

□

Figure 4.3 illustrates the learning gradient with exploratory noise $\sigma^2 = 0.2$. The qualitative behavior is drastically changed compared to exploration without noise (figure 4.2). Noise removes the erroneous fixpoints on $MW = 0$. The gradient is not concentric around $W = 0$ anymore and allows W to change the column space. On the solution manifold $MW = 1$ the gradient pulls W towards the pseudoinverse, which is $W = M^{\#} = (1.0, 1.0)^T$ in the example.

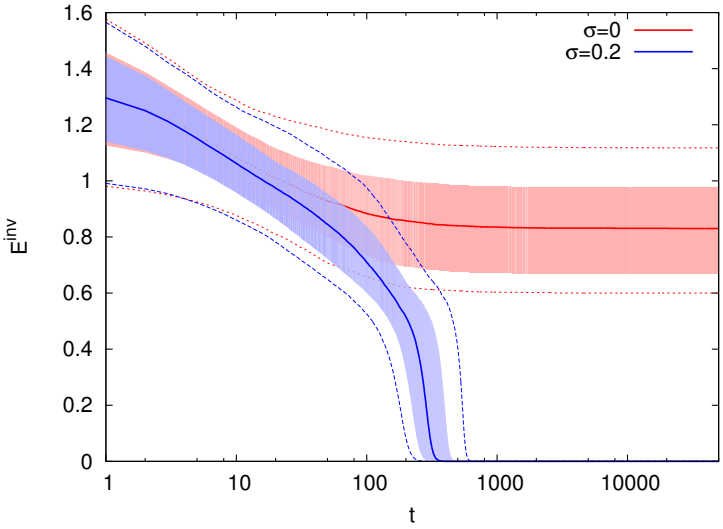
4.3 Numeric simulation results

In order to illustrate the theoretical findings, this section provides numerical simulation results for goal-directed exploration with and without exploratory noise. Forward functions are chosen with random forward matrices M in $n = 3$ and $m = 10$ dimensions. The components of the forward matrices were drawn from a normal distribution $\mathcal{N}(0.0, 1.0)$ and then normalized to $||M||_2 = 1.0$, where $||\cdot||_2$ is the l^2 norm. 50 different forward matrices were chosen, each with 50 different random initializations of W, also drawn component-wise from $\mathcal{N}(0.0, 1.0)$. Plain goal-directed exploration ($\sigma = 0.0$) was performed with a step-width $\eta = 0.2$. Goal-directed exploration with noise was simulated with $\eta = 0.2$, $\sigma = 0.2$, and 30 random perturbations $\varepsilon \sim \mathcal{N}(0.0, \sigma^2)$ within each learning step. Both setups were simulated with three target positions $x^* \in \{(1,0,0)^T, (0,1,0)^T, (0,0,1)^T\}$.

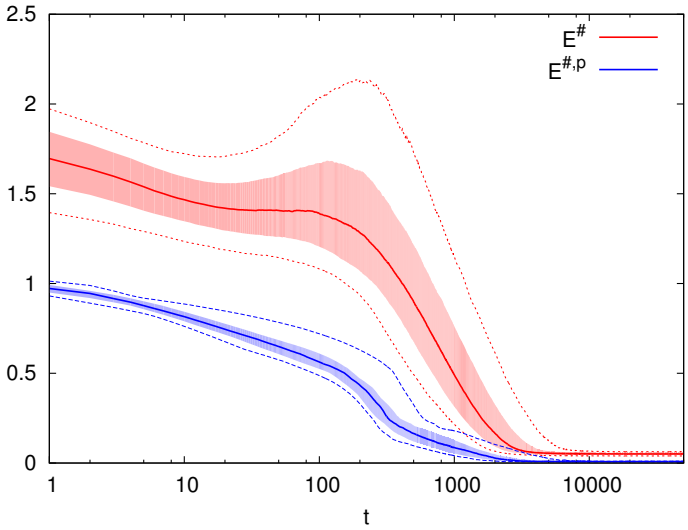
Figure 4.4(a) shows how the distance from the right-inverse condition $MW = \mathbb{1}_n$

$$E_t^{inv} = ||MW_t - \mathbb{1}_n||_2 \ / \ ||\mathbb{1}_n||_2 \tag{4.12}$$

develops over time. Exploration with noise solves the problem exactly in all trials: the error reaches numeric limits ($1.2 \cdot 10^{-16} \pm 5 \cdot 10^{-17}$) after approximately $t = 3200$. Without noise the error can not be reduced to zero. Only few individual trials (below the 10% quantile) reach $E^{inv} \approx 0.0$, which incidentally hit a good initialization. Most trials get stuck in partial solutions.



(a) Deviation from $MW = \mathbb{1}_n$ over time.



(b) Deviation from $W = M^{\#}$ over time (for $\sigma^2 = 0.2$).

Figure 4.4: Correct solutions with $MW = \mathbb{1}_n$ are only found with noise ($\sigma^2 = 0.2$), whereas W approaches the pseudoinverse. Bold lines show median values, thin lines the 10% and 90% quantiles and the filled areas correspond to the range between the 25% and 75% quantiles.

In order to illustrate theorem 4, the deviation $E^{\#}$ from the Moore-Penrose pseudoinverse was measured (for the $\sigma = 0.2$ setup only):

$$E_t^{\#} = ||W_t - M^{\#}||_2 \ / \ ||M^{\#}||_2 \tag{4.13}$$

$$E_t^{\#,p} = ||E[W_t|M] - M^{\#}||_2 \ / \ ||M^{\#}||_2 \tag{4.14}$$

The measure $E^{\#,p}$ indicates how well the population average of W (for a fixed M) fits the pseudoinverse. Figure 4.4(b) shows how both measures develop over time. The population-average measure decreases continuously and reaches a value of $8.2 \cdot 10^{-3} \pm 1.4 \cdot 10^{-3}$, which corresponds to a very good fit of the 30 matrix entries to the pseudoinverse. Not all trials show a monotonic behavior in $E_t^{\#}$ as indicated by the temporary increase of the quantiles above 50% around $t = 200$. In some regions of the parameter space the inverse estimates initially spread into different directions, before they come close to the solution manifold ($MW = \mathbb{1}_n$) and move towards the pseudo-inverse. This behavior is well visible in the upper left, and lower right corner of Figure 4.3. The error development shows that between $t = 1000$ and $t = 10000$, $E^{\#}$ is still decreasing significantly while E^{inv} is already very close to zero, which displays the optimization of "redundant" parameters inside the set of correct solutions $MW = \mathbb{1}_n$. Note that theorem 4 shows the fixpoint for the *expected* gradient. For a finite number of samples there remain small random movements on $MW = \mathbb{1}_n$ around $M^{\#}$ that prevent $E^{\#}$ from decreasing exactly to zero. For $n = 3$ and $m = 10$, this solution manifold has 21 dimensions in the 30 dimensional parameter space, since only 9 dimensions are bound by $MW = \mathbb{1}_3$. In the experiments $E^{\#}$ nevertheless reaches a low level of $5 \cdot 10^{-2} \pm 1 \cdot 10^{-2}$.

4.4 Discussion

This chapter has investigated the theoretical basis of learning inverse models from examples. When the forward function is linear, learning can generally proceed from arbitrary data sets because performance gradient and action gradient have a non-negative angle. The fixpoint analysis shows that learning inverse models from examples leads, in linear domains, always to partial right-inverse functions which solve the coordination problem at least for those observations in the data set. If the data set does not span the entire observation space, the learned inverse models are only valid in the corresponding subspace, which leads to the failure of plain goal-directed exploration. Sanger already described a failure mode in which the learner already fits the self-generated data [Sanger, 2004]. This analysis is complemented by the analysis of redundancy in this chapter which shows that plain goal-directed exploration can also get stuck in the Nullspace of the forward function, where data can not be fitted due to different target outputs for identical input of the learner. Contrary to such negative results, the analysis of exploratory noise provides the first affirmative results on goal-directed exploration: if noise is added, the exploration does not only succeed in inverting the causal relation between actions and outcomes, but results in the unique

Moore-Penrose pseudoinverse, which is the least-squares solution.

Albeit linear domains are considerably simpler than non-linear ones, they allow to study the relation between left- and right-inverses, and in particular the effect of redundancy, which explicitly includes high-dimensional action spaces. The analysis showed that redundancy weakens the relation between performance and action gradient, which is strictly positive without redundancy, but only non-negative when linear redundancy is present, plus it leads to additional failure modes when plain goal-directed causes degenerated data sets. However, redundancy does not impair the functioning of goal-directed exploration when exploratory noise is included, which is an important results for the overall scope of this thesis to consider the bootstrapping of coordination skills in high-dimensional domains. In linear domains it is not strictly necessary to perform exploration in a goal-directed manner, since learning can be successful from arbitrary – also random – data sets. This situation changes when non-convex solution sets occur in non-linear domains. The next chapter investigates this case on the basis of goal-directed exploration as presented in this linear-case analysis.

Chapter 5

Coordination Problems with Non-Convex Solution Sets

The analysis of linear domains shows that goal babbling can solve the inversion of causality and that the presence of multiple solutions for the same goal still allows for successful learning when the redundancy manifolds are linear subspaces. Many practically relevant coordination problems, however, are non-linear and have solution sets that are not convex. This prohibits learning from arbitrary data sets [Jordan and Rumelhart, 1992], because averaging of examples with the same outcome can lead to actions with a different outcome (see section 2.4). Prior to the work described in thesis, no method was known to learn inverse models in the presence of non-convex solution sets. This chapter investigates how goal babbling allows to deal with that problem and provides the first algorithm that can learn inverse models from examples, even in the presence of non-convex solution sets. The methods and results presented in this chapter have been published in [Rolf et al., 2010c].

Plain Goal-Directed Exploration: revisited

As a starting point to understand goal babbling in domains with non-convex solution sets, figure 5.1 shows the failure of plain goal-directed exploration [Oyama and Tachi, 2000, Sanger, 2004] for the toy-example of inverse kinematics as illustrated in section 2.1. The figure shows how the inverse estimate changes over the course of learning in the action space. Examples (x_k, q_k) (green dots) are iteratively generated by querying the inverse estimate with different goals x_k^*. An inverse model that solves the coordination problem would place all colored markers on the solution set with the corresponding color, which is clearly not achieved by plain goal-directed exploration. Of course, this scheme can not solve the inversion of causality, as analyzed in the previous chapter, but it allows a concise view on two additional problems in non-linear redundant domains: Firstly, the exploration scheme is exposed to arbitrary drifts through the action space. In figure 5.1(d) the inverse estimate has already approached the boundary of the action space and afterwards completely degenerates into the limits. Secondly, the exploration scheme generates inconsistent examples *although* the exploration is clearly not exhaustive. The inverse estimate crosses the non-convex solution sets multiple times (for instance the redly colored manifolds in figure 5.1(b)-(d)) so that averaging leads to erroneous results. Section 5.1 presents an in-depth analysis of this problem and proposes a solution scheme. Based on that scheme, section 5.2

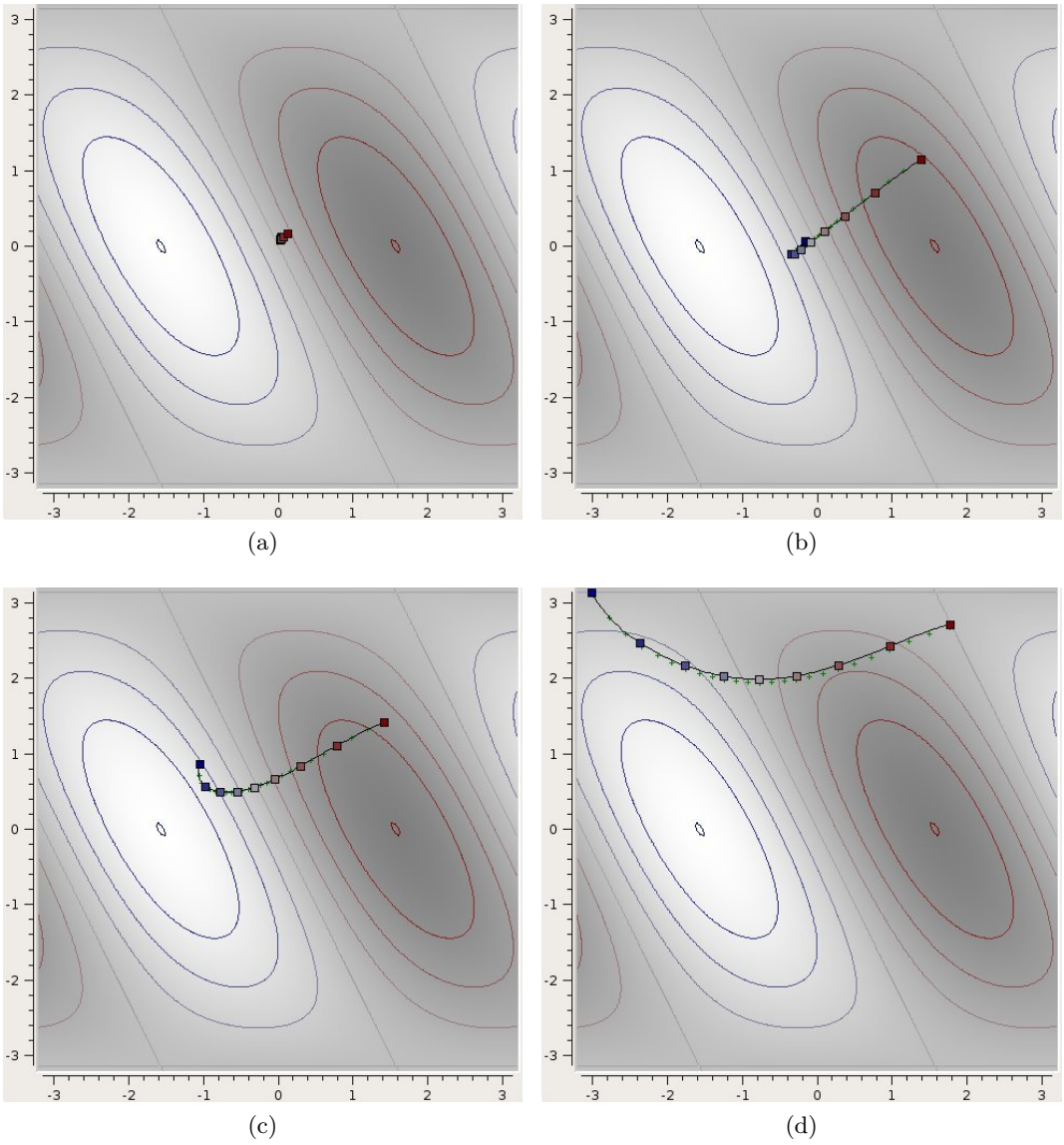


Figure 5.1: Learning dynamics with plain goal-directed exploration from (a) to (d). Learning occurs from inconsistencies, as the controlled manifold intersects some redundancy manifolds multiple times. The estimate drifts in its orthogonal direction, where no training data is available.

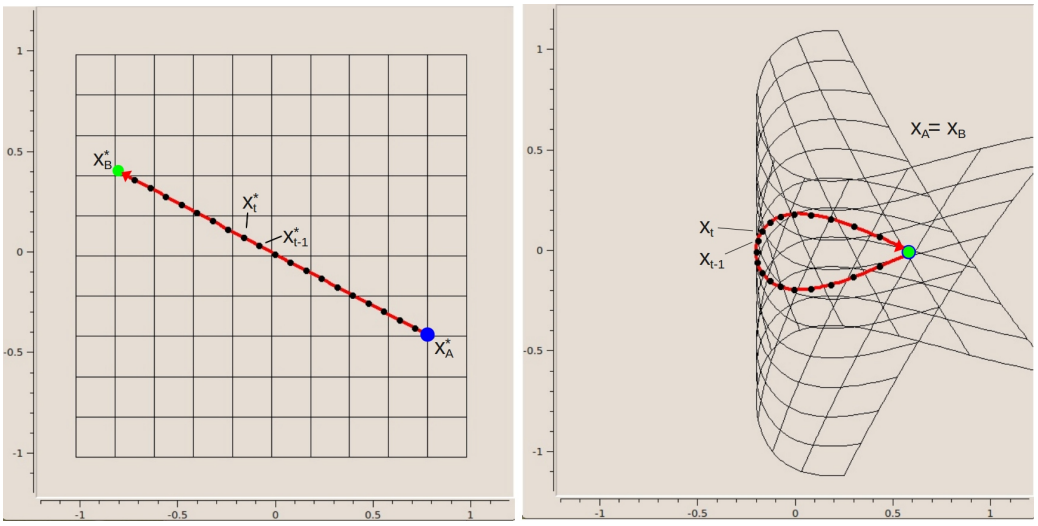


Figure 5.2: (Left) Space of target positions x^*. A linear target motion shall be produced between two targets. (Right) Space of results $x = f(g(x^*))$. An inconsistency occurs e.g. when the grid is folded. The formerly straight line now has a circular shape.

formulates an algorithm that re-integrates exploratory noise to solve the inversion of causality, and a mechanism to prevent arbitrary drifts.

5.1 Non-convex Solution Sets during Goal Babbling

Two examples (x_A, q_A) and (x_B, q_B) are inconsistent, if they represent the same outcome $x_A = x_B$ but different actions $q_A \neq q_B$ which can not be averaged without provoking a different outcome $(f(\alpha \cdot q_A + (1-\alpha) \cdot q_B) \neq x_A)$. Regardless of the kind of exploration that is used to generate samples, two examples with exact same effector pose will rarely be found. Resolving inconsistencies solely based on the samples is therefore hardly possible. The central step to understand and resolve inconsistencies is to consider the example generation method itself, instead of considering isolated examples.

Structure of Inconsistencies Generally, an exploration mechanism can denoted to generate examples from some set of actions $\mathbf{Q}^{expl} \subseteq \mathbf{Q}$. For random motor babbling this set is $\mathbf{Q}^{expl} = \mathbf{Q}$. Goal-directed exploration, however, generates examples in a very structured manner by exploring only actions that suggests by the inverse estimate, such that $\mathbf{Q}^{expl} = g(\mathbf{X}^*)$. Even in that case, inconsistent examples can be generated, but the structure of goal-directed exploration restricts the number of ways inconsistencies can occur: Assume that two inconsistent examples $q_A \neq q_B, x_A = x_B$ are generated in the goal-directed exploration $(q_A, q_B \in \mathbf{Q}^{expl} = g(\mathbf{X}^*))$. An important, first observation

is that these two examples must have been generated by two different goals $x_A^* \neq x_B^*$. If the goals were identical, then also the actions would be the same, since $x_A^* = x_B^* \Rightarrow g(x_A^*) = g(x_B^*)$. Hence, inconsistent examples can only be generated if two different goals lead to the observation of the same outcome. This allows to utilize goals as a *reference structure* which can be compared to the actually observed outcomes. A hypothetical example is illustrated in figure 5.2: a set of two-dimensional goals is arranged in a grid-structure. The right side shows how that grid is deformed by performing a goal-directed exploration step $g(x^*)$ and observing the outcome $f(g(x^*))$ for all of the goals. Inconsistencies can occur when different goals $x_A^* \neq x_B^*$ evaluate to the same outcome $x_A = x_B$. This implies that the corresponding grid of outcomes must have a certain degree of *self-overlap*, such as a folding in the example illustration. It is hardly possible to detect this case solely based on the isolated examples (e.g. the grid's vertices). In particular, exact matches $x_A = x_B$ do not necessarily occur on finite data sets. Yet, the self-overlap becomes clearly visible if the two-dimensional topological grid-structure (also the grid's edges which indicate topological vicinity) of the goals is considered. Such topological distortions can be detected by first generating examples for all goals on such a grid, and then checking for crossings of the grid's edges – or hyper-edges in more than two dimensions. This, however, would require to analyze the entire set of goals at once in a computationally rather expensive operation. This thesis proposes a simpler scheme that is based on executing and analyzing *continuous paths* of actions, which are performed between different goals. This corresponds to the physical act of reaching, which generally requires the execution of physical movement paths. Moreover, considering paths paves the way towards an online learning scheme that is introduced in chapter 6.

Suppose, the inverse estimate is used to attempt a linear target movement between x_A^* and x_B^* (see figure 5.2, left), i.e. to perform "trying to reach" from x_A^* to x_B^*. The system starts with an action q_A, corresponding to x_A^*. Subsequently, targets x_t^* are interpolated between x_A^* and x_B^* and each of them results in an examples $q_t = g(x_t^*), x_t = f(q_t)$ so that the final action is q_B. At the beginning and the end of the movement, the same outcome $x_A = x_B$ is observed. When the intermediate outcomes x_t are observed while trying to follow that straight path, two cases can occur:

1. The observed outcomes change while using the inverse estimate $g(x^*)$ to follow the goals between x_A^* and x_B^*. Since the observation returns to the same outcome $x_A = x_B$, the intermediate movement must have a closed shape (see figure 5.2, right). The goal is to follow a straight line, i.e. to keep the movement direction constant, but the observed movement direction changes.

2. The observed outcome remains constant, in spite of different actions between q_A and q_B. This case can occur when the inverse estimate moves exactly along one redundancy manifold. In the case of reaching, this means that the robot is moving its joints, but the effector position does not change, which is characterized by a minimum of movement efficiency.

Hence, inconsistencies during goal-directed exploration occur only if either (*i*) unintended changes of movement direction or (*ii*) movements with zero efficiency are present, which both can be detected from observation of the movement.

Path-based Inconsistency Resolution The general idea how to use these insights about the structure of inconsistencies is to assign weights $w_t \in \mathbb{R}$ for each example (x_t, q_t) along a continuous movement path. Unintended changes of movement direction can be tackled with a scheme that bases upon a special case: If the observed movement direction never deviates by 90° or more from the intended movement direction, circular shapes as shown in figure 5.2 can not occur. This fact is utilized in the following weighting scheme:

$$w_t^{dir} = \frac{1}{2}\left(1 + cos\sphericalangle(x_t^* - x_{t-1}^*,\ x_t - x_{t-1})\right). \tag{5.1}$$

Thereby $\sphericalangle(x_t^* - x_{t-1}^*, x_t - x_{t-1})$ is the angle between the intended and actual movement direction of the effector. If both are identical the angle is 0.0° and the weight becomes $w_t^{dir} = 1.0$. If the observed movement has the exact opposite direction, the angle is 180.0° and the weight becomes $w_t^{dir} = 0.0$. If a circular motion occurs for a linear target motion, one half of the motion receives a higher weight than the other one and the inconsistency can be broken. If the estimate $g(x^*)$ is rather accurate and the intended movement direction can always be realized, all samples receive full weight 1.0.

The second case of an inconsistency (low movement efficiency) can be resolved by weighting with the ratio of effector motion and joint motion, which becomes 0.0 if the joints move without effector motion:

$$w_t^{eff} = \frac{||x_t - x_{t-1}||}{||q_t - q_{t-1}||} \quad . \tag{5.2}$$

Since both weights are necessary for inconsistency resolution, they are multiplied such that an example is ignored if any of the two criteria assigns a weight zero:

$$w_t = w_t^{dir} \cdot w_t^{eff} \ . \tag{5.3}$$

The weighting scheme relies on the temporal order of examples along the trajectory, since the actual and the last sample is taken into account. In particular, it relies on goals: unintended changes of movement direction can only be detected if there is an intended direction. The path based exploration generates an n-dimensional manifold within the joint space, where the information about continuity along this manifold allows for evaluation of the movement directions. It is this very information structure that allows for a resolution of inconsistencies and distinguishes the proposed scheme from all previous ones. The rules are local in space and time, since only the immediate temporal and spatial context is considered. Therefore both rules are imperfect, since only one movement direction can be observed at a time. However, the experiments in this chapter show that the rules are sufficient to resolve inconsistencies.

5.2 Structured Variation and Regularization

The overall algorithm to learn inverse models by means of goal babbling is organized in epochs. In each epoch, the first step is to select a sequence of goals x_t^*. In order to apply the weighting scheme as described above, this sequence consists of piecewise linear movements. At first, a sequence of $k = 1 \dots K$ goals $x_{k \cdot L}^*$ is randomly chosen from the set of goals $\mathbf{X}^*$. Then, successive goals are connected by a linear path with $l = 1 \dots L$ intermediate goals:

$$x_{k \cdot L + l}^* = \frac{L - l}{L} \cdot x_{k \cdot L}^* + \frac{l}{L} \cdot x_{(k+1) \cdot L}^*, \tag{5.4}$$

This scheme generates $K \cdot L$ temporally ordered goals x_t^* which cover the set of goals $\mathbf{X}^*$ and are the starting point for goal directed exploration.

Structured Variation for Efficient Exploration Chapter 4 has pointed out the importance of exploratory noise. The most simple possible way to insert such noise is to add i.i.d. noise (for instance Gaussian noise) to each action q_t before executing it. However, such noise would affect the information structure that is necessary to resolve inconsistent examples. Moreover, it is not very practical for a physical execution of reaching movements, which would be erratic and jerky. In order to obtain a continuous mechanism for exploratory noise that allows to apply the weight-based inconsistency resolution, this thesis proposes to distort the entire inverse estimate, instead of isolated actions. In the linear case analysis in chapter 4 this has been achieved by adding a random term to the matrix W inside the inverse estimate. In the general, non-linear case this can be formulated by adding a small perturbation term $E^v(x^*)$ to the inverse estimate:

$$g^v(x^*) = g(x^*) + E^v(x^*). \tag{5.5}$$

Examples are then generated with this *structured variation* instead of the actual inverse estimate: $q_t^v = g^v(x_t^*)$, $x_t^v = f(q_t^v)$. The set of examples generated for a variation v is denoted as

$$D^v = \{ \ (\ f(g^v(x_t^*)), \ g^v(x_t^*) \ ) \ \}_t. \tag{5.6}$$

The assumptions and arguments for the inconsistency resolution still hold, since $g^v(x^*)$ is again a function and spans a n dimensional manifold in the action space along the respective path. For a given set of examples D^v, the weighting scheme can be applied as proposed above. The index v is added to identify weights for examples of a specific variation:

$$w_t^{v^{dir}} = \frac{1}{2} \left(1 + cos\sphericalangle(x_t^* - x_{t-1}^*, \ x_t^v - x_{t-1}^v)\right), \tag{5.7}$$

$$w_t^{v^{eff}} = \frac{||x_t^v - x_{t-1}^v||}{||q_t^v - q_{t-1}^v||}, \tag{5.8}$$

$$w_t^v = w_t^{v^{dir}} \cdot w_t^{v^{eff}}. \tag{5.9}$$

Home Postures for Regularization Structured variations and the weighting scheme allow to solve the inversion of causality and to deal with the possible generation of inconsistent examples. Another problem that is specific to plain goal-directed exploration is that the inverse estimate is exposed to uncontrolled drift (see figure 5.1), which can degenerate the performance of the inverse estimate. A developmentally plausible, and technically feasible way to prevent such drifts has been proposed independently in [Rolf et al., 2010c] and [Baranes and Oudeyer, 2010a]: Instead of permanently performing goal-directed movements, the learner returns to a "rest" or "home" position after some time, which corresponds to executing some action q^{home}. For biological motor coordination this could, for instance, correspond to relaxing all muscles and resting for a while. The central insight for the learning of inverse models is that such an action can not only be used as a starting point for goal-directed movements, but it can be directly used for learning. For an initial algorithm formulation, this idea is integrated in the most simple possible way: In each epoch, the example $q_0^v = q^{home}, x_0^v = f(q^{home}) = x^{home}$ is added to the data set generated with goal-directed exploration:

$$D^v \longleftarrow \{(f(q^{home}), q^{home})\} \cup D^v \tag{5.10}$$

The "home" example receives the full weight $w_0^v = 1.0$.

A home position is a stable point in exploration, and thus in learning. The inverse estimate will generally tend to reproduce the connection between q^{home} and x^{home} if it is used for learning: $g(x^{home}) \approx q^{home}$. The easiest way to achieve the result of applying the home posture is: applying the home posture. This stable point largely prevents the inverse estimate to drift away. Learning can start around the home posture and proceed to other targets.

Exploration and Learning Algorithm In each epoch, example data (and corresponding weights) from multiple different variations $g^v(x^*), v = 1...V$ is combined for learning, where $V \in \mathbb{N}$ is the number of different variations. The complete set of examples is then

$$\begin{aligned} D &= (x^{home}, q^{home}) \cup \bigcup_v D^v \\ &= (x^{home}, q^{home}) \cup \bigcup_v \{ \left(f(g^v(x_t^*)),\ g^v(x_t^*) \right) \}_{t=0...T} . \end{aligned}$$

The multiple variations allow to discover new, relevant action by chance which solves the problem of plain goal-directed exploration to reliably invert causality. All directions in the action space are locally covered if the number of variation V exceeds the action dimension m.

In the learning step, the parameters θ of the inverse estimate $g(x^*, \theta)$ are updated using the generated examples $(x_t^v, q_t^v), t = 0...T$ (including the home posture) and

Algorithm 1 Goal babbling pseudocode for non-linear domains

Require: Forward function: $f(q)$
Require: Home posture q^{home}
Require: Set of target positions: $\mathbf{X}^*$
  Initialize learner: $\theta \leftarrow \theta_0$, $g(x^*) \leftarrow g(x^*, \theta)$
  for Number of epochs **do**
    Select target sequence from $\mathbf{X}^*$: $x_t^*,\ t = 1 \dots T$ (Equation 5.4)
    $D \leftarrow \varnothing$
    for $v = 1 \dots V$ **do**
      Select disturbance term: $E^v(x^*)$
      Get variation: $g^v(x^*) = g(x^*) + E^v(x^*)$
      Generate examples: $D^v \leftarrow \{ \left(f(g^v(x_t^*)),\ g^v(x_t^*) \right) \}_t$
      Compute weights w_t^v (Equations 5.7, 5.8 and 5.9)
      Add home posture: $D^v \leftarrow D^v \cup (f(q^{home}), q^{home})$
      $D \leftarrow D \bigcup D^v$
    end for
    Reduce error $E_w^Q(\theta)$ on D using gradient descent
  end for

weights w_t^v in a regression step to reduce the weighted action error

$$E_w^Q(D, \theta) = \sum_v \sum_t w_t^v \cdot ||g(x_t^v, \theta) - q_t^v||^2. \tag{5.11}$$

Any regression algorithm can be used for this step (e.g. linear regression schemes).

The overall procedure works in epochs. The inverse estimate is initialized with some parameters θ. The experiments in this chapter use a random initialization such that the inverse estimate generates actions closely around the home posture for all goals. Within one epoch, examples are generated from multiple variations, weights are assigned and the learning is performed with the examples. The next epoch repeats the procedure with the updated inverse estimate. The entire procedure is also detailed in algorithm 1.

The introduction of multiple variations in the exploration locally adds multiple solutions. However, if the disturbance terms $E^v(x)$ have numerically small values, these solutions are located in a small region in the joint space. Therefore the error induced by the non-convexity problem is generally very small and can safely be neglected. The weighting based on intended movement directions prevents learning from significantly inconsistent examples. The efficiency weighting allows to "select" examples generated by different variations. Solutions with higher movement efficiency (see equation 5.2) will receive a higher weight and therefore dominate the learning [Peters and Schaal, 2008]. This causes the inverse estimate to be aligned along such optimal configurations. The averaging is therefore constructive (compared to the destructive averaging in motor babbling) which is only possible due to the combination of variation *and*

weighting. In fact, striving for such optimal movement efficiency is not a luxury for the learning of inverse models. It is necessary to resolve inconsistent solutions and guide the exploration systematically towards new targets.

5.3 Examples

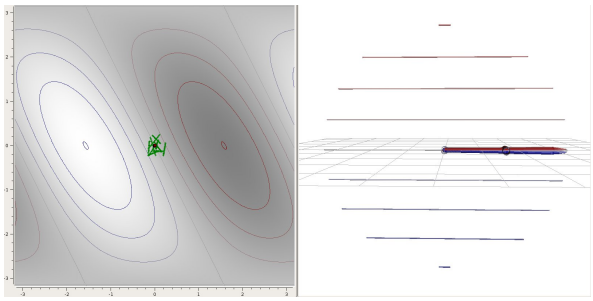
Learning Dynamics An example of inverse kinematics learning with this goal babbling algorithm for the minimal 2-DOF problem (see figure 2.2) is shown in figure 5.3. The inverse estimate is initialized in a small region around the home posture, which is set to $q^{home} = (0.0, 0.0)$. The next images show the progress of the method after several epochs. Each image shows the current inverse estimate together with the currently generated example data. The aim is to control the effector's height within the full range from -1.0m to 1.0m. Initially, only heights around $f(q^{home}) = 0m$ are reachable. Target positions between the extremes -1.0m and 1.0m are tried to reach from the very beginning, although these attempts are not successful at first.

Three qualitative stages can be observed in the progress of bootstrapping the inverse kinematics. These stages are not preprogrammed, but they arise naturally from the learning dynamics. In the *first stage* (orientation), the manifold spanned by the inverse estimate is still close to the home posture. Only a small set of effector poses x can be observed, such that the weighted action error E_w^Q is rather small (similar to the case of a constant function $g(x^*)$). Triggered by the exploration of variations, the inverse estimate starts to align with the correct movement directions and for optimal movement efficiency. Thereby the weights of the examples slowly increase.

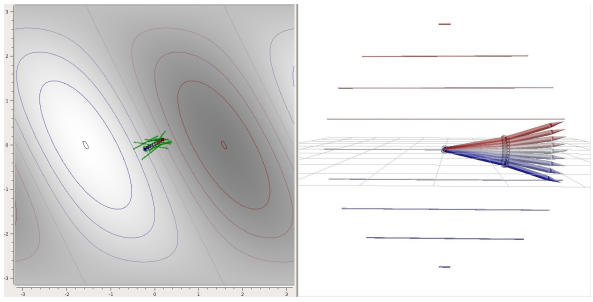
Once the inverse estimate is aligned with optimal directions, the *second stage* (expansion) can be observed. The extrapolation of the inverse estimate causes a rapid expansion of the inverse estimate in the joint space. This stage is characterized by a rapid decrease of the performance error. Due to the efficiency weighting, the expansion approximately follows the steepest, most efficient direction. The inverse estimate is aligned nearly orthogonal to the redundancy manifolds.

The expansion saturates when the ridge of the forward kinematics is hit. More expansion would not discover new effector positions, but only introduce more inconsistencies since the same redundancy manifolds would be crossed again. Examples generated beyond the ridge are, however, filtered out by the weighting of correct movement directions (equation 5.7). Then the *third stage* (tuning) can be observed. The inverse estimate finds the non-linearities that are necessary to reach for the extreme positions and to further optimize the movement efficiency. The performance error decreases slowly until it converges.

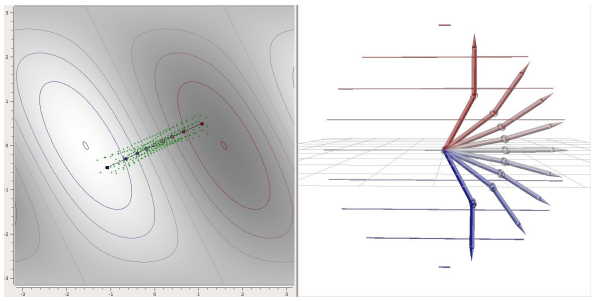
In particular, this example of learning inverse models visualizes how goal babbling differs from exhaustive exploration. The method never samples the entire two-dimensional actions space. Instead, it explores the local surrounding of the inverse estimate, which has a one-dimensional structure since $n = 1$. The same image can be drawn for higher dimensions: even if the same task is learned with more degrees of freedom, the exploration will only explore locally around a $n = 1$ dimensional man-



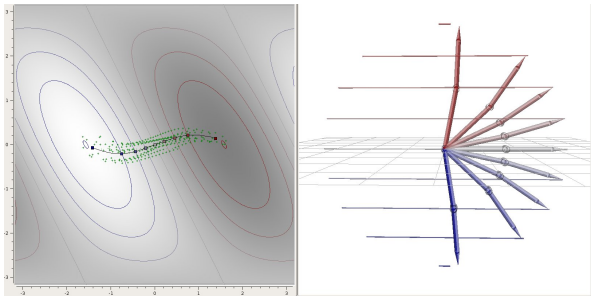
(a) The inverse estimate is initialized around the home posture.



(b) Orientation: the inverse estimate has aligned with the steepest direction.

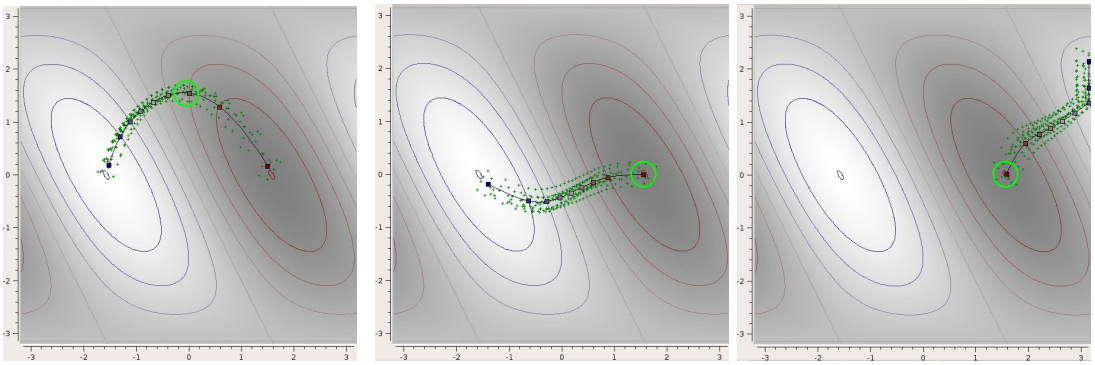


(c) Expansion: the performance error decreases rapidly.



(d) Tuning: the inverse estimate finds the necessary non-linearities to reach for extreme positions.

Figure 5.3: The inverse estimate is initialized around zero in joint space. During goal babbling it unfolds successively and reaches an accurate solution.



(a) Outcome for $q^{home} = (0.0, \frac{\pi}{2})$ (b) Two possible outcomes for $q^{home} = (\frac{\pi}{2}, 0.0)$, which is a singularity of the forward kinematics.

Figure 5.4: The inverse estimate can be shaped by the choice of the home posture, which is shown as green circle. Learning is still possible from a singularity as start point. However, learning can no longer proceed if the joint limits are hit.

ifold in the arbitrarily high-dimensional action space. Redundant choices of action are systematically ignored in order to learn one coherent solution to the coordination problem.

Influence of the Home Posture The home posture is an open parameter of the exploration procedure, which can be used to *shape the inverse estimate*, and to control which way is used to resolve the redundancy. The goal babbling algorithm works robustly for a wide range of home postures. An example of a different home posture is shown in figure 5.4(a). The inverse estimate aligns with the optimal efficient movement direction with respect to the home posture, which acts as origin. The algorithm can also be successful, if the home posture is placed in a singularity, as shown in figure 5.4(b). In that case multiple ways exist to leave the singularity with optimal movement efficiency (two in the example). This symmetry is broken by the randomized exploration of variations. The learning can get stuck if the inverse estimate hits the joint limits, such that a further local improvement of the inverse estimate is not possible, see figure 5.4(b). Goal Babbling shares this problem with feedback-error learning and learning with distal teacher. All three approaches operate iteratively, based on local improvements. Although error-based methods do not make explicit use of a home posture, they require an initial placement of the inverse estimate and cannot proceed if local improvements are not possible. Home postures that cause such ill-posedness are, however, biologically not plausible for systems that need to bootstrap their motor repertoire. Also, they are easy to avoid engineering-wise by choosing a position in the center of the action space and nearby the target positions that are tried to reach.

5.4 Experiments

In this section, results of the goal babbling algorithm for reaching with different robot morphologies are shown. The experiments start by extending the simple 2-DOF arm (see figure 2.2) by more degrees of freedom and finish with goal babbling on a humanoid morphology. All experiments use polynomial regression [Poggio and Girosi, 1990] to represent the inverse estimate $g(x^*, \theta)$. The input vector $x^* \in \mathbb{R}^n$ is expanded by a feature mapping $\Phi^P(x^*) \in \mathbb{R}^p$ which calculates all polynomial terms of the entries of x^*. Thereby P is the maximum degree of the polynomial terms and p is the number of polynomial terms that can be calculated from an n dimensional vector. For a two dimensional input vector $x = (x_{(1)}, x_{(2)})$ and a polynomial degree $P = 2$, $\Phi^P(x)$ calculates the terms $(1.0, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)} \cdot x_{(2)}, x_{(2)}^2)^T \in \mathbb{R}^6$. A linear regression with parameters $\theta = \mathbf{W}$ operates on these features:

$$g(x^*, \mathbf{W}) = \mathbf{W} \cdot \Phi^P(x^*), \quad \mathbf{W} \in \mathbb{R}^{p \times m}. \tag{5.12}$$

The entries of the regression matrix $\mathbf{W}$ are adapted by gradient descent in the weighted action error as defined in equation (5.11). The learning rate is set to 0.2. Before exploration and learning proceed, $\mathbf{W}$ is set to zero and few random adaptions are performed such that $g(x^*, \mathbf{W})$ produces joint angles in a range of 0.1 radian around the home posture.

Linear perturbation terms are used for exploration:

$$E^v(x^*) = \mathbf{A} \cdot x^* + b, \quad \mathbf{A} \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m. \tag{5.13}$$

The values of $\mathbf{A}$ and b are chosen randomly, such that the perturbation of any joint-angle never exceeds a range R within the bounded set of target positions $\mathbf{X}^*$:

$$E^v(x) = (e_1, \ldots, e_m)^T, \quad |e_i| <= R \quad \forall\ i = 1 \ldots m, x \in \mathbf{X}^*.$$

The *mean euclidean deviation* D^X is used as error measure in order to assess the accuracy of the inverse models on the set of goals $\mathbf{X}^*$:

$$D^X(\mathbf{X}^*, \theta) = D^X(\mathbf{X}^*, \mathbf{W}) = \frac{1}{K} \sum_{k=0}^{k<K} ||f(g(x_k^*, \mathbf{W})) - x_k^*|| \ . \tag{5.14}$$

In contrast to the performance error (see equations 2.12 and 4.4), this measure does not evaluate on the squared deviation between $f(g(x^*))$ and x^*, but on the ordinary distance between both values. While the square value is important for theoretical investigations, the error D^X is well suited for experimental evaluations because it can be easily interpreted as average distance between desired and actual outcome in cartesian coordinates.

Due to the coherent initialization in the home posture, the error variances across trials in the following experiments are considerably lower than in the linear domain experiments presented in chapter 4, which used entirely random initializations. There-

fore, the temporal characteristics of different parameter values are directly compared by the average error across trials in order to provide a compact overview. The variance is illustrated by additional minimum and maximum values for the final epoch.

5.4.1 Planar Arm: 1D Coordination Task

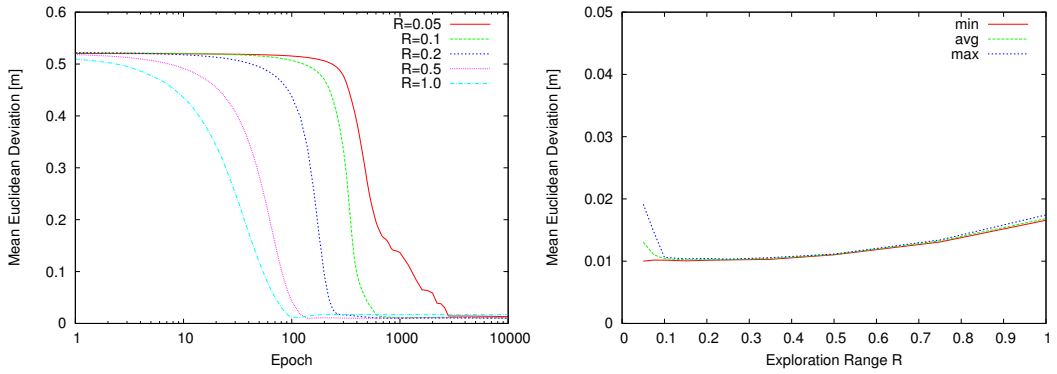
The first experiments concern the simulated robot arm in figure 2.2. The arm with initially two degrees of freedom ($m=2$) is used to coordinate only the height of the end effector ($n=1$). If only one dimension is coordinated, $K=1$ linear target motion is enough to cover the whole space of goals. This target movement spans the entire range between $x^*=-1.0m$ and $x^*=1.0m$ with $L=25$ intermediate steps. The home posture is $q^{home}=\vec{0}$ such that the arm is stretched and at height 0.0.

The most important parameter of the algorithm is the exploration range R. Figure 5.5(a) shows results for R varying between 0.05 and 1.0 radian over 10000 epochs and for 20 independent trials. The number of variations was set to $V=20$ and third order polynomials ($P=3$) are used for regression.

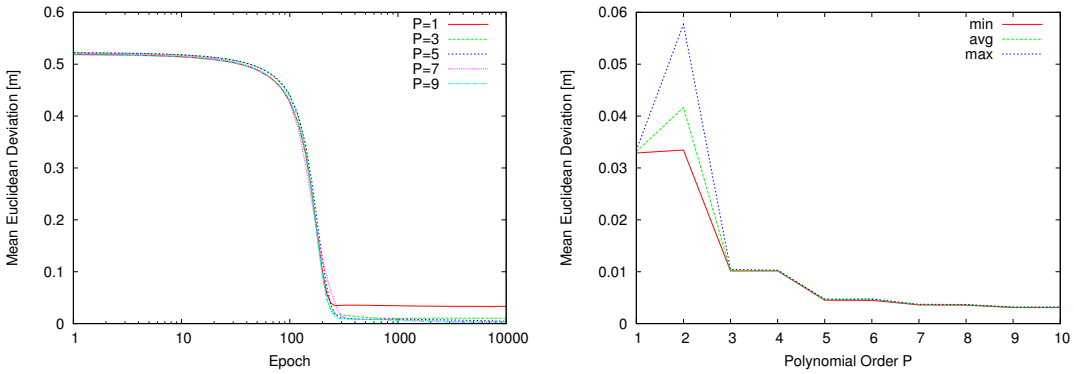
The left plot shows the mean euclidean deviation (see equation 5.14) over time for different values of R. The error decreases continuously. The qualitative stages orientation, expansion and tuning can be identified in all curves, where the expansion shows a rapid decrease of the error. High values like $R=1.0$ display the fastest convergence, but remain at a higher absolute error. The right plot shows the final error reached after 10000 epochs. For $R=0.05$ not all inverse estimates are converged after that time, depending on the initialization. For $R=0.1$ or higher, all trials have converged and show a very low error (1-2cm for an arm length of $1m$). An increase of error is visible for high values of R. Here examples are rather distant and the residual averaging error between the variations has a higher impact compared to small values of R. For $R=1.0$ the examples are generated in almost the entire joint space. However, the error is – in contrast to motor babbling – still small since the inconsistency resolution filters large portions of the generated examples. Although the speed varies, the general success of the goal babbling algorithm is rather insensitive to the concrete exploration range.

Figure 5.5(b) shows the same setup, but the exploration range is fixed to $R=0.2$ and the polynomial degree P is varied. The temporal characteristics of the error do not differ significantly for different polynomial degrees. Higher polynomial degrees allow a more accurate approximation of the examples. While first and second order polynomials do not yield a very accurate inverse estimate, the error reaches few millimeters for higher polynomial degrees (ca. $3mm$ error for $P=10$). The averaging error between variations must therefore be smaller than $3mm$. The error has converged in all cases and shows – depending on the expressiveness of the polynomials – a good performance. Goal babbling was successful for all values of P and in all independent trials.

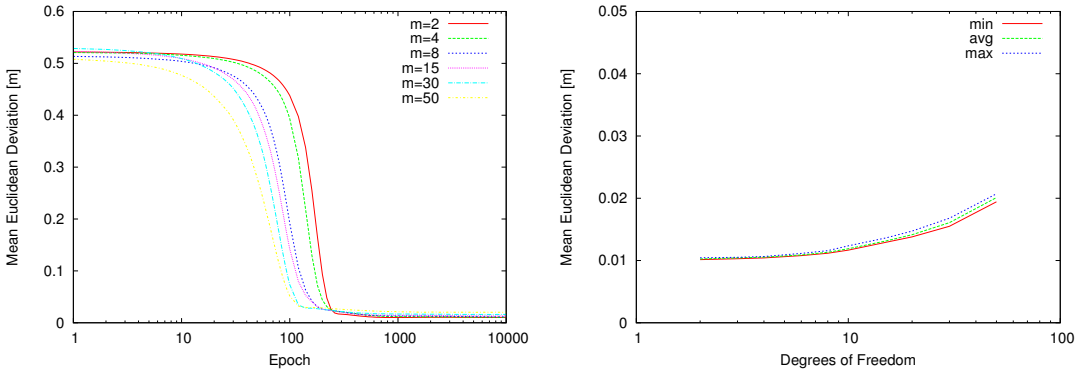
For the overall scope of this thesis, the most important question is how the method scales with the degrees of freedom m. Results for up to 50 degrees of freedom are shown in figure 5.5(c). For each value of m the arm was divided in segments of equal length, whereas the arm length is kept constant at $1m$. For instance, an arm with



(a) Higher exploration ranges R cause a faster convergence, but higher residual error.



(b) Polynomial with higher degrees P reach more accuracy, while the learning speed does not differ.



(c) Successful bootstrapping of inverse kinematics is possible also for 50 DOF.

Figure 5.5: Performance of goal babbling over 10000 epochs for the planar arm, whereas only the height is coordinated ($n = 1$). The left plots show the error over time, averaged over 20 independent trials. The maximum, average and minimum final error of 20 trials are shown of the right side.

$m=10$ comprises 10 segments with each $10cm$ length. Parameters $R=0.2$ and $V=20$ are used in order to compare the results to the previous experiments. The results show a rapid and reliable decrease of the error for all values of m and in all trials. The simulated arm with 50 degrees of freedom can be coordinated with an accuracy of $2cm$ after 10000 epochs. The method is systematically successful for such hyperredundant setups.

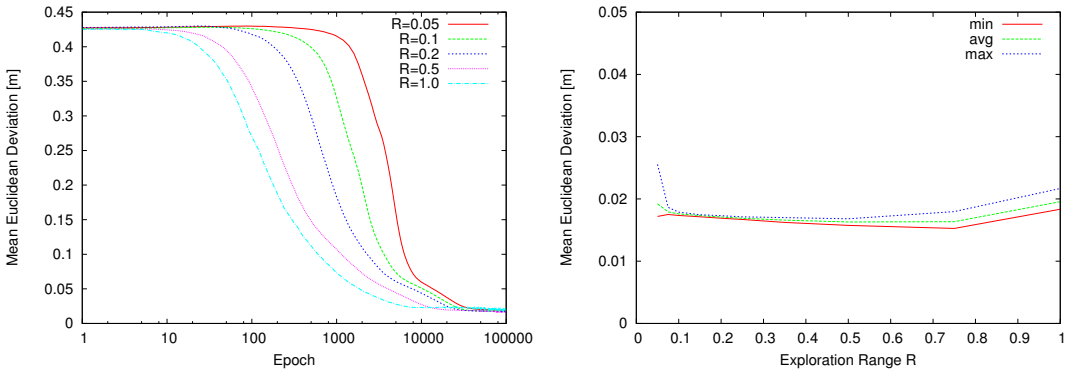
5.4.2 Planar Arm: 2D Coordination Task

The experiments continue with the simulated planar arm, but increase the dimension of the coordination task. Instead of coordinating only the height ($n = 1$), the 2D position of the effector ($n = 2$) is considered. The position is encoded in cartesian coordinates with origin in the base of the arm. The step from $n=1$ to $n=2$ is essential to show the validity of the movement direction weighting for redundancy resolution (equation 5.7). In 1D the angle between intended and actual movement direction can only be 0.0° or 180.0°. In $n=2$ arbitrary angles can occur. Since the weighting scheme only uses the immediate temporal and spatial context, each goal position must be passed from different directions for a correct resolution of inconsistencies.

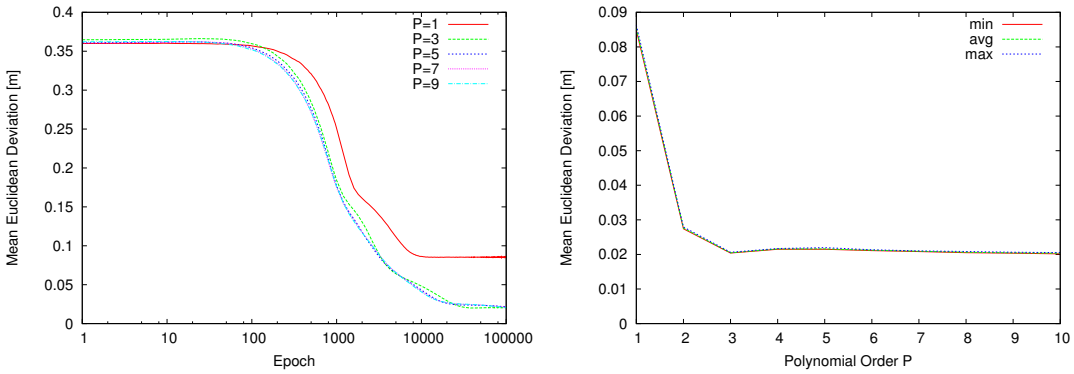
The aim in this set of experiments is to gain control over a part of the possibly reachable positions as shown in figure 5.7. The set of goal positions is shown as a grid. The home posture is set to a slightly curved shape, since a stretched position corresponds to a singularity in the 2D task. Learning would still be possible from that position, as either an "elbow-up" or "elbow-down" configuration could be chosen. However, it takes more time for the exploration to leave the singularity. A new sequence of targets x_t^* is generated in each epoch. $K=15$ goals are randomly selected from the target grid shown in figure 5.7. One after the other is connected by a linear target motion with $L=7$ intermediate target positions ($l = 0 \dots L-1$). As in the $n=1$ experiment, the target selection does not depend on learning progress and $\mathbf{X}^*$ does not change over time. However, in $n = 1$ one linear motion covers the entire target space. In $n = 2$ multiple linear series are required.

The experiments of the $n = 1$ case are entirely repeated with this $n = 2$ setup. The results are summarized in figure 5.6. The algorithm requires more epochs for convergence than in the $n = 1$ setup. Except for the speed, all results can be reproduced for $n = 2$. Again parameters $P = 3$, $R = 0.2$ and $V = 20$ are used as default values, and one redundant degree of freedom is incorporated, such that $m = 3$. Higher exploration ranges (see figure 5.6(a)) result in a faster convergence. The converged performance error only shows marginal differences across values of R. In all cases the error converges below $2cm$. Only for very small exploration ranges the error has not yet converged after 100000 epochs. The variation of polynomial degrees P (see figure 5.6(b)) shows a good and reliable performance for all $P \geq 2$. In the case of 2D position control, linear models ($P=1$) are not expressive enough to represent an accurate inverse solution.

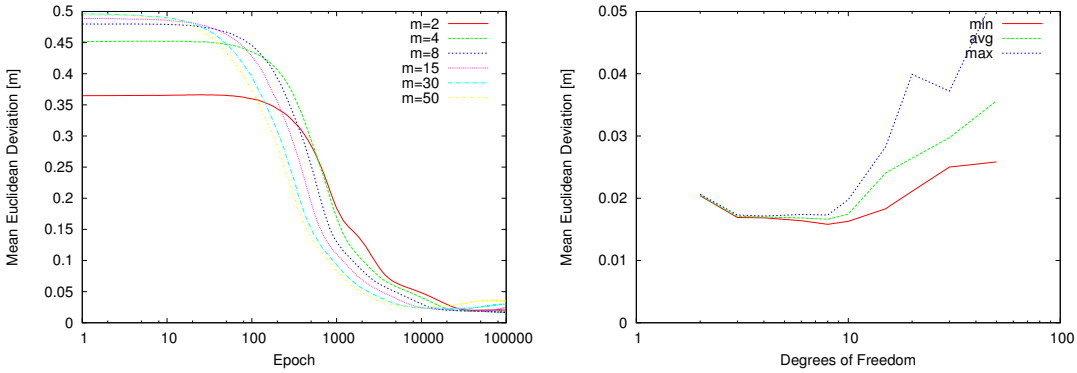
Also for the 2D coordination task, goal babbling is successful for hyperredundant setups. Figure 5.6(c) shows results for up to 50 degrees of freedom. An example



(a) Higher exploration ranges R cause a faster convergence, but higher residual error.

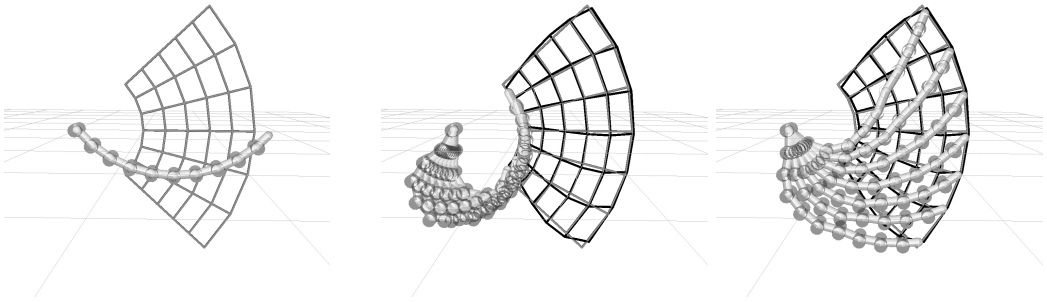


(b) Polynomial with higher degrees P reach more accuracy, while the learning speed does not differ.



(c) Successful bootstrapping of inverse kinematics is possible also for 50 DOF.

Figure 5.6: Performance of goal babbling over 10000 epochs for the planar arm, where the 2D position of the effector is coordinated ($n = 2$). The left plots show the error over time, averaged over 20 independent trials. The maximum, average and minimum final error of 20 trials are shown of the right side.



(a) Target positions x^* are shown as gray grid. The arm shows the home posture.

(b) The actually reached positions $f(g(x^*))$ are shown as black grid. Multiple postures $g(x^*)$ are overlaid to show how the redundancy is resolved.

Figure 5.7: An inverse estimate for 2D position coordination of a planar 10 DOF arm generated with the goal babbling algorithm. A third order polynomial was used as approximation model. The inverse estimate is very accurate as the reached positions are close to the target positions. The inverse estimate makes efficient use of all degrees of freedom.

solution $g(x^*)$ for $m=10$ is shown in figure 5.7. Goal babbling reliably yields accurate inverse estimates for $n=2$. The results confirm that the weighting-based resolution of inconsistencies is valid, although it only uses local information.

5.4.3 Humanoid robot: 3D Coordination Task

A further increase of complexity is investigated with a kinematic simulation of a humanoid robot (see figure 5.8), where $m=15$ degrees of freedom need to be coordinated. Five joint angles are controlled in each arm: three rotational joints in the shoulder, one in the elbow, and the rotation of the hand around the forearm axis. Four virtual joints are controlled in the hip: its orientation around all three spatial axes and the height over ground. The hip degrees of freedom are implemented by means of leg motion, whereas the leg joints are automatically adjusted to realize the desired hip pose [Takenaka, 2006]. As additional degree of freedom, the head-pan direction is controlled. This joint is, like the joints in the left arm, irrelevant for the task. The kinematic structure is rather complex compared to the planar arm, as the joints have offsets and rotate the hands around different axis. Since the ranges of the possible angles differ significantly between different joints, the values are normalized to the range $q_i \in [-1.0;\ 1.0]\ \forall i=1...15$.

This experiment concerns the coordination of the 3D spatial position of the right hand ($n=3$). Nine degrees of freedom are relevant for this task (five in the arm and four in the hip). The set of target positions is defined in a cube with $20cm$ edge length in front of the upper body (see figure 5.10). A sequence of targets x_t^* is generated

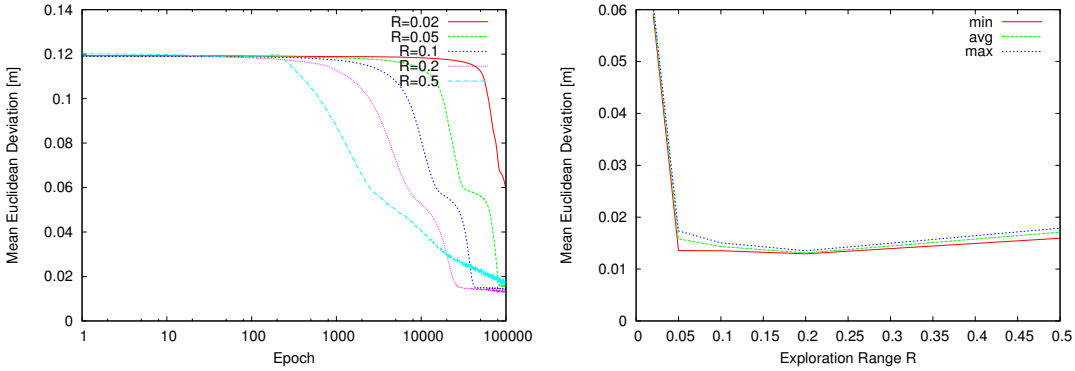
Figure 5.8: Humanoid robot morphology used for reaching in 3D. *Figure from [Rolf et al., 2010b].*

newly in each epoch according to equation (5.4) with $K = 50$ goals $L = 10$ intermediate steps. Default parameters values are $P = 3$, $R = 0.2$ and $V = 25$.

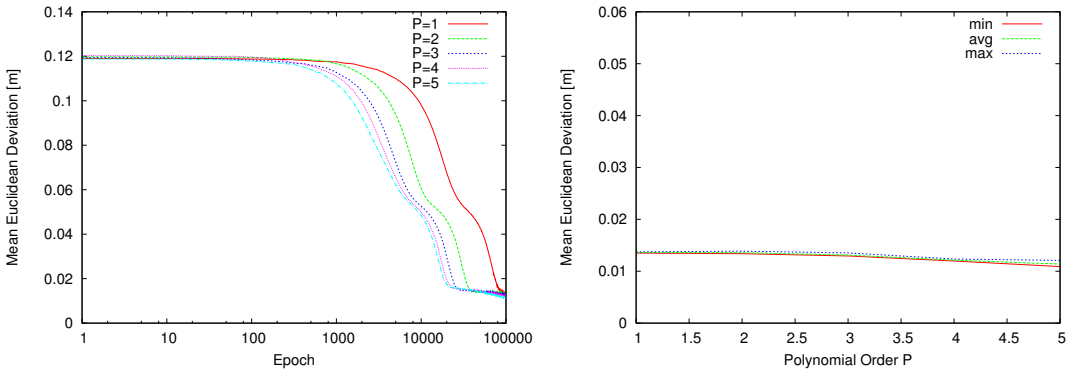
The results are shown in figure 5.9. The mean euclidean deviation is shown over time for different exploration ranges R and polynomial degrees P. For $n = 3$ it takes more time for the inverse estimate to orient with the correct movement directions. The error decreases slowly, but continuously. The temporal curves, but also the converged errors have the same characteristics as in the planar arm experiments. Higher exploration ranges cause faster convergence but higher residual errors. The performance benefits from higher polynomial degrees, indicating that the full expressiveness of the model can be used. Already linear models ($P = 1$) yield accurate inverse estimates with performance errors around $1.5cm$ inside the cube of targets. An example solution with a third order polynomial is shown in figure 5.10. The task-relevant degrees of freedom in the hip and in the right arm are used effectively. The task-irrelevant joints are stabilized in an approximately fixed position, which is the most efficient way to deal with irrelevant joints. The algorithm shows a reliable performance also on humanoid morphologies with complex kinematic structure in three dimensions.

5.5 Discussion

Inverse models can not be learned from arbitrary data sets in non-linear redundant domains due to the non-convex solution sets that forbid averaging. Inconsistent examples can also occur during goal-directed exploration although it generates examples in a highly structured manner. The analysis presented in this chapter, however, showed that during this exploration, inconsistencies can only occur in very specific ways. Con-



(a) Higher exploration ranges R cause a faster convergence, but higher residual error.



(b) Polynomial with higher degrees P reach more accuracy, while the learning speed does not differ.

Figure 5.9: Performance of goal babbling over 100000 epochs for the humanoid robot, where the 3D position of the right hand is coordinated ($n = 3$). The left plots show the error over time, averaged over 5 independent trials. The maximum, average and minimum final error of 5 trials are shown of the right side.

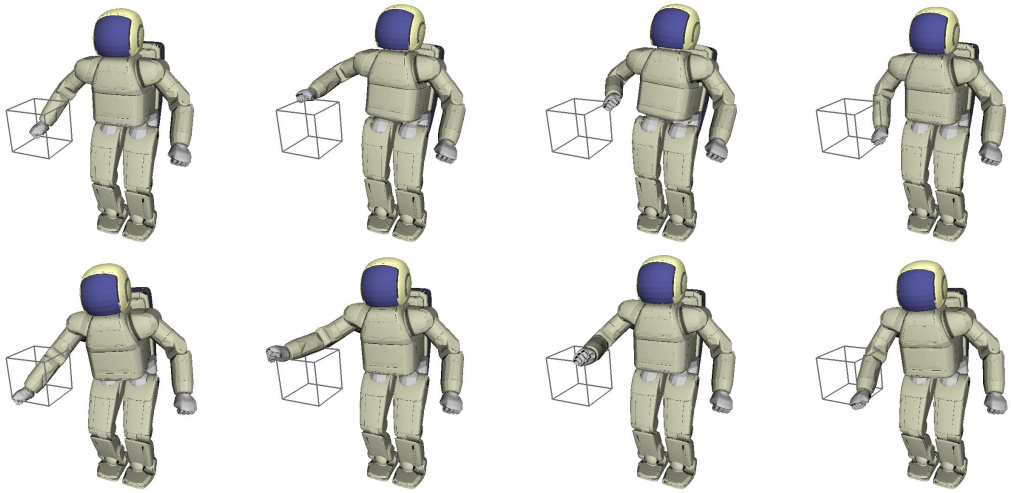


Figure 5.10: An inverse estimate for the humanoid morphology generated with goal babbling. Goal positions are located in a cube with $20cm$ edge length in front of the body. Several postures show how the inverse estimate reaches for the corners. All relevant degrees of freedom are effectively used. Irrelevant degrees of freedom (e.g. in the left arm) stay approximately fixed. *Figure from [Rolf et al., 2010c].*

sidering goals as a reference structure allows to detect and remove them by means of a simple weighting scheme that considers continuous paths of movements. This is possible solely from information that is directly observable, like current movement directions and amplitudes. This stands in contrast to error-based methods for the learning of inverse models [Kawato, 1990, Jordan and Rumelhart, 1992], which rely on an error-signal that is not directly observable and needs to constructed by prior knowledge.

The weighting based resolution is thereby rather heuristic, although it is motivated by strict logic about the geometry and structure of goal-directed exploration. The weighting scheme only considers one position and direction at a time. It is plausible that there are pathologic cases when this is not sufficient and the entire space of goals needs to be considered, as it is visualized in figure 5.2. However, the experiments show conclusively that this heuristic scheme is successful, at least when learning starts locally in a home posture. It is likewise plausible that such pathologic cases can generally not be reached during this highly structured and regularized learning procedure. A nevertheless important aspect for future work is to extend the theory developed for linear cases to this setup.

The experiments show that the goal babbling algorithm allows to learn inverse models for different morphologies, in domains with entirely different dimensions, for a wide set of parameter values, and without prior knowledge about the structure of the

corresponding coordination problems. This chapter has therefore introduced the first algorithm that can learn inverse models from examples in spite of non-convex solution sets [Jordan and Rumelhart, 1992]. Together with chapter 4, this work has clearly fulfilled the research goal to enable example-based learning of inverse models.

The number of examples necessary thereby largely depends on the task dimension n. Three-dimensional tasks like reaching on the humanoid morphology requires more examples to cover the sets of goals $\mathbf{X}^*$ and observations $\mathbf{X}$. This increase of cost for higher n is very natural: a task with $n = 3$ can be viewed as three interfering tasks with $n = 1$ at the same time, which is intuitively more difficult than a single one-dimensional task. Typical tasks, however, are substantially lower dimensional than the corresponding action space with dimension m. The experiments in this chapter provide a first indication that goal babbling indeed permits an excellent scalability to high dimensional action space. For a fixed task dimension n, the exploratory cost barely depends on the action dimension m, and even additional, irrelevant degrees of freedom do not impair the performance. This stands in harsh contrast to motor babbling, which does not take into account the task and does not allow to scale to high-dimensional action spaces. This scalability can be achieved with goal babbling because it does not attempt to learn all different solutions to the same outcome in redundant domains. In particular when goal babbling is used to learn inverse models – as it is approached in this thesis – only one solution is essentially explored and represented. This behavior is well visible in figure 5.3: only a n-dimensional manifold is explored at a time, even if the action space has a much higher dimension.

On a conceptual level it can be summarized that goal babbling *is* a successful strategy for the bootstrapping of coordination skills. Two mechanisms are necessary to enable this approach, besides the elementary formulation of plain goal-directed exploration as presented in [Oyama and Tachi, 2000, Sanger, 2004]. *Exploratory noise* is necessary to avoid a degeneration of example data into subspaces. In redundant domains, a *regularization* mechanism is necessary that prevents unstable drifts into unknown regions of the action space. This can be effectively solved by using a home posture as starting and return point for learning and exploration. An important characteristic of goal babbling is the highly structured exploration across goals instead of arbitrary random actions. This is clearly visible even from a distal perspective, since examples are only collected on a low-dimensional manifold in the action space. This structure does not only permit to scale to high-dimensional action spaces, but this very structure can be exploited to resolve inconsistent solutions by using goals as reference.

Chapter 6

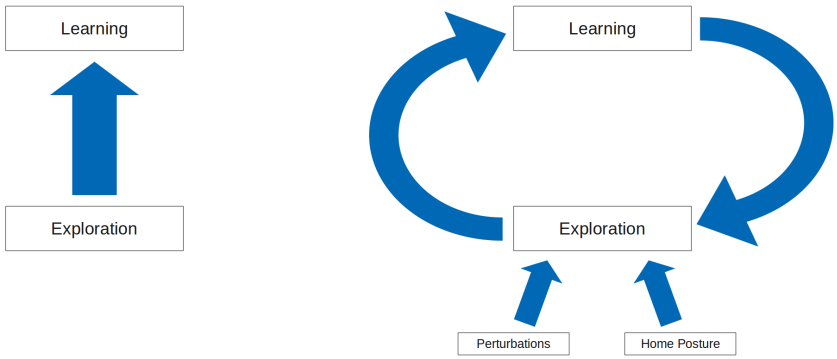
Online Learning Dynamics during Goal Babbling

The previous two chapters have shown that goal babbling allows for a successful learning of inverse models. This chapter concerns the practical applicability of the approach and investigates the absolute *speed* of learning. The algorithm formulation of the previous chapter relies on the collection of several examples for various goals and variations. The experiments already showed that the approach is rather scalable to high-dimensional action spaces. The absolute number of necessary examples, however, is comparably high since each epoch can contain several thousand examples and several hundred to thousand epochs are necessary depending on the task dimension.

This chapter first revises the data generation formulation in order to generate movement paths that are entirely continuous in time. The previous chapter motivated to switch between a discrete number of different variations because each of these variations can be seen as a simple function that permits a concise analysis of the structure of inconsistent solutions. This chapter relaxes the constraint to organize exploration in such a discrete way and introduces a formulation that continuously blends between variations. This step is important for the application of reaching on a real robot, since the movements must be physically continuous. Based on this formulation, this chapter investigates how the demand for examples can be reduced by applying online learning, i.e. performing a learning step after each generated example. Distinct measurements of the bootstrapping speed show that the method scales with almost constant, and very low exploratory cost across different dimensions of action spaces. The exploration formulation introduced in this chapter as well as the experiments have been published in [Rolf et al., 2011].

6.1 Online Learning in the Loop

Online learning with gradient descent is a widely used approach for a variety of machine learning problems [Rumelhart et al., 1986, Jordan and Rumelhart, 1992, Bottou and LeCun, 2004, Peters and Schaal, 2007]. Instead of performing a "batch" gradient step on an entire data set, single examples are selected in order to perform a gradient step. In machine learning tasks with fixed data sets, online learning is typically regarded as a stochastic approximation of batch gradients. This implies that online gradient learning is a sound mechanism to reduce some error functional and that it exposes



(a) Learning with uninformed exploration (b) Goal babbling

Figure 6.1: Uninformed exploration processes like motor babbling allows to understand online gradient learning as a stochastic approximation of batch gradients. During goal babbling, exploration and learning mutually inform each other. This loop breaks the assumptions of the stochastic gradient approximation and leads to very particular learning dynamics that are investigated and exploited in this chapter.

characteristics that are comparable to those of batch gradient learning.

The fundamental assumption that underlies the idea of a stochastic approximation is that examples must be drawn (*i*) *independently* and (*ii*) *identically distributed* from some real-world or empirical data distribution [Bottou, 1998]. These assumptions typically hold when forward models are learned by means of motor babbling. The distribution of examples in that case is a uniform distribution over all actions. Each example is chosen from that distribution, so that they are identically distributed. Typically, an entirely new example is chosen in each step so that they are also independent[1]. Even if learning and exploration are temporally intertwined by means of online learning, the exploration is not informed by the learning, but purely random (see figure 6.1(a)). Hence, the stochastic approximation holds which implies that for motor babbling, there is no fundamental difference between batch and online learning.

Goal babbling follows an entirely different organization of exploration and learning. Not only the learning is informed by exploration, but also the exploration is informed by learning (see figure 6.1(b)) when goal-directed movements are attempted. Thereby the learning forms, and continuously changes the distribution of example data (see section 4.2.2). Initially only examples close to the home posture are explored, before learning starts to unfold the inverse estimate which leads to generation of different example distributions (see figure 5.3). Examples during this initial bootstrapping are clearly *not identically distributed*. When online learning occurs from continuous paths,

[1]This is not the case when *random paths* of actions are considered, such as [Schillaci and Hafner, 2011]

examples are also temporally correlated and thus *not independent*.

The use of temporally correlated examples is known to cause the problem of "catastrophic interference" [McCloskey and Cohen, 1989]. Although this problem is barely theoretically solved [Biehl and Schwarze, 1995, Sollich and Barber, 1996], there are practical solutions that are based on the formulation of local learning mechanisms, and that are used in the following experiments (see section 6.2.3). The consequences of a "loop" between exploration and learning, and the resulting change of example distributions, are not generally clear. Of course, goal babbling can not be done in a pure batch manner since it defines an incremental process. However, the results in this chapter will challenge the view to regard online learning steps as an approximation of the iterative batch updates used in the previous chapter. The experiments provide evidence showing that this loop during goal babbling is in fact a *positive feedback loop* that permits substantial, non-trivial speedups of learning.

6.2 Online Goal Babbling Formulation

As a first step towards an online implementation, this section introduces an exploration formulation that generates entirely continuous paths. The general pattern of goal-directed exploration is the same as in the previous chapters. In each timestep t, an action q_t is explored and the outcome is observed:

$$x_t \quad = \quad f(q_t). \tag{6.1}$$

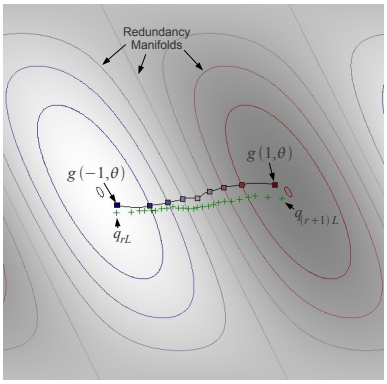
In order to generate examples, the algorithm starts with an initial inverse estimate $g(x^*, \theta_0)$ that always suggests the home posture: $g(x^*, \theta_0) = const = q^{home}$. Then continuous paths of target positions x_t^* are iteratively chosen from the set of goals $\mathbf{X}^*$. Exploration tries to reach for these targets with the inverse estimate:

$$q_t \quad = \quad g(x_t^*, \theta_t) + E_t(x_t^*). \tag{6.2}$$

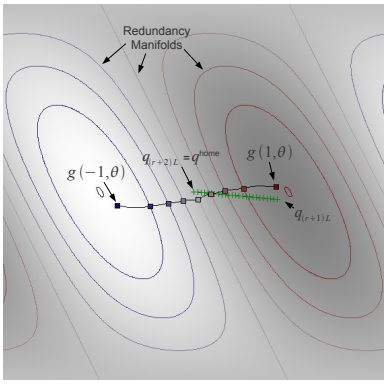
The outcome x_t is observed and the parameters θ_t of the inverse estimate are updated immediately before the next example is generated. The perturbation term $E_t(x^*)$ adds exploratory noise in order to discover new positions or more efficient ways to reach for the targets. This allows to unfold the inverse estimate and finally find correct solutions for all positions in the set of goals $\mathbf{X}^*$.

6.2.1 Continuous Path Generation

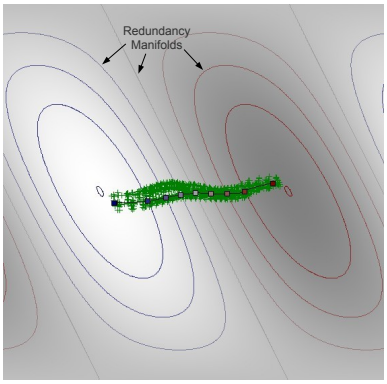
Beyond isolated timesteps t, the exploration is organized into reaching *movements*, which are counted by an index r. Each movement describes a linear path with a fixed number L of intermediate timesteps, which is set to $L = 25$ throughout this chapter. Hence, a movement r starts at timestep $t = rL$ and ends at timestep $t = (r + 1)L$. The starting point of each movement is the last point of the previous movement, such that the exploration is entirely continuous in t.



(a) A linear goal-directed path shown in the action space.



(b) A subsequent homeward movement in the action space.



(c) 1000 successive examples.

Figure 6.2: Online goal babbling in the action space of example figure 2.2. (a) The inverse estimate is used for trying to move from x^*_{kL} (here -1) to some other target $x^*_{(k+1)L}$ (here $+1$). (b) The effector moves from the last goal-directed action back to the home posture. Figure (c) shows how the perturbation terms cover the local surrounding of the inverse estimate.

Goal-directed movements are thereby performed in the same way as in the previous chapter. Starting from some goal x^*_{rL}, a new goal for $t = (r+1)L$ is randomly drawn from $\mathbf{X}^*$ and these endpoints are linearly interpolated with $l = 1 \dots L$ steps:

$$x^*_{r\cdot L+l} = \frac{L-l}{L} \cdot x^*_{r\cdot L} + \frac{l}{L} \cdot x^*_{(r+1)\cdot L} \ . \tag{6.3}$$

An example is generated for each of these targets according to equations (6.1) and (6.2).

The initial target ($t=0$) is the outcome of the home posture: $x^*_0 = f(q^{home})$. In the first movement, the system tries to move to another target x^*_L which is drawn from $\mathbf{X}^*$. Between the timesteps 0 and L, the target positions are defined by the linear sequence between x^*_0 and x^*_L. Afterwards a new target x^*_{2L} is chosen from $\mathbf{X}^*$ and the second movement is attempted from x^*_L to x^*_{2L}. An exemplary movement generated in this way is shown in figure 6.2(a).

During these goal-directed movements, each example receives a weight w_t in order to resolve inconsistent solutions (see section 5.1):

$$w^{dir}_t = \frac{1}{2}\left(1 + cos\sphericalangle(x^*_t - x^*_{t-1},\ x_t - x_{t-1})\right) \ , \tag{6.4}$$

$$w^{\mathit{eff}}_t = ||x_t - x_{t-1}|| \cdot ||q_t - q_{t-1}||^{-1} \ , \tag{6.5}$$

$$w_t = w^{dir}_t \cdot w^{\mathit{eff}}_t \ . \tag{6.6}$$

While the generation of goal-directed movements follows the same pattern as described in the previous chapter, the integration of a home posture demands a different mechanism when continuous paths are desired. Algorithm 1 simply added the home posture q^{home} with the corresponding outcome x^{home} to the set of training data, which corresponds to a discrete and potentially huge jump from the performance of the last goal x^*. In order to generate continuous movements, this jump is explicitly interpolated. After the execution of a linear goal-directed path r, a randomized decision is made whether the next movement $(r+1)$ is also goal-directed (equation 6.3), or the next movement approaches the home posture. The probability p^{home} for a "homeward" movement is set to $p^{home} = 0.1$ throughout this chapter. Such movements are modeled by a linear path in action space: the system moves from the last goal-directed action $q_{r\cdot L}$ to its home posture q^{home}, whereas equation (6.2) is replaced by the following expression:

$$q_{r\cdot L+l} = \frac{L-l}{L} \cdot q_{r\cdot L} + \frac{l}{L} \cdot q^{home} \ . \tag{6.7}$$

For every generated joint configuration, the resulting effector pose is observed (equation 6.1) and learning is applied online in the same way as for goal-directed movements. These examples are only weighted with $w_t = w^{\mathit{eff}}_t$, because targets x^*_t for the evaluation w^{dir}_t do not exist during this homeward movement[2]. After the home posture has been reached, a goal-directed movement is attempted from the initial target

[2] An alternative formulation of this behavior that allows for an elegant implementation is to assume "virtual" goals $x^*_t \stackrel{!}{=} x_t$, which implies $w^{dir}_t = 1$.

$x^*_{(r+1)\cdot L} = f(q^{home})$. An example of this movement type is shown in figure 6.2(b).

Using this formulation of goal-directed and homeward movements allows for an intuitive assessment of exploratory costs. In contrast to epochs that consist of distinct paths, variations, and isolated home examples, the entire exploration can be assessed with the number of movements r that have already been executed. In the case of reaching this measure of time is also more meaningful than counting the number of isolated examples, since physically continuous movements must be executed in any way on a real robot.

6.2.2 Structured Continuous Variation

In order to generate continuous paths of actions, the above formulation of goal-directed and homeward movements needs to complemented with a formulation of *exploratory noise* that is likewise continuous in time. The previous formulation in chapter 5 used random linear function that are added to the inverse estimate in order to perform goal-directed movements with that variation. Therefore a discrete number of variations is chosen and goal-directed paths are performed with each of them. A temporally continuous re-formulation of this scheme can be found by considering a *random walk* of linear perturbation terms. At any point in time the perturbation is modeled by a linear function:

$$E_t(x^*) = A_t \cdot x^* + b_t, \quad A_t \in \mathbb{R}^{m\times n},\ b_t \in \mathbb{R}^m \tag{6.8}$$

Initially, all entries e^i_0 of the matrix A_0 and the vector b_0 are chosen i.i.d. from a normal distribution with zero mean and variance σ^2:

$$e^i_0 \sim N(0, \sigma^2), \tag{6.9}$$

In order to explore different variations of the inverse estimate over time, these parameters slowly varied with a normalized Gaussian random walk. A small value δ^i_{t+1} is chosen from a normal distribution $N(0, \sigma^2_\Delta)$ with $\sigma^2_\Delta \ll \sigma^2$, and added to the previous value e^i_t. The variance of the resulting value is the sum of the individual variances $\sigma^2 + \sigma^2_\Delta$. In order to maintain a stable amplitude of the perturbations, this term is normalized with the factor $\sqrt{\sigma^2/(\sigma^2+\sigma^2_\Delta)}$, which keeps the overall amplitude stable at σ:

$$\delta^i_{t+1} \sim N(0, \sigma^2_\Delta)\ , \tag{6.10}$$

$$e^i_{t+1} = \sqrt{\frac{\sigma^2}{\sigma^2+\sigma^2_\Delta}} \cdot (e^i_t + \delta^i_{t+1}) \sim N(0, \sigma^2)\ . \tag{6.11}$$

Hence, $E_t(x^*)$ is a slowly changing linear function. It is smooth at any time, which is important for the evaluation of the weighting scheme that resolves inconsistent solutions. It is furthermore zero centered and limited to a fixed variance which leads to a local exploration around the inverse estimate (see figure 6.2(c)). This process can indeed be seen as an online approximation of the discrete variations in chapter 4 and

Algorithm 2 Online Goal Babbling Formulation

Require: Forward function: $f(q)$
Require: Home posture q^{home}
Require: Set of target positions: $\mathbf{X}^*$
  Initialize learner: $\theta \leftarrow \theta_0$ such that $g(x^*, \theta) = q^{home}$
  Initialize variation: $E_0(x^*)$ (Equation 6.9)
  Origin for new movement paths: $x^*_{(e)} \leftarrow x^{home}$, $q_{(e)} \leftarrow q^{home}$
  The first movement is goal-directed: $G \leftarrow true$
  while true **do**
    if G is $true$ **then**
      Chose $x^*_{(new)}$ from $\mathbf{X}^*$
    end if
    for $l = 1 \dots L$ **do**
      if G is $true$ **then**
        Interpolate goal $x^*_t = \frac{L-l}{L} \cdot x^*_{(e)} + \frac{l}{L} \cdot x^*_{(new)}$
        Generate goal-directed action q_t (Equation 6.2)
      else
        Interpolate towards homeposture $q_t = \frac{L-l}{L} \cdot q_{(e)} + \frac{l}{L} \cdot q^{home}$
      end if
      Observe outcome x_t (Equation 6.1) and compute weight w_t (Equation 6.6)
      Update inverse model with (x_t, q_t, w_t) and update variation (Equation 6.11)
    end for
    if G is $true$ **then**
      $x^*_{(e)} \leftarrow x^*_{(new)}$, $q_{(e)} \leftarrow q_t$
      Set $G \leftarrow false$ with probability p^{home}
    else
      $x^*_{(e)} \leftarrow x^{home}$, $q^*_{(e)} \leftarrow q^{home}$
      $G \leftarrow true$
    end if
  end while

5, which allows to solve the inversion of causality (compare figure 5.3). The entire exploration procedure is summarized in algorithm 2.

6.2.3 Incremental Regression Model

For learning during this goal babbling algorithm, a regression mechanism for the inverse estimate $g(x^*)$ is needed that can cope with the temporally correlated presentation of examples during continuous movements. The experiments in the previous chapter did not expose this problem: although the examples are generated along temporally ordered paths, the examples were presented to the learner all together. When examples are used for learning in their temporal order, "catastrophic interference" [McCloskey and Cohen, 1989] can degenerate the entire learning. This thesis follows a

standard approach to contain this problem by means of *locally linear learning* [Ritter, 1991, Atkeson et al., 1997]. The inverse estimate consists of different linear functions $g^{(k)}(x)$, which are centered around prototype vectors $p^{(k)}$ and active only in its close vicinity which is defined by a radius d. The function $g(x^*)$ is a linear combination of these local linear functions, weighted by a Gaussian responsibility function $b(x)$:

$$\begin{aligned} g(x^*) &= \frac{1}{n(x^*)} \sum_{k=1}^{K} b\left(\frac{x^* - p^{(k)}}{d}\right) \cdot g^{(k)}\left(\frac{x^* - p^{(k)}}{d}\right) \\ b(x) &= \exp\left(-||x||^2\right) \\ n(x) &= \sum_{k=1}^{K} b\left(\frac{x - p^{(k)}}{d}\right) \\ g^{(k)}(x) &= W^{(k)} \cdot x + o^{(k)} \end{aligned}$$

The normalization $n(x)$ scales the sum of influences of the components to unity, which is known as *soft-max*.

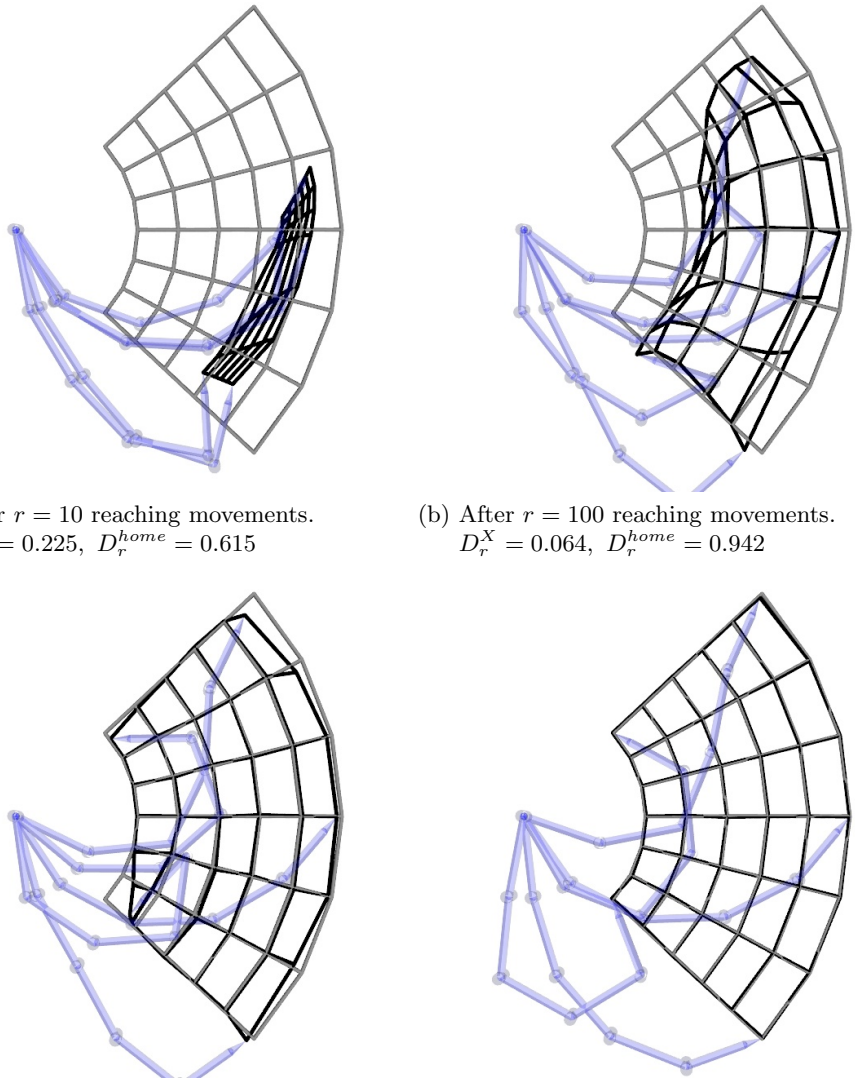
The inverse estimate is initialized with a single local function with center $p^{(1)} = f(q^{home})$, that outputs the constant value q^{home} ($W^{(1)} = 0_{m \times n}$ and $o^{(1)} = q^{home}$). New local functions and their prototype vectors are added dynamically. Whenever the learner receives an input x, that has a distance of at least d to all existing prototypes, a new prototype $p^{K+1} = x$ is created. In order to avoid abrupt changes in the inverse estimate, the function $g^{K+1}(x)$ is initialized such that its insertion does not change the local behavior of $g(x^*)$ at the position x. The offset vector o^{K+1} is set to the value of the inverse estimate before the insertion of the new local function: $o^{K+1} = g(x)$. A simple way to initialize the weight matrix is to use the average weights of topological neighbors [Fritzke, 1995], but which does not necessarily reflect the input-output behavior of g at the position x. The exact behavior is characterized by the Jacobian matrix $J(x) = \frac{\partial g(x)}{\partial x}$ of the learner. For the experiments in this chapter, the weight matrix is initialized accordingly: $W^{K+1} = J(x)$.

In each timestep, the inverse estimate is fitted to the currently generated example (x_t, q_t) by reducing the weighted action error:

$$E_w^Q = w_t \cdot ||q_t - g(x_t)||^2 \ .$$

The parameters $\theta = \{W^{(k)}, o^{(k)}\}_k$ of $g(x^*)$ are updated using online gradient descent on E_w^Q with a learning rate η:

$$\begin{aligned} W_{t+1}^{(k)} &= W_t^{(k)} - \eta \frac{\partial E_w^Q}{\partial W^{(k)}} \ , \\ o_{t+1}^{(k)} &= o_t^{(k)} - \eta \frac{\partial E_w^Q}{\partial o^{(k)}} \end{aligned}$$



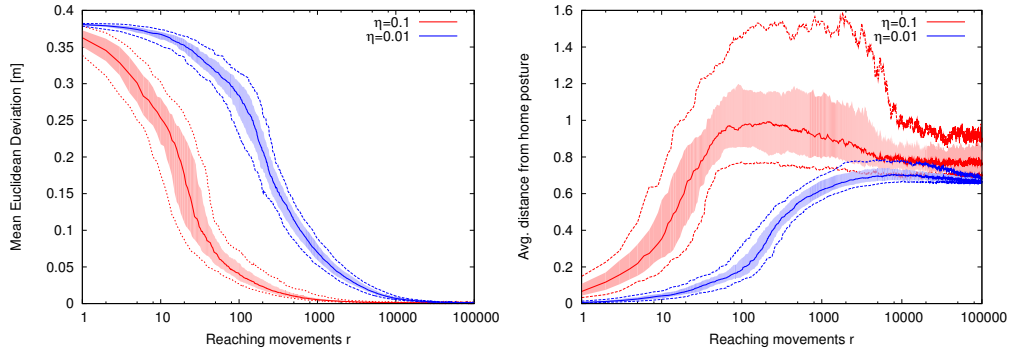
(a) After $r = 10$ reaching movements. $D_r^X = 0.225,\ D_r^{home} = 0.615$

(b) After $r = 100$ reaching movements. $D_r^X = 0.064,\ D_r^{home} = 0.942$

(c) After $r = 1000$ reaching movements. $D_r^X = 0.011,\ D_r^{home} = 0.995$

(d) After $r = 10000$ reaching movements. $D_r^X = 0.002,\ D_r^{home} = 0.737$

Figure 6.3: Example of the bootstrapping dynamics for a five DOF arm with learning rate $\eta = 0.1$. The inverse estimate rapidly finds valid solutions as the actual positions (black grid) become congruent with the targets (gray grid). Blue postures show how the redundancy is resolved.



(a) (b)

Figure 6.4: Statistics of the bootstrapping dynamics for two different learning rates. (a) The euclidean deviation D^X decreases rapidly over the number of movements. A ten times higher learning rate results in a speed up of approx. 20. (b) The distance from the home posture initially increases as the inverse estimate unfolds. High learning rates η initially select less comfortable solutions which are then gradually optimized.

6.3 Experiments

This section investigates the functioning and learning dynamics of the online goal babbling formulation. The coordination problem is to learn reaching in the 2D plane ($n = 2$) with a planar revolute joint robot arm. The problem setup is identical to section 5.4.2. An example with five degrees of freedom is shown in figure 6.3. The target positions x^* are arranged in the gray grid structure. The black grid shows the actually reached positions ($x = f(g(x^*))$). Initially, the inverse estimate is fixed at the home position, but expands rapidly towards the target positions. After a number of movements, target and actual grids are in congruence. An accurate inverse estimate has been bootstrapped. Blue postures show configurations generated by the inverse estimate for several different target positions and thus how the redundancy is resolved.

Three different experimental measures are used to assess the temporal characteristics of the bootstrapping:

1. *Accuracy* of the bootstrapped inverse models.

2. *Comfort* of the selected solution.

3. *Speed* of the bootstrapping process.

The accuracy is measured in the same way as in the previous chapter: The *mean euclidean deviation* D^X (see also equation 5.14) measures the distance between the goal positions $x^*_k \in \mathbf{X}^*$ and the actually reached position, where D^X_r indicates the

value after performing r movements:

$$D_r^X = D^X(\mathbf{X}^*, \theta_{rL}) = \frac{1}{K} \sum_{k=0}^{k<K} ||f(g(x_k^*, \theta_{rL})) - x_k^*|| \ .$$

In order to assess the comfort of the selected solutions, D^{home} measures how far the suggested postures are away from the home posture:

$$D_r^{home} = D^{home}(\mathbf{X}^*, \theta_{rL}) = \frac{1}{K} \sum_{k=0}^{k<K} \left|\left| q^{home} - g(x_k^*, \theta_{rL}) \right|\right| \tag{6.12}$$

This measure can not be zero for a bootstrapped model, because the home posture has to be left in order to reach for different targets. Nevertheless it allows to compare how comfortably different inverse estimates resolve the redundancy.

The speed of bootstrapping is assessed by measuring the number of movements until a certain percentage of independent trials has reached some accuracy level:

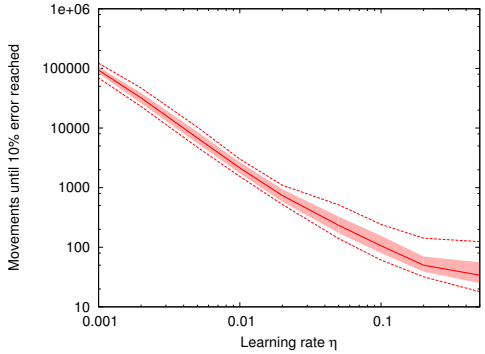
$$S(Q, d^X) = \underset{r}{argmin} \left(\ Q \leq p\left(D_r^X \leq d^X\right) \ \right) \tag{6.13}$$

For instance, $S(0.9, 0.1)$ counts the number of reaching movements, until 90% of the trials have reached a error below or equal to 0.1. The statistics presented in this section are all computed over 100 independent trials.

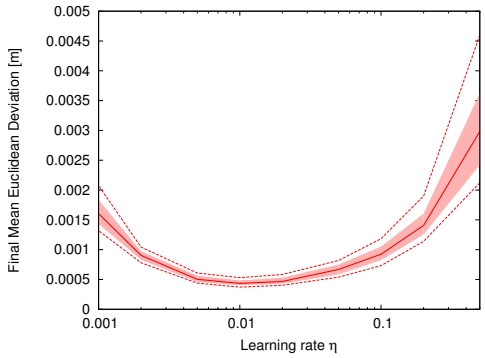
6.3.1 Effects of the Learning Rate

The most important variable for online learning from goal-directed exploration is the learning rate η. In supervised learning from fixed data sets, online learning is used as stochastic approximation of batch methods. In goal-directed exploration, however, the data set is not fixed but continuously constructed by the learner. This interweaved relation of data generation and learning leads to non-trivial effects with respect to the choice of the learning rate.

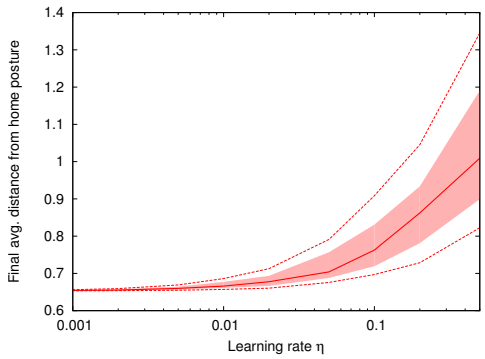
The default parameters for this experiment are $\sigma = 0.05$, $\sigma_\Delta = 0.005$ and $d = 0.1$. The home posture is set to a slightly bent shape with the effector at zero height, which is realized by setting the first joint to $-\frac{\pi}{3}$ and the remaining joints to $\frac{\pi}{6}$. Figure 6.4(a) shows the development of the error D_r^X over the number of movements r for the 5 DOF planar arm with a total length of $1m$. Bold lines show the median error, thin lines the 10% and 90% quantiles and the filled areas correspond to the range between the 25% and 75% quantiles. For both $\eta = 0.1$ and $\eta = 0.01$ the error decreases reliably and an accurate inverse model is obtained. Obviously the bootstrapping is faster for the higher learning rate, but the speedup does not scale with the factor 10 between the learning rates. For $\eta = 0.1$ the error has reached a median level of 0.04 after 100 movements. For $\eta = 0.01$ it takes 2000 movements to reach the same error level. Hence, the bootstrapping is 20 times faster for the high learning rate, although the rate itself is only 10 times higher.



(a)



(b)



(c)

Figure 6.5: Bootstrapping results for various learning rates between 0.001 and 0.5. (a) The number of movements needed to reach 10% of the initial error decreases rapidly for higher learning rates. (b) The euclidean deviation after 10^7 movements is very low for all learning rates. Very low learning rates are not fully converged. (c) The final distance from the home posture increases gradually for higher learning rates.

The distance from the home posture D_r^{home} for the same trials is shown in figure 6.4(b) and displays another qualitative effect of the learning rate. Low learning rates let the distance from the home posture increase gradually as the inverse estimate unfolds. It finally reaches a stable level which corresponds to a comfortable solution. High learning rates, in contrast, cause a rapid increase with high variance. The bootstrapping initially sticks to the very first solution that is observed due to the random perturbation term. This can result in a less comfortable redundancy resolution. After several thousand movements, the distance decreases again as comfortable solutions receive higher weights w^{eff} and dominate the learning in the long term.

An example trial for $\eta = 0.1$ is shown in figure 6.3. Already after 10 movements the inverse estimate has expanded from the home posture and is roughly aligned with the correct movement directions, and rapidly expands further. After 1000 movements, the inverse estimate starts to consolidate the redundancy resolution and the selected postures become closer to the home posture.

Results for a high range of learning rates $[0.001; 0.5]$ are summarized in figure 6.5. The bootstrapping speed is continuously increased for higher learning rates. Figure 6.5(a) shows the number of movements, until the euclidean deviation D^X is reduced to 10% of its initial value ($S(Q, 0.1 \cdot D_0^X)$, quantiles Q shown are 10, 25, 50, 75, 90). For the highest rate $\eta = 0.5$, 50% of the trials have reached this level already after 34 movements ($S(0.5, 0.1 \cdot D_0^X) = 34$). Non-trivial speedups can be seen across the several orders of magnitude span of learning rates. While the speedup between $\eta = 0.01$ and $\eta = 0.1$ is approximately 20, the speedup from $\eta = 0.001$ and $\eta = 0.01$ is even 50, which is substantially more than the factor 10 between the learning rates. After a total number of 10^7 movements the trials for all learning rates have reached an error in the millimeter range (figure 6.5(b)). For very low learning rates the inverse estimates are not fully converged, as indicated by the slightly increased error. For high learning rates both final error and the home posture distance (figure 6.5(c)) increase gradually.

The enormous learning-rate dependent speedups can not be explained by considering online learning as a stochastic approximation of batch gradient learning. The central reason for these speedups is the ongoing change of the example distribution due to the incremental and informed character of the goal-directed data-generation. Because the creation of each example is already informed by learning from the previous examples, learning does not only improve the inverse estimate, but will also result in a more informative next example. This example will in turn improve the inverse estimate which can then generate an even more informative example in the subsequent timestep. This phenomenon can be seen as a *positive feedback loop* (see figure 6.6). This feedback loop is also present for incremental batch updates as used in the previous chapter, but only becomes tight in an online scenario. Higher learning rates imply a higher "gain" in this loop and accelerate the bootstrapping over the sheer values of the learning rates. The presence of a positive feedback loop also explains the overshoot of the home-distance for high learning rates, since any observed movement direction can be reinforced in the beginning of the learning process. This leads to the very rapid selection of solutions, which can be suboptimal, but which are incrementally regularized by the weighting scheme and the home posture.

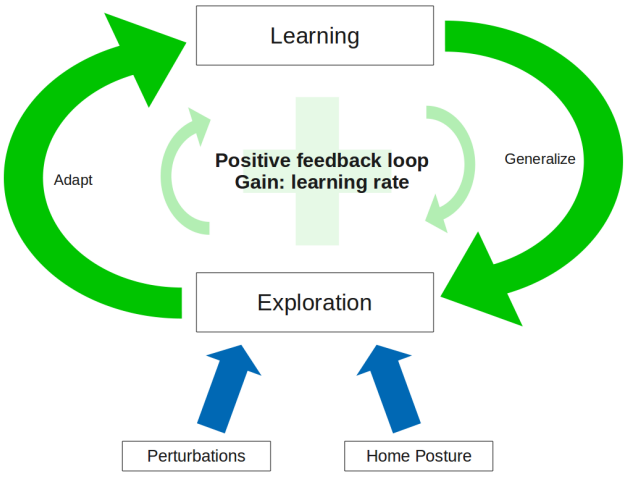


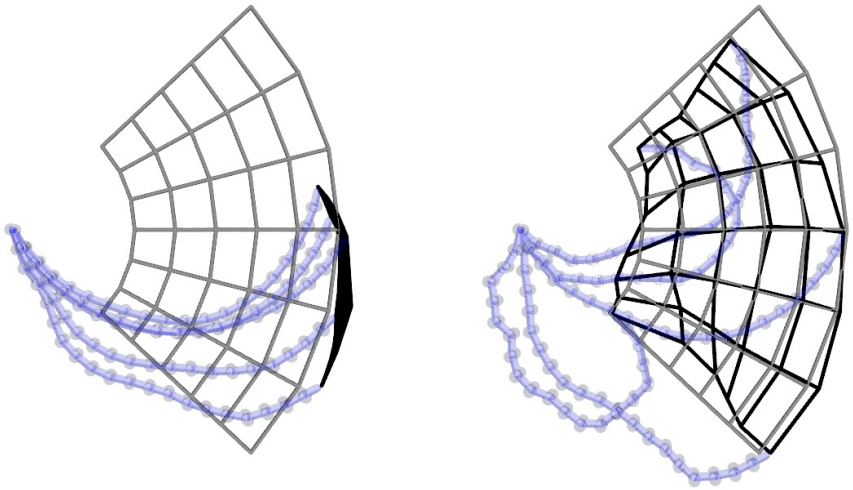
Figure 6.6: Goal babbling defines a positive feedback loop in which exploration and learning reinforce each other during the bootstrapping. The learning rate determines the amplitude of the learning process and thereby acts as gain of the feedback loop.

6.3.2 Scalability

The overall scope of this thesis is the design of exploration mechanisms that scale to high-dimensional action spaces. While the previous chapter has already shown the general feasibility in high dimensions, the following experiments assess the exact bootstrapping speed. In order to directly compare to the previous experiment with five degrees of freedom, the experiments consider the same setup, but the arm is split in m segments of equal length, each actuated by one joint. Hence, an arm with 20 degrees of freedom comprises 20 segments of length $0.05m$. The home posture is chosen as $-\frac{\pi}{3}$ for the first joint and $\frac{2\pi}{3(m-1)}$ for the remaining joints, which generalizes the bent shape used in the previous experiment to varying dimensionalities. The target positions are identical to those in the first experiment as indicated in figure 6.7. For a fair comparison between different dimensionalities, the perturbation term that generates variations needs to be scaled: if the variability per joint is constant, it has a higher effect on the end effector for high dimensional systems. This leads to a faster discovery of effector positions but also more instability. The deviation of outcomes σ_X can be approximated for an entirely stretched arm as a function of the joint variability σ and the number of DOF m:

$$\sigma_X = \sigma \cdot \sqrt{\frac{m+1}{2}} \ . \tag{6.14}$$

For this experiment σ is scaled such that σ_X is constant at $0.05 \cdot \sqrt{3}$ which is the variability in the five DOF experiment. The update parameter is set to $\sigma_\Delta = 0.1 \cdot \sigma$.



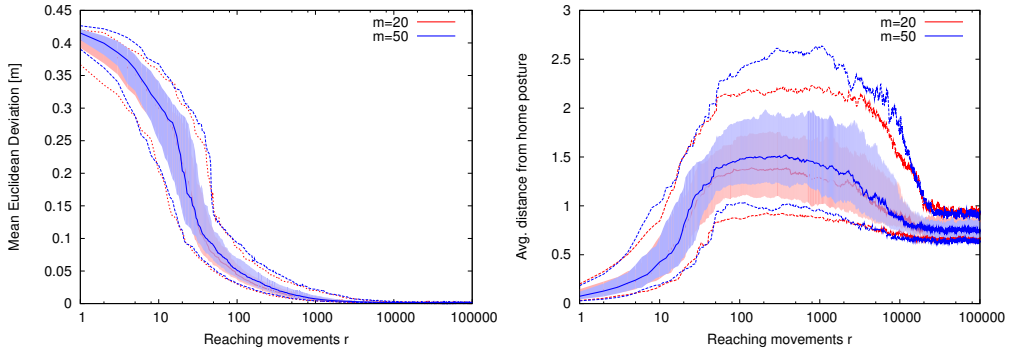
(a) After $r = 10$ reaching movements. $D_r^X = 0.353,\ D_r^{home} = 0.262$

(b) After $r = 100$ reaching movements. $D_r^X = 0.038,\ D_r^{home} = 1.173$

(c) After $r = 1000$ reaching movements. $D_r^X = 0.004,\ D_r^{home} = 0.940$

(d) After $r = 10000$ reaching movements. $D_r^X = 0.002,\ D_r^{home} = 0.922$

Figure 6.7: Example of the bootstrapping dynamics for 20 degrees of freedom. The inverse estimate unfolds with high speed also in high dimensions. The selected postures get smoother and more comfortable over time.



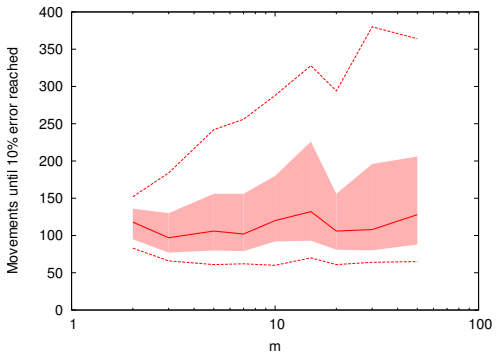
(a) (b)

Figure 6.8: Statistics of the bootstrapping dynamics for 20 and 50 degrees of freedom. Both euclidean deviation (a) and home posture distance (b) show a very similar behavior for 20 and 50 DOF. Goal Babbling scales without substantial extra cost in high dimensions.

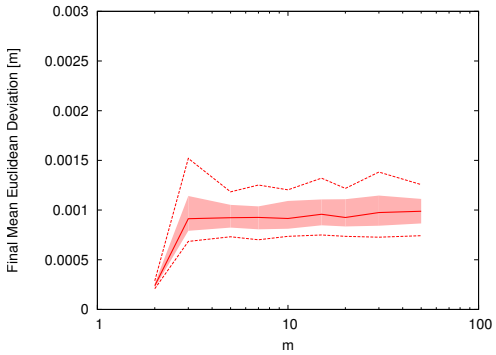
The learning rate is set to $\eta = 0.1$.

An example trial for $m = 20$ is shown in figure 6.7. The behavior over time, and in particular the speed of bootstrapping, is very similar to the previous five DOF example. The deviation D^X between goals and outcomes is reduced very rapidly during the first 100 movements. After 1000 movements the inverse estimate is already very accurate, but does not yet use optimally comfortable joint configurations. These are further optimized in the following movements as the configurations get smoother and the average distance to the home posture decreases. Figure 6.8 shows a comparison between $m = 20$ and $m = 50$ over time. The temporal characteristics of the euclidean deviation are virtually identical in the two cases and also compared to the $m = 5$ experiment (see figure 6.4). Also the home distance values show the same behavior, with slightly increased values for $m = 50$ in the intermediate movements.

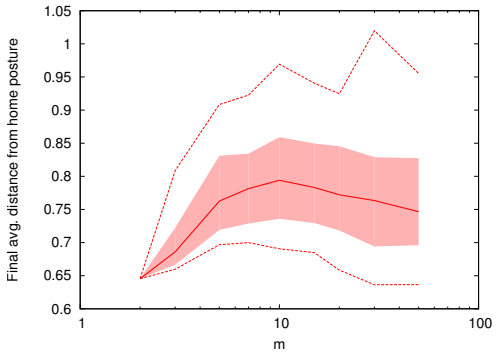
Results for values of m between 2 and 50 are summarized in figure 6.9. The most important result is that the median bootstrapping speed is virtually constant across the entire range of m. Even for 50 degrees of freedom, 50% of the trials have reached the 10% error level after after 128 movements ($S(0.5, 0.1 \cdot D_0^X) = 128$). However, the distribution becomes increasingly heavy-tailed as the values for the 90% quantile $S(0.9, 0.1 \cdot D_0^X)$ grow very slowly (e.g. $S(0.9, 0.1 \cdot D_0^X) = 364$ for $m = 50$). The 90% curve in the plot has an approximately linear shape, which indicates an empirical relation $S(0.9, 0.1 \cdot D_0^X) \sim log(m)$ due to the logarithmic scale. After a total number of 10^6 movements the euclidean deviation is approximately constant and very low at $1mm$. Only for $m = 2$ it is even lower with almost zero variance. Here the problem does not contain local redundancy, but only two separated choices "elbow up" and "elbow down" that can not be flipped by local perturbations. Higher values of m allow to modify the redundancy resolution continuously, which causes minor averaging errors.



(a)



(b)



(c)

Figure 6.9: Bootstrapping results for various numbers of joints. (a) The number of movements needed to reach 10% of the initial error increases only very gradually. (b) The euclidean deviation after 10^6 movements is very low in all cases. (c) The final distance from the home posture.

6.4 Discussion

This chapter has proposed an online formulation of goal babbling for the learning of inverse models. Exploration is organized along entirely continuous movement paths which prepares the practical use for reaching with a physical robot. Therefore a continuous formulation of exploratory noise, and a continuous integration of a home posture for regularization have been formulated.

The experimental evaluation shows that online learning during goal babbling is not only possible, but highly beneficial. The experiments show that the proposed algorithm is both *highly scalable and very fast* in bootstrapping inverse models. The measurement of the bootstrapping speed reveals that the algorithm scales with almost constant exploratory cost between two and 50 degrees of freedom. This result shows that the concept of goal babbling can indeed be implemented such that an efficient bootstrapping is possible, and is highly important for the practical use in real-world scenarios.

In terms of absolute speed, the online implementation provides a several orders of magnitude speedup compared to the batch update formulation in chapter 5, which requires several thousand epoch with few hundred movements in each epoch in order to solve the coordination of the planar arm morphology. The online algorithm only requires few hundred movements all together for the same task which is a speed that has not been achieved with any previous learning approach to coordination problems. This speed is also sufficient in practical scenarios when exploratory movements must be conducted physically. In fact, this speed is *competitive with human learning* [Sailer et al., 2005]: Sailer and Flanagan investigated how adults learn to solve novel coordination tasks, for which the participants had no prior knowledge. The authors found that participants need few hundred point-to-point movements until they can approximately achieve the desired outcomes.

The comparison between learning curves for different learning rates reveals the reason for the enormous speed of the online algorithm: the interplay between exploration and learning during goal babbling constitutes a positive feedback loop in which both processes inform and accelerate each other. Learning leads to more informative examples which lead to faster learning. This phenomenon can not be explained by the traditional view of online learning steps as a stochastic approximation of batch processes. Two successive online learning steps in this scenario have a stronger impact than a learning signal that averages two examples without learning in between, since the second example is already informed by the first one. It is therefore plausible that the formation of a positive feedback loop is not particular to the algorithm described in this chapter, but a direct *conceptual property* of goal babbling, which defines exploration as a process that is goal-directed and informed by previous learning steps.

Chapter 7

Application on a Bionic Elephant Trunk

Figure 7.1: The *Bionic Handling Assistant* mimics an elephant trunk.

This chapter demonstrates the practical use of goal babbling on the *Bionic Handling Assistant* (BHA) [Grzesiak et al., 2011] which is a new, award-winning [D. Zukunftspreis 2010] continuum robot platform inspired by elephant trunks and manufactured by *Festo* (see figure 7.1). The robot is pneumatically actuated and made almost completely out of polyamide which makes it very flexible and lightweight (ca. $1.8kg$). In contrast to standard robots with revolute joints, this robot moves by means of continuous deformations of the entire morphology, which is referred to as *continuum kinematics* [Jones and Walker, 2007]. Continuous deformations correspond to infinitely many mechanical degrees of freedom, which can neither be sensed, actuated, nor simulated. Deformations that are caused by a finite number of actuators on the robot, however, can be assumed to have effects that are (within certain limits) predictable and reproducible. Numerous studies investigate how such behavior can be analytically modeled [Hannan and Walker, 2003, Jones and Walker, 2006, Godage et al., 2011, Rolf and Steil, 2012a], but these approaches are limited to be approximations that can not capture the full complexity of continuous deformations.

Even if the behavior of a particular robot can be described exactly, the analytical modeling reaches its practical limit for robots with elastic elements. Such robots, like

the BHA, face additional problems with non-stationary behaviors due to hysteresis effects, visco-elasticity, and wear out effects of the mechanically exposed material. Learning becomes an essential tool in such scenarios in order to capture otherwise unmodeled non-linear behaviors of the continuous deformations and the ongoing changes and drifts in the actuation. An *efficient* approach to exploration and learning is even more important when the robot is non-stationary. If the action space is too high-dimensional to explore it exhaustively even once, then it is particularly pointless to attempt a full re-exploration in order to react to a change. A first indication that goal babbling allows for the mastery of such change has been provided in [Rolf et al., 2010a].

This chapter uses goal babbling in order to learn the inverse kinematics of the BHA, which is used to perform reaching movements. Section 7.1 introduces the details and the particular challenges of the BHA, as well as the experimental setup. Section 7.2 proposes several minor re-formulations of the online goal babbling algorithm in order to practically deal with these challenges. The sections 7.3 and 7.4 then apply the algorithm and show extensive real-world experiments with the BHA, as well as simulation experiments that allow to investigate the impact of non-stationary behavior in more detail. The results presented in this chapter have been published in [Rolf and Steil, 2013b].

7.1 Bionic Handling Assistant Setup

7.1.1 Actuation and Sensing

The BHA comprises three main segments, each with three pneumatic bellow actuators, a ball-joint as wrist, also actuated by three actuators, and a three finger gripper actuated by one bellow actuator. The experiments in this chapter only use the main segments, so that $m = 9$ actuated degrees of freedom are used. Each actuator can be supplied with compressed air, which unfolds and extends the actuator. The combination of three actuators per segment then allows to bend, and – in contrast to standard robots with revolute joints – *stretch* the entire robot.

For a reliable positioning, it is not sufficient to control the pressure alone: Friction, hysteresis and non-stationarities can cause largely different postures when supplying the same pressure several times. In particular during dynamic movements the pressure is not sufficient to determine the posture or position of the robot, since it only expresses a force on the actuators. This force reaches an equilibrium with the mechanical tension of the bellows after some time, so that the robot stands still. This physical process can, however, take up to 20 seconds because of a strong mechanical interplay between different actuators. Since pressure does not provide reliable information about the robot's position and movement in space, reaching solely concerns the geometric information from the BHA's *length-sensors* (see figure 7.2). These sensors permit to determine the outer length of each actuator. Although they do not allow for a "direct" actuation like pulling, the length values can be controlled by dynamically adjusting the pressure in each actuator. The system comprises a length-controller that performs

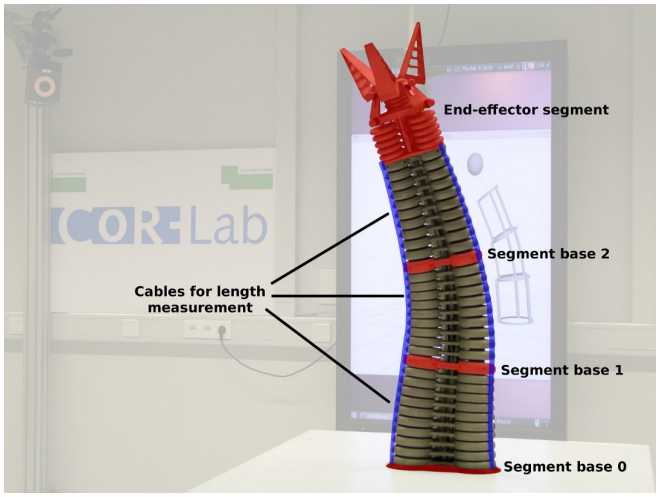


Figure 7.2: The kinematic structure of the BHA comprises three main segments, each consisting of three parallel pneumatic bellow actuators. The length of these actuators can be determined with cable-potentiometers.

this task automatically by means of PID-control and an additional, learned feedforward controller [Neumann et al., 2013]. Although this controller works accurately, the execution of a length-command takes a certain amount of time and the length-sensing is rather noisy. Hence, performing an action (i.e. applying some effector length) can generally not be done perfectly or even instantaneously. In order to disentangle desired and measured length values, this chapter refers to the desired length as $q^* \in \mathbb{R}^9$, while the measured actual length is referred to as $q \in \mathbb{R}^9$.

The forward kinematics function of this robot is not exactly known analytically, although approximations exists (see Sec. 7.4). For the experiments, the end-effector position is measured with a [Vicon] motion tracking system. Auto-reflective markers allow to measure the position with high accuracy by means of triangulation. The central position inside the gripper's palm is used as measurement and its *cartesian* value is referred to as $x \in \mathbb{R}^n, n = 3$. This measurements probes the unknown forward function $f(q) = x$. This function can not be evaluated directly for the BHA, but examples x and q can be observed on the physical robot.

7.1.2 Accuracy and Limits

Although the length of the actuators can be controlled, there are limitations to the positioning accuracy that need to be considered for learning experiments. The first important property of the BHA's morphology is that even minimal changes of the actuator lengths can lead to large, and direction-wise inhomogeneous changes of the effector position. In order to illustrate this phenomenon, the end-effector position for 200 random postures has been recorded, each drawn i.i.d. from normal distribution

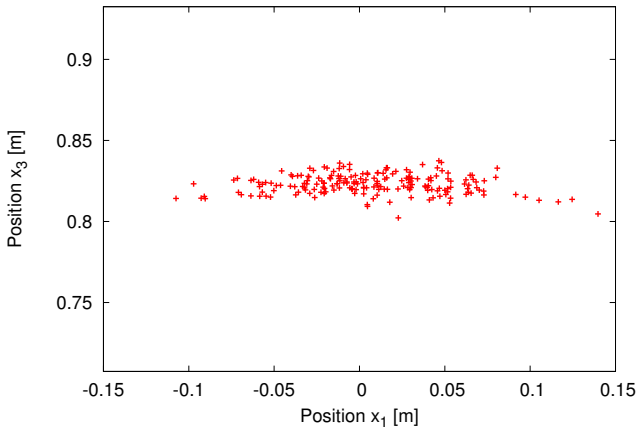


Figure 7.3: Effector positions for an i.i.d. variation of the nine actuator lengths. Already a deviation of $5mm$ on each actuator-length causes several centimeters sideward movement (x_1 axis) of the end-effector, but only small stretching movements in the top/down direction (x_3 axis).

around a stretched position $q_i = 0.225m \; \forall \; i = 1 \ldots 9$ with standard-deviation $5mm$ per actuator. Figure 7.3 shows the resulting positions of the end-effector from a sidewise perspective. The resulting distribution extends to almost $15cm$ sideward deviation (the first axis, x_1), while top/down stretching movements (x_3) only vary within $\pm 2cm$. The standard deviations of the generated distribution are $4.4cm$ in x_1 and x_2 direction and $0.6cm$ in x_3 direction. The large amplitude of sideward movements implies a very high sensitivity of the end-effector position to length-changes. In reverse, a positioning of the end-effector with low deviation (e.g. $1cm$) requires a control of the actuator lengths with *sub-millimeter accuracy*. This is clearly difficult to achieve on the BHA due to long delays in the pneumatic actuation, and strong sensory noise in the length-sensing (ca. $1mm$ amplitude).

In order to obtain a baseline how accurately the BHA's end-effector can be positioned, $P = 20$ entirely random postures q_p have been chosen. These postures were set as target for the length-controller, which had time to reach and stabilize each posture for 20 seconds. This procedure was repeated $R = 20$ times with different permutations of p. Each time the resulting cartesian end-effector position x_p^r has been recorded. The results are condensed by the distance of these positions from the average position per q_p:

$$\begin{aligned} \overline{x}_p &= \frac{1}{R} \sum_r x_p^r \; , \\ D^{rep} &= \frac{1}{P} \sum_p \frac{1}{R} \sum_r ||x_p^r - \overline{x}_p|| \; , \end{aligned}$$

Before the experiments

Pressure [bar]			Length [m]		
0	0	0	0.1825	0.1873	0.1834
0	0	**1.2**	0.1727	0.1782	**0.2513**
0	**1.2**	0	0.1748	**0.2681**	0.1749
1.2	0	0	**0.2545**	0.1757	0.1760
.....					
1.2	**1.2**	**1.2**	**0.2476**	**0.2647**	**0.2338**

After the experiments

Pressure [bar]			Length [m]		
0	0	0	0.1839	0.1870	0.1859*
0	0	**1.2**	0.1750	0.1783	**0.2581****
0	**1.2**	0	0.1754	**0.2709***	0.1744
1.2	0	0	**0.2615****	0.1761	0.1771
.....					
1.2	**1.2**	**1.2**	**0.2538****	**0.2654**	**0.2388****

Table 7.1: Measured actuation limits for the three parallel actuators in the third segment before and after the learning experiments. Changes of more than $2.5mm$ are marked with *, changes of more than $5mm$ with **.

where $||\cdot||$ is the euclidean norm. Results show that $D^{rep} = 0.0047m$. Hence, the end-effector can only positioned with approximately $5mm$ accuracy.

A central problem for the control of the BHA is that the limits, in which the actuator lengths can be controlled, are very narrow, but not exactly known. Limits for the pressure are easily formulated: each actuator has a minimum pressure of $0bar$. The maximum pressures are $0.9bar$, $1bar$, and $1.2bar$ for the first, second and third segment, so that the set of possible pressure combinations is a hyper-rectangle in nine dimensions. In contrast, the set of possible length combination is clearly not a hyper-rectangle. This is illustrated in the first part of table 7.1: combinations of min./max. pressure were supplied to the three actuators in the third segment, and the resulting three actuator lengths were recorded. Two effects are clearly visible:

1. The different actuators have different limits, even within the same segment, due to visco-elasticity and wear out effects. This is particularly visible in the last line of the table, where maximum pressure for each actuator generates significantly different lengths.

2. There are significant interdependencies between the limits of different actuators: The maximum reachable length (i.e. the length for maximum pressure) depends on the length of the other actuators.

Such combinations of min. and max. pressure give some insight into the structure of the length-ranges. Yet, the analytic shape of the set of possible length combinations is

not known. For the coordination problem that implies that not only f in not explicitly known, but also the action space $\mathbf{Q} \subset \mathbb{R}^9$ is unknown. Each vector in $\mathbf{Q}$ represents a length-combination that is reachable for the robot. Each vector that is not in $\mathbf{Q}$ can not be reached. $\mathbf{Q}$ is not only not known, it is *not stationary*. The upper part of table 7.1 was recorded before the experiments described in section 7.3. The same procedure has been repeated after the experiments and shows that the limits have changed significantly (lower part of table 7.1). For instance the maximum values in the last column have changed by 6-7mm which is substantially above sensory noise and can cause large changes of the effector positions. For practical experimentation with the BHA this means that whenever some posture q^* is desired, it is not even clear whether the posture *can* be reached.

7.1.3 Kinematic Coordination Problem

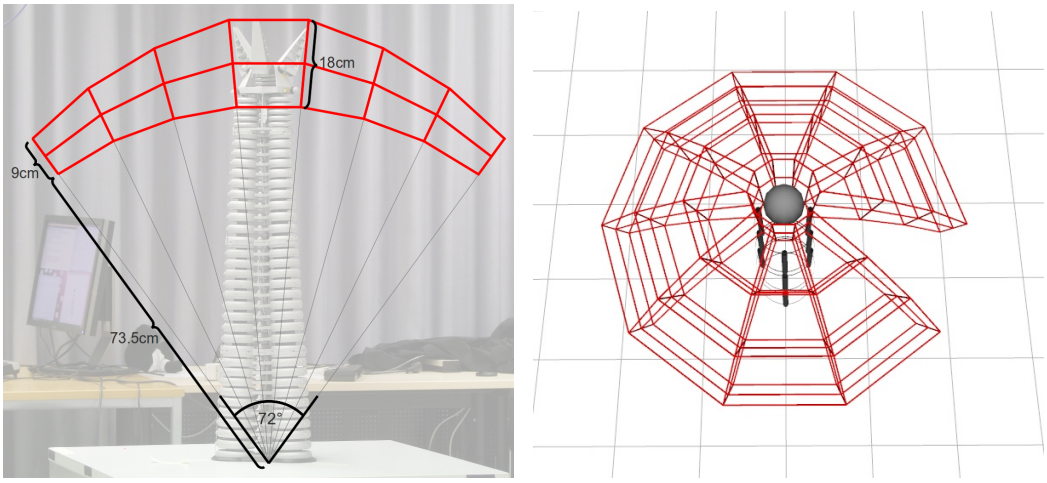
Reaching for some desired cartesian position $x^* \in \mathbb{R}^3$ with this robot means to find some posture, i.e. a combination of lengths q, that results in a end-effector position $x = x^*$. The following experiments consider the learning of reaching skills for the set of targets illustrated in figure 7.4. A side view is shown in figure 7.4(a): the target positions are the 24 vertices of the red grid, which is shown in relation to the BHA. A three-dimensional workspace is constructed from this plain grid by rotating it around five different angles. Figure 7.4(b) shows the resulting workspace in a 3D view from above. The overall set of targets $\mathbf{X}^*$ comprises $K = 120$ target positions x_k^*, which are the vertices of the three-dimensional grid. Note that the "gaps" in the 3D visualization are only for visual orientation. The goal of the experiments is to learn an inverse model $g(x^*)$ for the volume enclosed in this set of targets.

7.2 Online Goal Babbling Formulation

In order to solve the coordination problem on the BHA by means of goal babbling, the experiments in this chapter essentially rely on the exploration and learning formulation presented in chapter 6. However, several changes are proposed in order to deal with the BHA in an optimal way.

Action Execution and Observation An important difference between the simulation experiments presented throughout the previous chapters, and physical reaching on the BHA is that actions can not always be performed perfectly. Sending a set of desired lengths q^* to the length controller does not necessarily result in the exact achievement or these lengths, but in an actual posture q. The actions that are suggested by the goal babbling scheme are generally desired lengths q_t^*. Whenever a command q_t^* is sent to the controller, the outcome x_t can simply be observed. If the command q_t^* has not been achieved at the time of observation, both sizes do not correspond to a sample of the underlying forward function:

$$q_t^* \neq q_t \quad \Rightarrow \quad x_t \neq f(q_t^*) \ .$$



(a) (b)

Figure 7.4: An inverse model is learned for the red workspace, shown (a) from a sidewise perspective in 2D, and (b) from a top view in 3D.

For the exploration algorithm this implies that commands and actual postures must be strictly separated. Only the current observation x_t and sensed posture q_t reflect a causal, physical relation and therefore correspond to a sample of the forward function

$$x_t = f(q_t).$$

Hence, examples (x_t, q_t) are used for learning, instead of relying on q_t^* and learning from (x_t, q_t^*). In most situations the deviation between q_t^* and q_t along continuous movement paths is not very large. Yet, the distinction is very important *if* the deviation is large which is mostly caused by the narrow actuation ranges. The algorithm can only find correct solutions within these actuation ranges if they are also respected in the example data.

Internal Coordinate Representation While all evaluations in this chapter are performed in cartesian coordinates in order to provide easily understandable distances in meters, the learning is performed in a different coordinate system. Since the exploration is based on the sampling of continuous paths it is desirable to have a convex workspace, which allows to sample a linear path between any two points. In order to achieve that for the given set of targets, the proposed algorithm formulation uses an internal coordinate system for the observations and goals that is based on angular coordinates. Therefore the following transformation is applied before spatial coordinates $x = (x_1, x_2, x_3)^T$ are used for learning:

$$\psi(x) = (\ sgn(x_3) \cdot ||x||,\ \sphericalangle(x, \mathbf{u_1}),\ \sphericalangle(x, \mathbf{u_2})\ )^T\ ,$$

where $\mathbf{u_1}$ and $\mathbf{u_2}$ are the unit vectors along the first and second axis. The first component of $\psi : \mathbb{R}^3 \to \mathbb{R}^3$ represents the radial component, i.e. the distance of some point from the BHA's base. The last two components express angles. The set of targets covers a *convex set* after the transform so that linear paths can be sampled without leaving the set. This coordinate system is consistently used for exploration and learning by considering goals $\psi_t^* = \psi(x_t^*)$ and observations $\psi_t = \psi(x_t)$.

Path Generation with Homogeneous Velocity Chapter 6 has formulated the exploration of continuous point-to-point paths that have a fixed number of intermediate steps L. The advantage of this formulation is that it permits to use the number of movements as a coherent and intuitive measure of time and exploratory cost. However, it generates movements with highly inhomogeneous velocity because the distance between successive goals x_{t-1}^* and x_t^* depends on the distance between the end-points of the movement. Reasonable velocities for physical movement of the BHA are rather limited. Very rapid movements are only possible in a ballistic manner without control of intermediate behavior, so that it is advantageous to limit the velocity if coordinated behavior is desired. Very slow movements, on the other hand, can be obscured by a low ratio between actual movement and sensory noise.

In order to cope with this aspect, this chapter revises the path generation such that the distances between successive goals are constant, so that more homogeneous velocities are generated. Therefore the end-points of the movement are again chosen randomly from $\mathbf{X}^*$, but the number of intermediate steps varies. In the internal coordinate system this scheme is denoted as

$$\psi_{t+1}^* = \psi_t^* + \frac{\delta_\psi}{||\Psi_r^* - \psi_t^*||} \cdot (\Psi_r^* - \psi_t^*) \ . \tag{7.1}$$

where Ψ_r^* is the end-point of the movement and δ_ψ is the step-length between successive time-steps. If the last goal has been closer than δ_ψ to the end-point, the next goal is set to the end-point $\psi_{t+1}^* = \Psi_r^*$ and a new end-point Ψ_{r+1}^* is chosen for the next movement.

Each goal ψ_t^* generates an action, in the same way as described in the previous chapter, by "trying to reach" with the inverse estimate plus the variation term:

$$q_t^* \quad = \quad g(\psi_t^*, \theta_t) + E_t(\psi_t^*). \tag{7.2}$$

The generation of homeward paths through the action space $\mathbf{Q}$ occurs accordingly with a step-length δ_q:

$$q_{t+1}^* = q_t^* + \frac{\delta_q}{||q^{home} - q_t^*||} \cdot (q_l^{home} - q_t^*) \ . \tag{7.3}$$

If q_t^* has been closer than δ_q to the home posture, the next action is $q_{t+1}^* = q^{home}$, and a goal-directed movement is attempted afterwards.

Balancing of Variation Terms A final, minor re-formulation concerns the perturbation term E_t that generates variations of the inverse estimate. This term has the exact same linear form as described in the previous chapter:

$$E_t(\psi^*) = A_t \cdot \psi^* + b_t, \quad A_t \in \mathbb{R}^{m \times n},\ b_t \in \mathbb{R}^m \ .$$

However, the amplitudes of A_t and b_t are controlled separately instead of assigning the same amplitude σ to both of them. The update rule for both A and b is the same (see equations 6.9 and 6.11), but with amplitudes $\sigma^{(A)}$ and $\sigma_\Delta^{(A)}$ for the entries of A, and $\sigma^{(b)}$ and $\sigma_\Delta^{(b)}$ for the entries of b.

This allows to balance the exploration of new directions by A and new positions by b for varying numerical amplitudes of the goals. The goal positions in the BHA setup have a higher numerical amplitude than the cartesian positions of the planar arm setup in chapter 6. This would make the term $A_t \cdot \psi^*$ predominant in the perturbation term, but which can be balanced by separate amplitudes for A and b.

Algorithm Except the changes listed above, the algorithm used in this chapter has the same organization as the previous formulation in chapter 6. Learning and exploration start in the home posture. Continuous goal-directed paths are successively generated and the inverse estimate $g(\psi^*, \theta)$ is updated in each step. The examples (ψ_t, q_t) are weighted accordingly to the previous chapters. Including the notation of the internal coordinate system the weighting scheme is

$$\begin{aligned} w_t^{dir} &= \frac{1}{2}\left(1 + cos\sphericalangle(\psi_t^* - \psi_{t-1}^*,\ \psi_t - \psi_{t-1})\right) \ , \\ w_t^{eff} &= ||\psi_t - \psi_{t-1}|| \cdot ||q_t - q_{t-1}||^{-1} \ , \\ w_t &= w_t^{dir} \cdot w_t^{eff} \ . \end{aligned}$$

After a goal-directed movement has reached its end-point, a homeward movement is performed with probability p^{home}. The entire formulation is summarized in algorithm 3.

7.3 BHA Experiments

This section presents experiments with online goal babbling on the physical BHA robot. A first investigation illustrates how the reaching performance develops during learning. Then a method for local error correction is presented, which reduces residual errors due to non-reachable target postures.

The central measure of learning progress is again the *mean euclidean deviation*, which measures the distance between actual and desired *cartesian* positions:

$$D^X(\mathbf{X}^*, \theta) = \frac{1}{K} \sum_{k=0}^{k<K} ||f(g(x_k^*, \theta)) - x_k^*|| \ .$$

Algorithm 3 Modified Online Goal Babbling Formulation

Require: Home posture q^{home}
Require: Set of target positions: $\mathbf{X}^*$
  Initialize learner: $\theta \leftarrow \theta_0$ such that $g(\psi^*, \theta) = q^{home}$
  Initialize variation: $E_0(\psi^*)$
  Goals and actions: $\psi_0^* \leftarrow \psi^{home}$, $q_0^* \leftarrow q^{home}$
  The first movement is goal-directed: $G \leftarrow true$
  while true **do**
    if G is $true$ **then**
      Chose end-point x^* from $\mathbf{X}^*$, set $\Psi^* = \psi(x^*)$
      while $\delta_\psi < ||\psi_t^* - \Psi^*||$ **do**
        Interpolate new goal ψ_t^* (Equation 7.1)
        Generate goal-directed action $q_t^* = g(\psi_t^*, \theta) + E(\psi_t^*)$
        Apply q_t^* on the robot
        Observe resulting posture q_t and effector position ψ_t, compute weight w_t.
        Update inverse model with (ψ_t, q_t, w_t) and update variation
      end while
      Set $G \leftarrow false$ with probability p^{home}
    else
      while $\delta_q < ||q_t^* - q^{home}||$ **do**
        Interpolate action q_t^* towards home posture (Equation 7.3)
        Apply q_t^* on the robot
        Observe resulting posture q_t and effector position ψ_t, compute weight w_t.
        Update inverse model with (ψ_t, q_t, w_t) and update variation
      end while
      $\psi^* \leftarrow \psi^{home}$
      $G \leftarrow true$
    end if
  end while

Target step-length	δ_ψ	0.01
Posture step-length	δ_q	0.002m
Home probability	p^{home}	0.1
Perturbation amplitude	$\sigma^{(A)}$	0.0025
Perturbation amplitude	$\sigma^{(b)}$	0.005
Perturbation change-rate	$\sigma^{(A)}_\Delta$	$0.1 \cdot \sigma^{(A)}$
Perturbation change-rate	$\sigma^{(b)}_\Delta$	$0.1 \cdot \sigma^{(b)}$
Local learning distance	d	0.1
Learning rate	η	0.05

Table 7.2: Parameters used for exploration and learning.

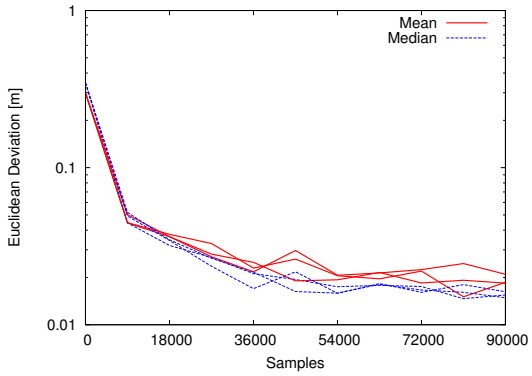
All experiments use the locally-linear regression formulation introduced in section 6.2.3. The parameter values used for the experiments are summarized in table 7.2.

7.3.1 Learning to Reach on the BHA

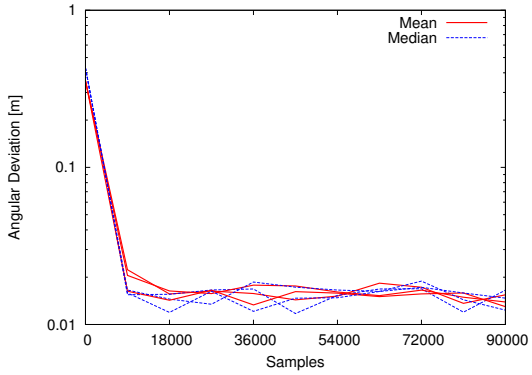
The exploration and learning algorithm is applied on the BHA in three independent trials. The set of targets $\mathbf{X}^*$ is used as illustrated in figure 7.4. The sampling rate on the robot is $5Hz$: in each second, five targets ψ_t^* are generated and the resulting samples are used for learning. With the target step length $\delta_\psi = 0.01$ in angular coordinates this corresponds to an approximate target velocity of $5\frac{cm}{s}$, which is suitable for the robot. The home posture q^{home} was set to a straight shape with a length of $0.225m$ for each actuator. In each trial, the method used $T = 90000$ samples, which corresponds to five hours real time.

Every 9000 samples the learning was interrupted in order to measure the current performance on the set of $K = 120$ targets shown in figure 7.4. The current inverse estimate $g(\cdot, \theta_t)$ was used to estimate the posture $q_k^* = g(\psi_k^*, \theta_t)$. The length controller had 20 seconds time to reach and stabilize q_k^*. Statistics of the euclidean deviations between the targets x_k^* and the actually observed positions x_k are shown in figure 7.5(a) for all three trials. The initial error is approximately $30cm$, which corresponds to the average distance of the home position, in which the learner is initialized, and the different target positions. Subsequently the exploration procedure reduces the error rapidly. After $T = 90000$ the errors consistently reach a mean level of ca. $2cm$ and a median level of ca. $1.5cm$ in all three trials. For an average robot-length of $80cm$ this corresponds to $2-3\%$ relative error, which already includes the general execution uncertainty of $5mm$ (see section 7.1.2). The learning clearly succeeds to bootstrap the reaching skill on the robot. The remainder of this section closely investigates the details of this performance curve, the reasons for residual errors, and how they can be removed by further exploitation of the learned inverse model with a feedback controller.

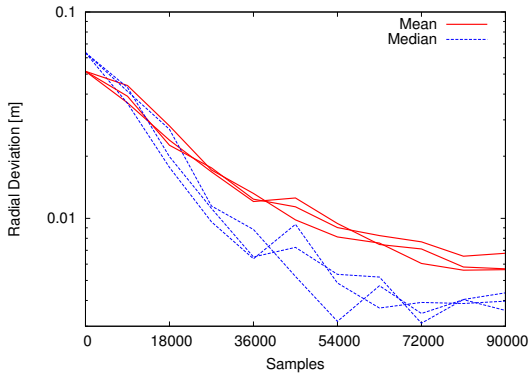
Figure 7.6 shows a more detailed view on the first trial. Histograms of the euclidean deviations are shown for $t = 0$, $t = 9000$, and $t = 90000$. The initial histogram simply



(a)



(b)



(c)

Figure 7.5: The mean euclidean deviation in cartesian coordinates (a) is reliably reduced in all three trials. The mean reaches approx. $2cm$ and the median value $1.5cm$. A decomposition into angular (b) and radial (c) components shows that the two-dimensional angular sub-problem is solved already within the first 9000 samples.

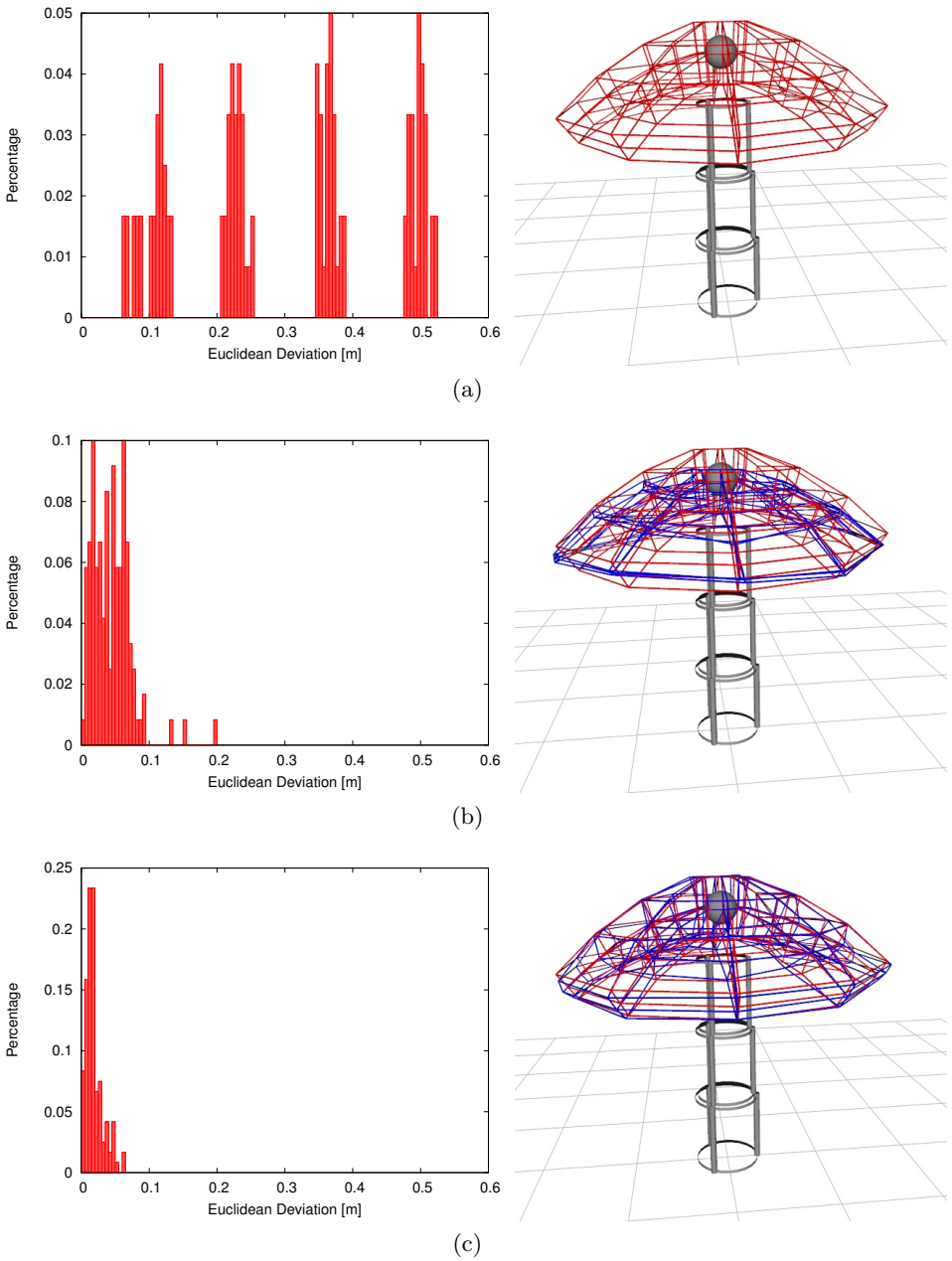


Figure 7.6: Histograms of the initial (a) euclidean deviations, and the deviations after $t = 9000$ (b) and after $t = 90000$ (c) samples. The initial histogram shows the ring structure of the target set, ongoing learning reduces the errors consistently. At $t = 9000$ the learner still has to make strong extrapolation, which lead to outliers (several points with errors above $10cm$), but which are consolidated by further learning.

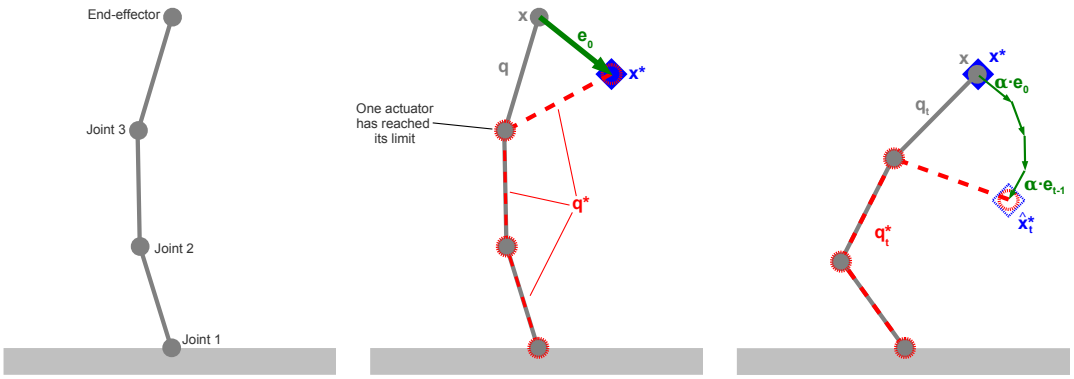
shows the distances of the initial posture from the four "rings" of the target grid. Further histograms show that the error is reduced continuously, but also that few, isolated targets show a comparably high residual error. The right side of the figure shows the behavior of the learner in the 3D space. The red grids again shows the set of targets. The blue grids show the measured behavior of the inverse estimate when trying to reach for the targets, i.e. the observed positions $x_k = f(g(\psi_k))$. Already after $t = 9000$ the positions are spread out along the angular directions, but do not yet cover the volume of the target set. After $t = 90000$ the learner has also discovered how to stretch along the radial axis, and target and actual grid are in good correspondence.

Stretching seems to be a simple movement on the robot: in a straight position all actuators need to be extended and the effector moves upwards. In fact, it is the most difficult movement: it requires a highly coordinated motor action, and the robot will deviate substantially if only one degree of freedom does not follow this movement. Due to the very restrictive actuation limits it is also necessary to include all three segments into the movement in order to reach from the very bottom of the workspace to the very top. In contrast, angular motions are much simpler and can be done in a lot of different ways. Due to the high sensitivity of the robot to movements in these directions (see figure 7.3) they are also easily discovered during autonomous exploration. Since the combination of goal-directed exploration and online learning forms a positive feedback-loop during the initial bootstrapping, the learner can basically master angular movements already after a few minutes. Radial stretching movements have lower sensitivity which implies a lower gain in the feedback loop. Hence, it requires more time to learn this movement direction.

This behavior occurs consistently over the three trials: Figure 7.5(b) and (c) show a decomposition of the euclidean deviation into *angular and radial components*. For the angular component, x_k and x_k^* are both projected onto the unit-sphere with radius $1m$, such that the radial component is erased, and the euclidean distance between the projected points is measured. This component is only evaluated for the central of the three target "layers" (see figure 7.4). The top and bottom layer are not considered in order to blend out the difficulties of stretching movements for this evaluation. The radial error is the difference between the first components of $\psi(x_k)$ and $\psi(x_k^*)$, which is evaluated for all target positions. The plots show that the angular error component is reduced from $30cm$ to $2cm$ already in the first evaluation episode, and further stabilizes around $1.3cm$. The bootstrapping and fine-tuning of radial movements takes significantly more time in all three trials. The difficulty to discover (and also control) stretching movements, while other directions are that simpler to find is very specific for the BHA's trunk morphology that combines bending and stretching. After all, this problem is solved by the exploration procedure.

7.3.2 Local Error Correction

While the average performance during learning quickly reaches a good level, there remain rather isolated outliers. This behavior is particularly visible in figure 7.6(b), where few targets are only reached with an error of more than $10cm$. These outliers



(a) (b) (c)

Figure 7.7: Cartesian feedback control on a simple robot with three revolute joints (a). If an inverse model suggests a posture that can not be executed due to actuation ranges (b), a shifting of the target position allows to exploit the redundancy and nevertheless reach the target (c).

are largely consolidated during learning, but a "heavy tail" in the error-histogram remains (see figure 7.6(c)). The reason for this behavior is grounded in the inevitable process of generalization and interference inside the regression of g. During the initial bootstrapping of a motor skill this is an enormously useful mechanism: already based on the first examples x the learner generalizes and makes extrapolations for other targets x^*. These extrapolations are, of course, not perfect but allow a quick coverage of the workspace.

Once the learner has roughly covered the workspace, this interference can become more problematic due to the highly constrained actuation limits of the BHA. Moving through the entire set of targets requires to operate very closely to the limits of the possible length configurations $\mathbf{Q}$. Any data used for learning lies inside $\mathbf{Q}$ since the values of q_t have been observed on the robot. Interference, however, can cause a projection of g beyond $\mathbf{Q}$ for other positions x than that one currently used for learning (x_t). Suppose the current learning step is done on an example (x_t, q_t). Due to interference, the learner's output is changed at another position $x \neq x_t$ to $g(x) = q^*$ and $q^* \notin \mathbf{Q}$. When the inverse estimate is now used to reach for x, it would suggest q^* which is not reachable. On the robot, this results in a different posture q. This mismatch $q^* \neq q$ is referred to as an *execution failure*. Due to the high angular movement sensitivity of the BHA already minor execution failures cause large deflections of the end-effector, and thus high deviations of the cartesian effector position. The tight connection between cartesian deviations and execution failures is shown in table 7.3. The upper part shows the final deviations in cartesian coordinates for all three trials.

Feedforward Control with Learned Model

	Mean D^x $[m]$	Median D^x $[m]$	Failure-Corr.
Trial 1	0.0186	0.0155	0.832
Trial 2	0.0184	0.0149	0.728
Trial 3	0.0209	0.0162	0.845

Additional Feedback Control

	Mean D^x $[m]$	Median D^x $[m]$	Failure-Corr.
Trial 1	0.0074	0.0067	-0.045
Trial 2	0.0088	0.0080	0.101
Trial 3	0.0071	0.0064	0.017

Table 7.3: Euclidean deviations in cartesian coordinates without, and with cartesian feedback control on top of the inverse model. The controller removes errors induced by execution failures, as indicated by the erased failure correlation.

The last column shows the *failure correlation* for the final evaluation after $t=90000$:

$$C_x^q = \varrho\,[\ ||x_k^* - x_k||,\ ||q_k^* - q_k||\ ]_k\ , \tag{7.4}$$

where $\varrho \in [-1:1]$ is the *Pearson correlation coefficient*. It measures how well deviations in cartesian coordinates are correlated with the occurrence of execution failures. The table shows very high positive correlation in all three trials, which indicates that the largest deviations are indeed caused by execution failures.

Although the interference is rather limited by the locally linear learning, it is sufficient to cause the heavy-tailed error-distributions. Also, the projection outside $\mathbf{Q}$ is hardly avoidable, since $\mathbf{Q}$ is not even known and changes during operation. The final experiment on the real BHA shows how the impact of such execution failures can be mitigated by means of an additional feedback controller. Figure 7.7(a) illustrates a simplified domain with a planar robot arm comprising three revolute joints. A learned inverse model is used to reach some target position x^* (figure 7.7(b)). The suggested posture q^* would indeed solve the task, but is not executable since the last joint has reached its actuation limit and can not be bent further downwards. The resulting posture q ends up in a position $x \neq x^*$. When an inverse model g has been established, feedback control can be applied without further learning by applying *cartesian corrections*: the target position is virtually shifted towards some value $\hat{x}_t^*$ and the posture $q_t^* = g(\hat{x}_t^*)$ is applied on the robot, which results in a posture q_t and an effector position x_t (see figure 7.7(c)). The shifting of goals thereby follows the currently observed cartesian error $e_t = x^* - x_t$:

$$\hat{x}_0^* = x^*\ , \quad \hat{x}_t^* = \hat{x}_{t-1}^* + \alpha \cdot e_{t-1}$$

This procedure is guaranteed to converge to the target position $\hat{x}_t^* = x^*$ if a shift of targets $\alpha \cdot e_{t-1}$ always results in an actual effector movement that has a positive angle

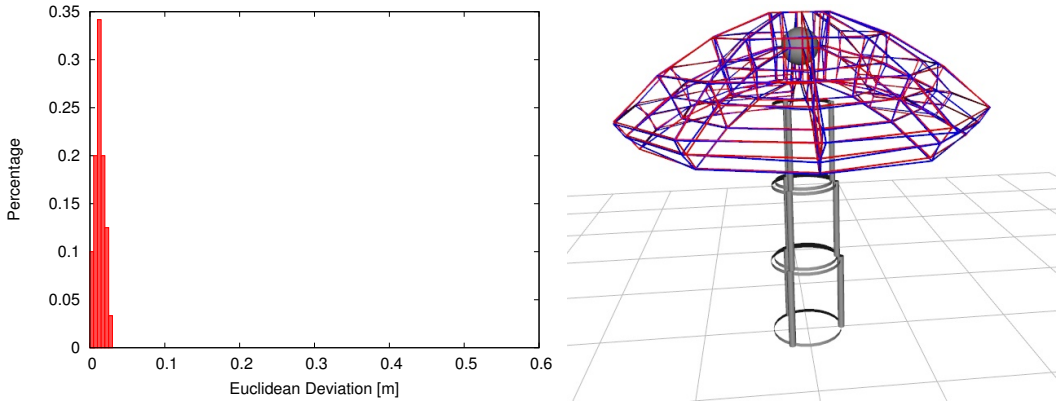


Figure 7.8: Cartesian performance of a learned model when cartesian feedback control is applied on top (compare figure 7.6(c)).

to the desired movement ($\sphericalangle(e_{t-1}, x_t - x_{t-1}) < 90°$). If, however, the inverse estimate is not able to generate a positive movement direction, the control can diverge. It is possible that the limited actuator is driven even deeper into its limit during this feedback-controlled movement, since also the feedback-controller is not aware of $\mathbf{Q}$. However, an important strength of the goal babbling methodology introduced in this thesis is that it can incorporate many degrees of freedom. The weighting scheme based on movement efficiency then causes learning to efficiently distribute movements over all actuators. This behavior can be exploited by the feedback control, even if one actuator is blocked. As long as other actuators are movable, the inverse estimate involves them to reach for $\hat{x}_t^*$ which brings the observed effector position x_t closer to x^*.

The final inverse estimates of all trials are evaluated with this procedure. For each target ψ_k^*, the initial inverse estimate $q^* = g(\psi_k^*)$ was sent to the length controller and was active for five seconds before the feedback control was activated. Then the feedback control on top of g was applied with $5Hz$ and gain $\alpha = 0.02$ for 15 seconds, so that the overall evaluation time per target was 20 seconds, consistently with other evaluations in this chapter. Results for the first trial are shown in figure 7.8: the heavy-tail in the error histogram has disappeared (compare figure 7.6(c)) and the maximum error is below $3cm$. The performance in 3D shows an excellent match between targets x_k^* (red) and actual positions x_k (blue).

Results for all three trials are shown in table 7.3 (bottom). The mean euclidean deviations are reduced to $7 - 9mm$ and the median deviations to $6 - 8mm$, which is a substantial improvement and close to the accuracy baseline of $5mm$. While the amplitude of execution failures (not shown) is not reduced by the feedback control, the *failure correlation* has dropped to zero. No divergence of the feedback control was observed in the experiment. These results clearly show that the combination of a kinematically efficient inverse estimate that exploits all degrees of freedom, and a cartesian feedback controller can cope with the problem of execution failures.

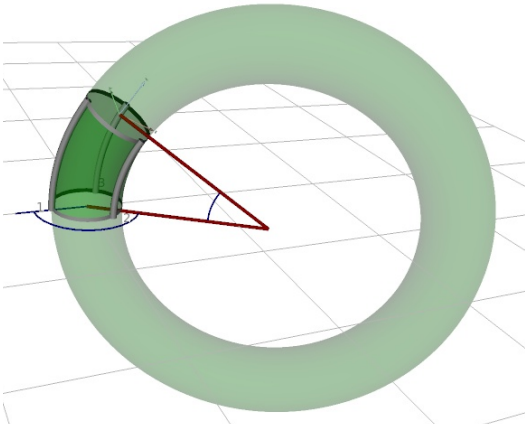


Figure 7.9: The simulated BHA models each segment of the robot as torus section.

7.4 Non-Stationary Behavior in Simulation

The experiments on the physical BHA have shown the success of the goal babbling method for the trunk morphology. Thereby a significant change of the actuation ranges $\mathbf{Q}$ has been observed. Other changes like drifting sensors or slight changes of the true forward function f due to visco-elasticity are known to occur but are hard to capture. This section complements the previous experiments with learning in a simulated environment in which such non-stationary behaviors can be controlled.

7.4.1 Kinematic Simulation of the Bionic Handling Assistant

An open source implementation [Rolf, 2012] of a *constant curvature continuum kinematics* model is used in order to simulate the kinematics of the BHA. This model assumes that bending and stretching movements of each robot segment behave like a torus section (see figure 7.9), which allows to infer the coordinate transformations for the forward kinematics. The model allows to predict the end-effector position x of the BHA based on the actuator lengths q with an average accuracy of $1cm$ [Rolf and Steil, 2012a]. Instead of applying a length on the robot, the end-effector position is simply computed with this library: $x = f^{sim}(q)$.

An important aspect of the true kinematic problem on the BHA are the actuation ranges $\mathbf{Q}$. Since this set is also not known analytically, the minimum/maximum pressure results recorded on the real BHA (see table 7.1) are used to simulate it. The eight combinations of min./max. pressure were recorded for each segment separately. The possible length combinations $\mathbf{Q}^{sim}_{(i)} \in \mathbb{R}^3$ for a segment i are modeled by the *convex hull* of the resulting eight lengths. The possible lengths of different segments are assumed to be independent: $\mathbf{Q}^{sim} = \mathbf{Q}^{sim}_{(1)} \times \mathbf{Q}^{sim}_{(2)} \times \mathbf{Q}^{sim}_{(3)}$. When the exploration suggests some

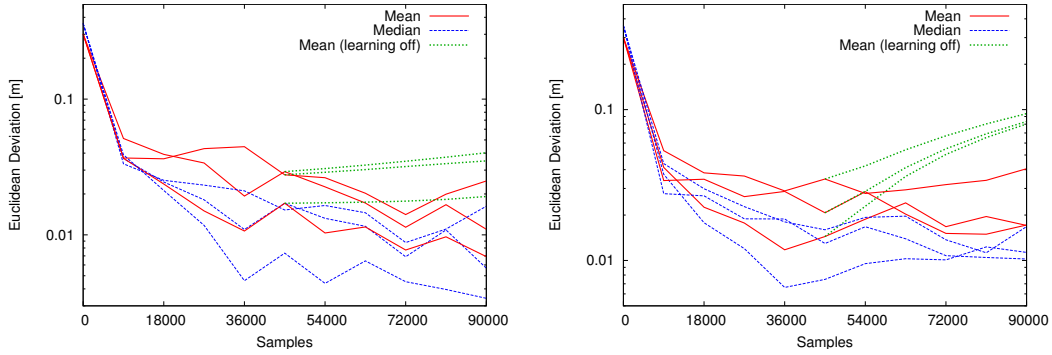


Figure 7.10: Performance for (a) shrinking ranges and (b) for drifting sensors.

posture $q^* \notin \mathbf{Q}^{sim}$, it is projected onto the surface of $\mathbf{Q}^{sim}$:

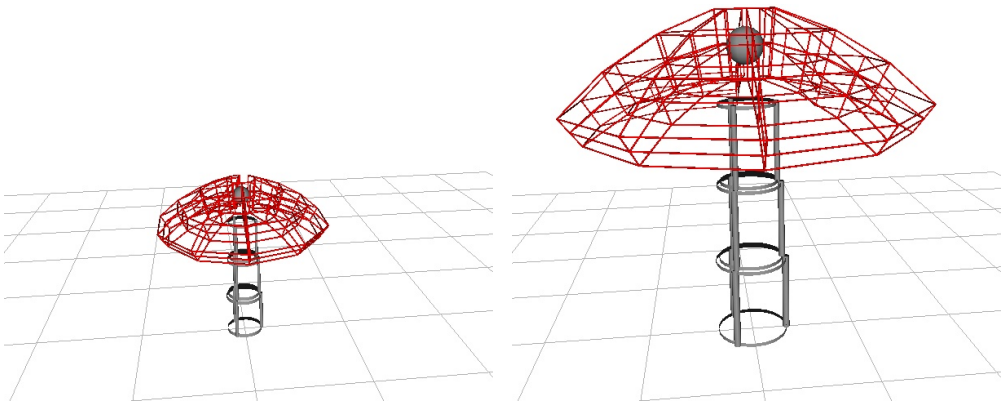
$$q = c(q^*) = \begin{cases} q^* & \text{if } q^* \in \mathbf{Q}^{sim}, \\ \underset{\hat{q} \in \mathbf{Q}^{sim}}{\operatorname{argmin}} ||q^* - \hat{q}|| & \text{else.} \end{cases}$$

7.4.2 Varying Ranges and Sensory Drifts

The first investigation concerns varying actuation ranges $\mathbf{Q}^{sim}$. Three independent trials are performed for a direct comparison with the real BHA results, each with $T = 90000$ examples and parameters identical to the previous experiments. Learning initially runs on a stationary system for $T_{(s)} = 45000$ examples. Between $T_{(s)} = 45000$ and $T = 90000$ the ranges of two actuators are continuously reduced. Both the minimum and maximum values are narrowed by 30% for the first actuator of segment 2 and the second actuator of segment 3. The progress of this narrowing is linear in t. The evaluation shows how the learning procedure can deal with this change, as well as how the performance develops if learning is stopped at the onset of non-stationary behavior. Results are shown in figure 7.10(a). Ongoing learning reduces the error even after the onset of change. When learning is turned off, the error increases slowly. While the increase of the average error is comparably mild, there are drastic differences in the maximum errors over the target set $\mathbf{X}^*$: The first simulated trial exposes a maximum error of $5.5cm$ after $T = 90000$. When learning is turned off, the same trial results in 10% of the target positions with more than $10cm$ error (max. $20cm$).

A different kind of non-stationary behavior on robotic systems is the drift of sensor values, when the physical sensors are not repeatedly calibrated. Such behavior of the BHA is modeled in simulation by defining a *drift function* $d : \mathbb{R}^9 \to \mathbb{R}^9$ that distorts the measurements of the actuator-lengths:

$$d(q) \quad = \quad (\mathbb{1}_9 + \beta \cdot \mathtt{diag}(\vec{s})) \cdot q + \beta \cdot \vec{o}\,,$$



(a) (b)

Figure 7.11: Simulation of morphological growth of the BHA from half its size (a) to its full scale (b).

where $\vec{s}$ and $\vec{o}$ are a linear distortion. β allows to scale its impact. When the learner operates with a length q, the "true" lengths with respect to effector position and ranges are $d(q)$:

$$f'(q) = f^{sim}(d(q)) \ , \quad c(q^*) = d^{-1}\left(\ c\left(d(q^*)\right) \ \right) \ .$$

Again, three trials are simulated over $T = 90000$, with a sensory drift beginning at $T_{(s)} = 45000$. The entries of $\vec{s}$ and $\vec{o}$ where drawn from a normal distribution with deviation 0.05 independently for each trial. The drift amplitude β was linearly scaled from 0.0 to 1.0 between $T_{(s)}$ and T. Results are shown in figure 7.10(b). Without learning, the error increases significantly and reaches a average level of $8 - 10cm$. With enabled learning the error is approximately stabilized, although the amplitude and rate of the drift are too strong to further reduce the error as in the previous experiment.

7.4.3 Morphological Growth

The last experiment deals with a non-stationary behavior that is, in particular in its amplitude, not a problem on the real BHA, but shows that the method can deal with even more drastic changes. Learning is performed on a *growing* simulation of the BHA. The simulation starts with a BHA that is scaled to half of its original size and grows to full size between $T_{(s)} = 45000$ and $T = 90000$ (see figure 7.11). The change goes on linearly and concerns the radius of the simulated segments, the actuation ranges, as well as the reachable workspace. In order to assess the learning performance for a workspace with varying size, the resulting errors are normalized to $\frac{1}{\gamma}D^X$, where $\gamma \in [0.5; 1.0]$ is the current relative size of the simulated BHA. The results (figure 7.12) show that the performance without learning degenerates to almost initial error values.

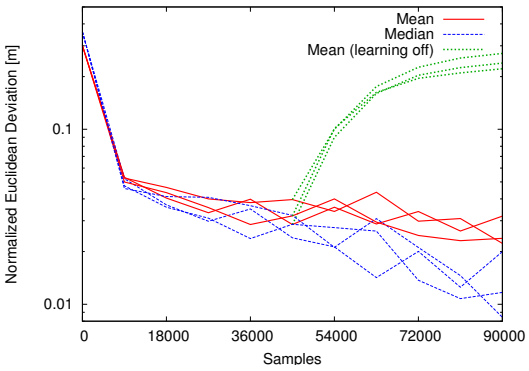


Figure 7.12: When learning is turned of during morphological growth, the error increases rapidly. Ongoing learning stabilizes and slightly reduces the error.

With enabled learning the median error is nevertheless decreasing, while the mean error is approximately constant. This experiment generates the largest gap between learning and non-learning during non-stationary behavior. Although the morphological change extinguishes the learned performance when learning is turned off, the change seems to comparably easy to track during learning.

7.5 Discussion

The experiments in this chapter demonstrate that online goal babbling allows to bootstrap the inverse kinematics of the pneumatically actuated Bionic Handling Assistant. The method is robust enough to cope with the inherent sensory noise, delays during the execution, and the varying actuator ranges. The successful learning of reaching skills is an important milestone for the practical applicability of such systems in real world scenarios. The method is thereby fast enough to perform on the robot in reasonable time. 90000 samples along continuous movement paths were used during the experiments. This corresponds to approximately 1000 crossings of the three-dimensional cartesian workspace, which is a plausible result considering the 100 movements necessary for two-dimensional tasks (compare chapter 6), and an excellent result considering the additional challenge of non-stationary system behavior. The algorithm has used nine degrees of freedom of the robot. Compared to the previous experiments in this thesis this is only mildly high-dimensional, but clearly high-dimensional enough to make exhaustive exploration unfeasible. The scalability experiments in chapter 6, together with the practical demonstration in this chapter, suggest that the method can also be practically useful in much higher-dimensional domains.

The learned coordination skill represents a direct, “feedforward” mapping from desired effector position to actuator lengths. Residual inaccuracies are unavoidable for feedforward control schemes. Yet, the experiments have shown that such errors can be handled with an additional cartesian feedback controller if necessary. The controller

exploits the learner's efficient use of all actuators, which even allows to correct errors that are caused by the narrow actuation ranges. The use of feedforward control is highly beneficial for a pneumatic robot: delays usually only allow to apply feedback-control with very low gains, which implies slow movements. A feedforward controller can quickly estimate the necessary motor commands which can be applied immediately. This is particularly useful due to the narrow actuation ranges, for which the learned model has already stored valid solutions while a pure feedback controller needs to search for them newly during each movement.

Besides learning on the non-stationary robot, the experiments have shown in simulation how the method copes with various changes like changing ranges, drifting sensors and even morphological growth. For each of these setups the performance degenerates significantly without learning, but is stable or improves for ongoing learning. Goal babbling provides an elegant way to deal with such behavior because it defines an incremental and ongoing process. This process is always based on currently observed data, and thus grounded on the current system behavior. Most importantly, it does not require an exhaustive exploration of the motor system. This could not even be done once on robots with many degrees of freedom like the BHA, so that a tracking of ongoing changes would even conceptually not be possible. Goal babbling discards redundant choices if multiple actions exist to solve the same goal. Hence, it only samples a low-dimensional sub-manifold in the space of motor commands, which can be quickly explored. Online learning then quickly reacts to a changing environment and allows to adapt to changes efficiently.

Chapter 8

Conclusion

8.1 Summary

High-dimensional motor systems can not be fully explored within the lifetime of any learning agent. Based on this simple observation, this thesis has criticized conventional methods to bootstrap coordination skills based on exhaustive exploration, and formulated the overarching goal to develop concepts and methods that allow for an efficient bootstrapping of reaching skills in high-dimensional domains. Inspired by infant developmental studies, which show that already newborns attempt goal-directed actions, the first goal was to conceptualize and understand early goal-directed actions as a bootstrapping mechanism for coordinations skills. For this purpose, the concept of *goal babbling* was introduced in chapter 3. Experimental results throughout this thesis show that goal babbling indeed allows for a successful, scalable, and rapid bootstrapping of coordination skills.

This thesis has investigated how goal babbling can be implemented in order to learn inverse models from examples. In redundant domains, inverse models select one of the infinitely many ways to achieve some goal, which provides the chance to solve a coordination skill without knowing all solutions. Chapters 4 and 5 have demonstrated the validity of this idea with theoretical and experimental results. Thereby the second goal has been achieved, i.e. to enable the learning of inverse models from examples at all. Chapter 4 has investigated the theoretical basis of example-based learning of inverse models, and provided several important results for linear domains. It has been shown that the learning gradients of error-based and example-based learning have a non-negative angle, and that learning from examples always leads to the acquisition of at least partial solutions in linear domains. Using that theoretical framework, new failure modes for goal-directed exploration without noise [Sanger, 2004] have been shown. Most importantly, a proof was given that, if noise is added to goal-directed exploration, goal babbling does not only find a valid solution and thereby solve the inversion of causality, but results in the unique least-squares solution. Chapter 5 showed that also the non-convexity problem [Jordan and Rumelhart, 1992] can be solved by means of goal babbling and provided the first algorithm that can learn inverse models from examples when non-convex solution sets are present. Based on the information structure of goal-directed exploration, it has been shown that inconsistent solutions can occur only in very characteristic ways, which can be detected and removed by means of a simple weighting scheme. While the non-convexity problem appears to be

a rather technical aspect, this solution sheds an interesting light on the concept of goal babbling. Goal babbling is a more *structured* approach to exploration than random motor babbling, which can be successfully exploited by using goals as a reference structure against which observed movements can be checked and assessed. Based on that solution, a first experimental proof of scalability was provided by showing the success in up to 50 dimensions for reaching with a planar arm. The generality of the algorithm was shown by the learning of full-body reaching on a humanoid morphology.

The third goal was to devise a scalable, fast, and practical algorithm that is applicable in real-world scenarios. Chapter 6 presented an online learning formulation of goal babbling and showed that it can bootstrap inverse models for planar robot arms within few hundred movements, which is competitive to human learning [Sailer et al., 2005]. This speed of learning is enabled by the close interplay of goal-directed exploration and learning. Learning from one example leads to a better estimate in the next exploration step, which in turn allows for a more effective learning step. The experiments show that an increase of the learning rate results in a speedup that is proportionally greater than the actual increase of the learning rate. This result shows that goal babbling constitutes a positive feedback loop in which learning and the generation of useful examples reinforce each other during bootstrapping. Measurements of the learning speed in relation to the dimension of the action space show that the algorithm scales with virtually constant exploratory cost between two and 50 degrees of freedom, with only slowly increasing cost for the slowest trials. Chapter 7 has shown that the method allows to bootstrap the inverse kinematics of the BHA, which is a practically relevant use-case due to the lack of analytical knowledge about this robot, and the difficulty to deal with its narrow and constantly changing actuation limits. The experiments show that the method can deal with sensory noise and delays as well as non-stationary behavior, which also provides the practical proof that goal babbling allows for a successful and efficient bootstrapping of coordination skills.

8.2 Discussion

Learning inverse models is a fundamental task in sensorimotor learning. If inverse models can be obtained, they directly represent a solution of a coordination problem. This solution can be applied without iterating toward some position and relying on possibly noisy or delayed feedback. This property is clearly useful for reaching on pneumatic robots like the BHA, but may be even fundamental for other coordination problems like facial expressions [Wu et al., 2009] or golf [Sanger, 2004], in which feedback takes substantial time or which must be solved in the first trial without iteratively approaching the goal. This thesis has introduced the first algorithm that can learn such inverse models from self-generated examples, in spite of non-convex solution sets. Compared to other approaches to the solution of coordination problems, the particular technological strength of the methods developed in this thesis is therefore their *simplicity*: Firstly, learning obtains an immediate solution of the coordination problem, instead of relying on complex analytical search or inversion mechanisms that

typically work on top of learned forward models. Secondly, the learning process itself uses simple example-data, which is technically straightforward and can be achieved with any function approximation scheme. In contrast to learning with distal teacher [Jordan and Rumelhart, 1992] and Jacobian-based methods [Sun and Scassellati, 2004], the learner does not have to be differentiated, which allows to treat the function approximation as a black box and permits to use learners for which input-to-output derivatives can not be trivially obtained [Jäger, 2001]. Thirdly, goal babbling defines an incremental and ongoing bootstrapping mechanism. In contrast to strictly staged approaches like motor babbling, this does not require a delicate decision when to stop exploration and to start goal-directed behavior.

In contrast to random motor babbling strategies, goal babbling provides a consistent approach for high-dimensional systems. Goal babbling leaves out redundant choices of actions and focuses on behaviorally relevant data. Experiments in this thesis have shown that the online algorithm displays enormous scalability with almost constant cost, which is opposed by the exponential cost of exhaustive exploration. Several very recent studies have already adopted the goal babbling concept based on [Rolf et al., 2010c, 2011]. The studies confirm the direct superiority of *goal babbling over motor babbling* by showing that it needs less examples for a successful bootstrapping, or yields better performance after a fixed number of examples [Jamone et al., 2011, Stalph and Butz, 2012, Hartmann et al., 2012]. A direct numerical comparison between both approaches is, of course, only possible when motor babbling is applicable at all, despite being potentially inefficient. This is not the case for inverse models which can not be learned from random data sets.

A comparison between *different implementations of goal babbling*, that have been recently proposed, shows that the online algorithm presented in chapter 6 clearly outperforms other published results. The approach memory-based approach with goal babbling [Baranes and Oudeyer, 2010b] was shown to require several ten to hundred thousand examples for the 2D coordination of planar robot arm with $m = 15$, which is substantially more than the few hundred movements (few thousand examples) necessary for the algorithm in this thesis, and includes a more complex organization of movements. Jamone *et al.* presented quantitative results for a simple 3D reaching problem with $m = 7$, which likewise requires several million examples [Jamone et al., 2011]. Stalph *et al.* presented a model based on goal babbling that learns the 3D control of an anthropomorphic arm with $m = 7$ and showed that bootstrapping can be successful within few hundred thousand examples [Stalph and Butz, 2012]. This result is closest to the several ten thousand examples required for the BHA, as presented in chapter 7. Yet, Stalph's setup did not involve such particular challenges like narrow actuation ranges and directional inhomogeneities like on the BHA. How these different implementations of goal babbling relate in higher dimensions is currently not clear, since none of the studies has systematically investigated how the exploratory cost depends on the dimensionality of the action space. It is, however, plausible that the exploratory cost of forward-model-based approaches increases faster than presented for the learning of inverse models in this thesis. Even if forward models are learned only in a certain regime, their input dimension increases with m, such that more ex-

amples are locally necessary for coordination in high dimensions, while inverse models have a low input dimension n.

On a conceptual level, the questions whether goal babbling works at all and whether it is an enabler for efficient bootstrapping can clearly be answered positively. Two important mechanisms can be identified to enable goal babbling: firstly, plain goal-directed exploration gets stuck in partial solutions, which makes *exploratory noise* mandatory. This thesis has used a noise formulation that distorts the inverse estimate. Other approaches have used explicit random action between goal-directed movements [Baranes and Oudeyer, 2010b], explicit random actions locally around the action suggested by the inverse estimate [Schenck, 2008], additive gaussian noise on corrective actions [Stalph and Butz, 2012], or entirely random corrective action in case of previously unexplored regimes [Jamone et al., 2011]. Secondly, a *regularization* mechanism is necessary that prevents arbitrary drifts into unexplored regions of the action space. The work presented in this thesis, and the model in [Baranes and Oudeyer, 2010a] have independently proposed the use of a home posture, to which the learner returns after a while and is thereby driven into a known regime. Another approach has been presented in [Jamone et al., 2011], which uses nullspace projections to stabilize the regime of corrective actions, which is a well known method for the analytical control of robots [Liegeois, 1977].

Two important aspects have been identified as observable characteristics of goal babbling: firstly, its *structuredness* compared to random explorations is a direct consequence of the attempt to solve behavioral goals. This very structure has been used to solve the non-convexity problem in chapter 5. Secondly, goal babbling unfolds a *positive feedback loop*, which enables a very rapid bootstrapping. Such feedback loops in exploratory learning have not been described before, although they are arguably present in any scenario in which learning and exploration are coupled in a way such that learning improves exploration and vice versa. This is particularly clear for exploitation-based algorithms in reinforcement learning. A possible reason why this has not been observed is that the amplitude of the speedup depends on how rapidly a learner can generalize and in particular generate extrapolations. If a learner exposes only slow or little generalization, the next exploration step will only marginally benefit from learning and the positive feedback loop is practically inhibited.

Concludingly, this thesis has introduced the concept of goal babbling, and showed that it allows to solve coordination problems in high dimensions. Several theoretical results for linear domains have been proven. The methods developed in this thesis provide the first successful approach to learn inverse models from examples when non-convex solution sets are present. The online algorithm substantially outperforms other published methods and can perform with human-level learning speed. Besides various theoretical and technological advances, and the practical usage on the BHA, an important contribution of this thesis is the concept of goal babbling itself, which provides new vocabulary to foster research on sensorimotor learning, and has already started to do so.

8.3 Outlook

In order to focus on the very principles of scalable sensorimotor learning and to enable the learning of inverse models from examples, several practical and theoretical topics have not been directly addressed, and new questions arise from the present work. For instance, this thesis has not investigated the generation and selection of goals themselves. A method that approaches the dynamical selection of goals based on active learning has been presented in [Baranes and Oudeyer, 2010a]. Recent progress along these lines has been generated by active learning formulations based on competence-progress [Oudeyer and Kaplan, 2008], i.e. trying to generate examples that improve the mastery of a skill, opposed to knowledge-based formulations that seek for generic new information, despite being potentially irrelevant for a certain task. A related problem that is largely unsolved for high-dimensional scenarios is to discover what observations are possible, instead of predefining a set of reachable goals. The demand for a known set of goals is an important restriction of current algorithmic approaches to goal babbling. Competence-progress approaches make a first step in that direction because they detect that unachievable goals to not result in progress, but still require to specify an explicit, and preferably small, superset of achievable observations. In low-dimensional domains this problem can be solved by simple motor babbling which will generate all possible outcomes. Enabling this *discovery* of possible outcomes, and therefore potential goals, in high-dimensional domains is an important objective for future work.

This thesis has focused on the learning of exactly one valid solution to the coordination problem by means of learning an inverse model. This approach consistently tackles the problem that not all actions can be explored in the lifetime of an agent. If not everything can be explored, it is likewise impossible that all actions can ever be exploited. However, it can be necessary to know more than one solution in order to react to varying contexts or environments. A typical scenario is reaching with dynamical obstacle avoidance, which demands for an execution of movements that navigate around the obstacle. Approaches that learn forward models or corrective actions potentially express solutions to react to such circumstances, but solve the problem only partially. The agent does not only have to know multiple solutions, it has to select them in an appropriate and context-aware manner. In the case of obstacle avoidance this has only been achieved with complex analytical schemes that require substantial knowledge about the geometry of the obstacle and the own body [Khatib, 1986, Park et al., 2008]. Future work in this direction needs to investigate how exploration schemes can discover alternative solution when they are needed, and how they can be represented and leveraged in a task-appropriate manner. A related problem is to deal with discrete branches of solutions. Exploratory learning that is based on incremental and local updates like goal babbling can not directly discover such different branches. Coordination approaches that use corrective actions can represent such solutions, but typically not exploit them because they get stuck in intermediate local minima. Associative approaches like [Reinhart and Steil, 2011] appear to be suitable approach for the representation and exploitation of such branches, but efficient exploratory mechanisms

have to be investigated to generate appropriate training data.

The theoretical work in this thesis has mostly concerned linear domains. The extension to non-linear domains is an obvious and important direction for future work. The linear analysis, and the analysis of the information structure of possibly inconsistent solutions during goal babbling, establish the necessary ground for a deeper investigation. Relevant questions in this area clearly concern the theoretical soundness of the algorithm in non-linear domains. Another important aspect are the online learning dynamics. Even for learning with fixed data sets, online learning is barely theoretically understood when examples are not chosen independently and randomly [Biehl and Schwarze, 1995, Sollich and Barber, 1996]. A better theoretical understanding of these aspects could not only support empirical results, but also connect this research to other domains of machine learning and lead to the development of improved algorithms that exploit the learning dynamics in an optimal manner.

Bibliography

M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida. Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, 2009.

C. G. Atkeson. Learning arm kinematics and dynamics. *Annual Reviews Neuroscience*, 12:157–183, 1989.

C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

J. Baillieul. Kinematic programming alternatives for redundant manipulators. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1985.

J. Baily. Adaptation to prisms: Do proprioceptive changes mediate adapted behaviour with ballistic arm movements? *The Quarterly Journal of Experimental Psychology*, 24(1):8–20, 1972.

A. Baranes and P.-Y. Oudeyer. Robust intrinsically motivated exploration and active learning. In *IEEE Int. Conf. Development and Learning (ICDL)*, 2009.

A. Baranes and P.-Y. Oudeyer. Maturationally-constrained competence-based intrinsically motivated learning. In *IEEE Int. Conf. Development and Learning (ICDL)*, 2010a.

A. Baranes and P.-Y. Oudeyer. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2010b.

A. Baranes and P.-Y. Oudeyer. The interaction of maturational constraints and intrinsic motivations in active motor development. In *IEEE Int. Joint Conf. Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2011.

A. Baranes and P.-Y. Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1): 49–73, 2013.

B. I. Bertenthal. Origins and early development of perception, action, and representation. *Annual Reviews Psychology*, 47:431–459, 1996.

N. E. Berthier and R. Keen. Development of reaching in infancy. *Experimental Brain Research*, 169(4):507–518, 2005.

M. Biehl and H. Schwarze. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and General*, 28(3):643–656, 1995.

H. Bloch and I. Carchon. On the onset of eye-head coordination in infants. *Behavioral Brain Research*, 49(1):85–90, 1992.

L. Bottou. Online algorithms and stochastic approximations. In D. Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998.

L. Bottou and Y. LeCun. Large scale online learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

D. Bullock, S. Grossberg, and F. H. Guenther. A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Cognitive Neuroscience*, 5(4): 408–435, 1993.

E. W. Bushnell. The decline of visually guided reaching during infancy. *Infant Behavior and Development*, 8(2):139–155, 1985.

M. V. Butz, O. Herbort, and J. Hoffmann. Exploiting redundancy for flexible behavior: unsupervised learning in a modular sensorimotor control architecture. *Psychological Review*, 114(4):1015–1046, 2007.

D. Caligiore, T. Ferrauto, D. Parisi, N. Accornero, M. Capozza, and G. Baldassarre. Using motor babbling and hebb rules for modeling the development of reaching with obstacles and grasping. In *Int. Conf. Cognitive Systems (CogSys)*, 2008.

R. K. Clifton, B. A. Morrongiello, W. Kulig, and J. Dowd. Developmental changes in auditory localization in infancy. In R. Aslin, J. Alberts, and M. Petersen, editors, *Development of Perception, Vol. 1, Psychobiological Perspectives*, pages 141–160. New York: Academic Press, 1981.

R. K. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson. Is visually guided reaching in early infancy a myth? *Child Development*, 64(4):1099–1110, 1993.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1):129–145, 1996.

A. Dearden and Y. Demiris. Learning forward models for robots. In *Int. Joint Conf. Artificial Intelligence (IJCAI)*, pages 1440–1445, 2005.

D. Demers and K. Kreutz-Delgado. Learning global direct inverse kinematics. In *Advances in Neural Information Processing Systems (NIPS)*, 1992.

Y. Demiris and A. Dearden. From motor babbling to hierarchical learning by imitation: A robot developmental pathway. In *Int. Conf. Epigenetic Robotics (EpiRob)*, 2005.

Deutscher Zukunftspreis (German Future-Award), 2010. URL `http://www.deutscher-zukunftspreis.de/en/content/2010`.

A. D'Souza, S. Vijayakumar, and S. Schaal. Learning inverse kinematics. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2001.

P. Ekman and W. Friesen. *Facial Action Coding System (FACS): A technique for the measurement of facial action*. CA: Consulting, 1978.

R. Featherstone and D. E. Orin. Chapter 2: Dynamics. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 35–65. Springer New York, 2007.

J. R. Flanagan and A. M. Wing. The role of internal models in motion planning and control: Evidence from grip force adjustments during movements of hand-held loads. *Journal of Neuroscience*, 17(4):1519–1528, 1997.

B. Fritzke. Incremental learning of local linear mappings. In *Int. Conf. Artificial Neural Networks (ICANN)*, 1995.

M. Gienger, H. Janssen, and C. Goerick. Task-oriented whole body motion for humanoid robots. In *IEEE RAS/RSJ Int. Conf. Humanoid Robots*, 2005.

I. S. Godage, D. T. Branson, E. Guglielmino, G. A. Medrano-Cerda, and D. G. Caldwell. Shape function-based kinematics and dynamics for variable length continuum robotic arms. In *ICRA*, 2011.

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

M. W. Hannan and I. D. Walker. Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *Journal of Robotic Systems*, 20(2), 2003.

C. Hartmann, J. Boedecker, O. Obst, S. Ikemoto, and M. Asada. Real-time inverse dynamics learning for musculoskeletal robots based on echo state gaussian process regression. In *Robotics: Science and Systems (RSS)*, 2012.

M. Haruno, D. M. Wolpert, and M. Kawato. Multiple paired forward-inverse models for human motor learning and control. In *Advances In Neural Information Processing Systems (NIPS)*, 1999.

M. Haruno, D. M. Wolpert, and M. Kawato. MOSAIC model for sensorimotor learning and control. *Neural Computation*, 13(10):2201–2220, 2001.

M. Hersch, E. L. Sauser, and A. Billard. Online learning of the body schema. *International Journal of Humanoid Robotics*, 5(2):161–181, 2008.

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

M. Ito. Control of mental activities by internal models in the cerebellum. *Nature Reviews Neuroscience*, 9:304–313, April 2008.

H. Jäger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report 148, German National Research Center for Information Technology, Sankt Augustin, 2001.

L. Jamone, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi. Learning task space control through goal directed exploration. In *IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, 2011.

B. A. Jones and I. D. Walker. Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1), 2006.

B. A. Jones and I. D. Walker. Limiting-case analysis of continuum trunk kinematics. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2007.

M. I. Jordan. Computational aspects of motor control and motor learning. In *Handbook of Perception and Action: Motor Skills*. Academic Press, 1996.

M. I. Jordan and D. E. Rumelhart. Forward models: supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354, 1992.

M. Kawato. Feedback-error-learning neural network for supervised motor learning. In R. Eckmiller, editor, *Advanced Neural Computers*, pages 365–372. Elsevier, 1990.

M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9(6):718–727, 1999.

M. Kawato and H. Gomi. A computational model of four regions of the cerebellum based on feedback-error learning. *Biological Cybernetics*, 68(2):95–103, 1992.

O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

J. Konczak, M. Borutta, H. Topka, and J. Dichgans. The development of goal-directed reaching in infant: hand trajectory formation and joint torque control. *Experimental Brain Research*, 106(1):156–168, 1995.

J. Konczak, M. Borutta, and J. Dichgans. The development of goal-directed reaching in infants ii. learning to produce task-adequate patterns of joint torque. *Experimental Brain Research*, 113(3):465–474, 1997.

S. Kopp and J. J. Steil. Special corner on cognitive robotics. *Cognitive Processing*, 12 (4):317–318, 2011.

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

M. Kuperstein. Neural model of adaptive hand-eye coordination for single postures. *Science*, 239(4845):1308–1311, 1988.

J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Advances In Neural Information Processing Systems (NIPS)*, 2008.

C. Laschi, B. Mazzolai, V. Mattoli, M. Cianchetti, and P. Dario. Design of a biomimetic robotic octopus arm. *Bioinspiration & Biomimetics*, 4(1), 2009.

Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

A. Liegeois. Automatic supervisory control of configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7(12):861–871, 1977.

M. Lopes and B. Damas. A learning framework for generic sensory-motor maps. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2007.

M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: A survey. *Connection Science*, 15(4):151–190, 2003.

R. Martinez-Cantin, M. Lopes, and L. Montesano. Body schema acquisition through active learning. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

B. W. Mel. A connectionist model may shed light on neural mechanisms for visually guided reaching. *Journal of Coghnitive Neuroscience*, 3(3):273–292, 1991.

A. Miyamura and H. Kimura. Stability of feedback error learning scheme. *Systems & Control Letters*, 45(4):303–316, 2002.

K. Neumann, M. Rolf, and J. J. Steil. Reliable integration of continuous constraints into extreme learning machines. *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, (Special Issue: Int. Symp. Extreme Learning Machines 2012), 2013. To appear.

D. Nguyen-Tuong and J. Peters. Model learning for robot control: a survey. *Cognitive Processing*, 12(4), 2011. Special Corner: Cognitive Robotics.

D. Nguyen-Tuong, M. Seeger, and J. Peters. Computed torque control with nonparametric regression models. In *American Control Conference*, 2008.

F. Nori, L. Natale, G. Sandini, and G. Metta. Autonomous learning of 3d reaching in a humanoid robot. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 1142–1147, 2007.

P.-Y. Oudeyer and F. Kaplan. How can we define intrinsic motivations? In *Int. Conf. Epigenetic Robotics (EpiRob)*, 2008.

L. Out, A. J. van Soest, G. P. Savelsbergh, and B. Hopkins. The effect of posture on early reaching movements. *Journal of Motor Behavior*, 30(3):260–272, 1998.

E. Oyama and T. Tachi. Goal-directed property of online direct inverse modeling. In *Int. Joint Conf. Neural Networks (IJCNN)*, 2000.

D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *IEEE-RAS Int. Conf. Humanoid Robots*, 2008.

U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini. An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 1668–1674, 2010.

J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Int. Conf. Machine Learning (ICML)*, 2007.

J. Peters and S. Schaal. Learning to control in operational space. *The International Journal of Robotics Research*, 27(2):197–212, 2008.

J. Piaget. *The Origin of Intelligence in the Child*. Routledge and Kegan Paul, 1953.

T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

J. Porrill and P. Dean. Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems. *Neural Computation*, 19(1):170–193, 2007.

J. Porrill, P. Dean, and J. V. Stone. Recurrent cerebellar architecture solves the motor-error problem. *Proc Biol Sci*, 271(1541):789–796, 2004.

F. R. Reinhart and J. J. Steil. Recurrent neural associative learning of forward and inverse kinematics for movement generation of the redundant pa-10 robot. In *Int. Symp. of Learning and Adaptive Behavior in Robotic Systems (LAB-RS)*, 2008.

F. R. Reinhart and J. J. Steil. Neural learning and dynamical selection of redundant solutions for inverse kinematic control. In *IEEE-RAS Int. Conf. Humanoid Robots*, 2011.

H. Ritter. Learning with the self-organizing map. In T. Kohonen, editor, *Artificial Neural Networks*. Elsevier Science, 1991.

P. Rochat. Self-sitting and reaching in 5- to 8-month-old infants: The impact of posture and its development on early eye-hand coordination. *Journal of Motor Behavior*, 24 (2):210–220, 1992.

M. Rolf. CoR-Lab Continuum Kinematics Simulation, 2012. URL `http://www.cor-lab.de/software-continuum-kinematics-simulation`.

M. Rolf and J. J. Steil. Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2012a.

M. Rolf and J. J. Steil. Explorative learning of right inverse functions: theoretical implications of redundancy. In *New Challenges in Neural Computation*, 2012b.

M. Rolf and J. J. Steil. Explorative learning of inverse models: a theoretical perspective. *Neurocomputing*, 2013a. Submitted.

M. Rolf and J. J. Steil. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE Trans. Neural Networks and Learning Systems*, 2013b. Submitted.

M. Rolf, J. J. Steil, and M. Gienger. Efficient exploration and learning of whole body kinematics. In *IEEE Int. Conf. Development and Learning (ICDL)*, 2009.

M. Rolf, J. J. Steil, and M. Gienger. Mastering growth while bootstrapping sensorimotor coordination. In *Int. Conf. Epigenetic Robotics (EpiRob)*, 2010a.

M. Rolf, J. J. Steil, and M. Gienger. Learning flexible full body kinematics for humanoid tool use. In *Int. Symp. Learning and Adaptive Behavior in Robotic Systems (LAB-RS)*, 2010b.

M. Rolf, J. J. Steil, and M. Gienger. Goal babbling permits direct learning of inverse kinematics. *IEEE Trans. Autonomous Mental Development*, 2(3), 2010c.

M. Rolf, J. J. Steil, and M. Gienger. Online goal babbling for rapid bootstrapping of inverse models in high dimensions. In *IEEE Int. Joint Conf. Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2011.

L. Rönnqvist and C. von Hofsten. Neonatal finger and arm movements as determined by a social and an object context. *Early Development and Parenting*, 3(2):81–94, 1994.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

U. Sailer, J. R. Flanagan, and R. S. Johansson. Eye–hand coordination during learning of a novel visuomotor task. *Journal of Neuroscience*, 25(39):8833–8842, 2005.

C. Salaün, V. Padois, and O. Sigaud. Learning forward models for the operational space control of redundant robots. In O. Sigaud and J. Peters, editors, *From Motor Learning to Interaction Learning in Robots*, pages 169–192. Springer, 2010.

T. D. Sanger. Failure of motor learning for large initial errors. *Neural Computation*, 16(9):1873–1886, 2004.

S. Schaal and C. G. Atkeson. Assessing the quality of learned local models. In *Advances in Neural Information Processing Systems (NIPS)*, 1994.

W. Schenck. Adaptive internal models for motor control and visual prediction. In *MPI Series in Biological Cybernetics*. Logos Verlag: Berlin, 2008. Doctoral Thesis.

G. Schillaci and V. Hafner. Prerequisites for intuitive interaction - on the example of humanoid motor babbling. In *Proceedings of the Workshop on The role of expectations in intuitive human-robot interaction (at HRI 2011)*, pages 23–27. 2011.

B. Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison, 2010.

P. Sollich and D. Barber. Online learning from finite training sets: An analytical case study. In *Advances in Neural Information Processing Systems (NIPS)*, 1996.

P. O. Stalph and M. V. Butz. Learning local linear jacobians for flexible and adaptive robot arm control. *Genetic Programming and Evolvable Machines*, 13(2):137–157, 2012.

J. Sturm, C. Plagemann, and W. Burgard. Unsupervised body scheme learning through self-perception. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2008.

G. Sun and B. Scassellati. Reaching through learned forward models. In *IEEE-RAS/RSJ International Conference on Humanoid Robots (Humanoids)*, 2004.

G. Sun and B. Scassellati. A fast and efficient model for learning to reach. *International Journal of Humanoid Robotics (IJHR)*, 2(4):391–413, 2005.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

T. Takenaka. The control system for the honda humanoid robot. *Age and Ageing*, 35 (2):24–26, 2006.

E. Thelen and J. P. Spencer. Postural control during reaching in young infants: A dynamic systems approach. *Neuroscience and Biobehavioral Reviews*, 22(4):507–514, 1998.

E. Thelen, Corbetta, Daniela, and J. P. Spencer. Development of reaching during the first year: Role of movement speed. *Journal of Experimental Psychology*, 22(5): 1059–1076, 1996.

E. Theodorou, J. Buchli, and S. Schaal. Reinforcement learning of motor skills in high dimensions: A path integral approach. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010.

A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

S. Ulbrich, V. Ruiz de Angulo, T. Asfour, C. Torras, and R. Dillmann. Kinematic bzier maps. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):1215–1230, 2012.

A. van der Meer. Keeping the arm in the limelight: Advanced visual control of arm movements in neonates. *European Journal of Paediatric Neurology*, 1(4):103–108, 1997.

A. van der Meer, F. van der Weel, and D. Lee. The functional significance of arm movements in neonates. *Science*, 267(5198):693–695, 1995.

VICON Motion Tracking Systems. URL `http://www.vicon.com`.

C. von Hofsten. Eye-hand coordination in the newborn. *Developmental Psychology*, 18(3):450–461, 1982.

C. von Hofsten. An action perspective on motor development. *Trends in Cognitive Science*, 8(6):266–272, 2004.

K. Waldron and J. Schmiedeler. Chapter 1: Kinematics. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 9–33. Springer New York, 2007.

J. Walter, C. Nölker, and H. Ritter. The psom algorithm and applications. In *Int. ICSC Symposium on Neural Computation*, 2000.

J. P. Welsh and R. Llins. Some organizing principles for the control of movement based on olivocerebellar physiology. *Progress in Brain Research*, 114:449–461, 1997.

W. A. Wolovich and H. Elliot. A computational technique for inverse kinematics. In *IEEE Conf. on Decision and Control*, 1984.

D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7–8):1317–1329, 1998.

D. M. Wolpert, R. C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Science*, 2(9):338–347, 1998.

D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan. Perspectives and problems in motor learning. *Trends in Cognitive Science*, 5(11):487–494, 2001.

T. Wu, N. J. Butko, P. Ruvulo, M. S. Bartlett, and J. R. Movellan. Learning to make facial expressions. In *IEEE Int. Conf. Development and Learning (ICDL)*, 2009.

S. Xu, P. Van Dooren, R. Stefan, and J. Lam. Robust stability and stabilization for singular systems with state delay and parameter uncertainty. *IEEE Transactions on Automatic Control*, 47(7):1122–1128, 2002.