

Self-Organizing Control for Autonomous Robots

A Dynamical Systems Approach Based on the
Principle of Homeokinesis

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
"Doctor rerum naturalium"
der Georg-August-Universität Göttingen

vorgelegt von
Frank Hesse
aus Heilbad Heiligenstadt (Eichsfeld)

Göttingen 2008

Prof. Dr. Theo Geisel (Referent)

Max-Planck-Institut für Dynamik und Selbstorganisation und
Abteilung für Nichtlineare Dynamik, Georg-August-Universität Göttingen

Prof. Dr. Ralf Der (Referent)

Arbeitsgruppe Neuroinformatik und Robotik, Universität Leipzig

Dr. J. Michael Herrmann

Institute of Perception, Action and Behaviour, University of Edinburgh

Dr. Lüder Mosler

Strategisches Technologie Management, Otto Bock HealthCare GmbH Duderstadt

Tag der mündlichen Prüfung: 19.01.2009

Abstract

This thesis presents a self-referential dynamical systems approach to adaptive robot control. The central idea consists of the maintenance of behavioral activity in a robot, according to the principle of Homeokinesis, which represents a dynamical counterpart to Homeostasis. An algorithm is described that is able to generate behavioral competencies specific to the robotic device and its environment. The underlying objective function, from which concrete learning rules for the controller parameters are derived, will be defined. This function does not rely on the specific realization of the robotic body nor the environment, but rather leads to environment related behaviors. Moreover the properties of the parameter dynamics are studied.

In experiments with various robots the emergence of sensorimotor coordination under homeokinetic control is shown and the applicability and properties of the approach are discussed. As a paradigmatic toy example we consider a one-dimensional hit-and-return task where the proposed controller out-performs a static controller with the same initial parameters. Crawling and jumping motions of a snake-like robot or a whole-body motion of the active as well as passive elements of an under-actuated robot are shown. Such modes of behavior are possible by exploiting the physical properties of the robotic device. Long-term experiments demonstrate that the parameter dynamics of the control system continues to explore the parameter space in accordance with the theoretical predictions. Also the activity of the system does not decrease for longer periods of time, even in an environment crowded with other active robots. Furthermore, the ability of the homeokinetic system to integrate and disintegrate, respectively, attached and removed sensors is shown and it is demonstrated that, if multiple sensors are available, those which show a more reliable response to the motor actions are predominantly integrated into the control loop.

Extensions to the homeokinetic controller, including a long-term memory and second-order learning, are presented. The long-term memory enables the system to adapt to different situations without relearning the internal parameters. Using second order learning the behavior of the homeokinetic system can be shaped, based on observations of previous behaviors and additional exteroceptive sensory information. This can be done either to advance the adaptation that is implied by the objective function or to fulfill an externally specified task.

The homeokinetic approach is also applied to the control of myoelectric hand prostheses, thereby realizing a new method to adapt the controller of a myoelectric prosthesis to the individual situation of an amputee during an interaction between patient and prosthesis. This interaction period is intended on the one hand to automatically select patient-specific features of the myoelectric signal and on the other hand to improve the signals provided by the amputee through training. The selected features are shown to be effective in the control of the prosthesis.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit einem selbstreferentiellen Ansatz aus dem Bereich der dynamischen Systeme zur adaptiven Steuerung von Robotern. Die zentrale Idee des Ansatzes besteht dabei in der Aufrechterhaltung aktiver Verhaltensformen, was ihn zu einem dynamischen Pendant der Homeostase macht und zu dem Namen Homeokinese führte. Ein Algorithmus wird vorgestellt, mit Hilfe dessen körper- und umweltangepaßte Verhalten generiert werden können. Die zugrundeliegende objektive Funktion ist dabei nicht auf einen speziellen Roboter oder eine vorgegebene Umwelt angewiesen. Weiterhin wird die Parameterdynamik des Systems diskutiert.

In Experimenten mit verschiedenartigen Robotern werden die Anwendbarkeit und die Eigenschaften des Ansatzes untersucht. Als paradigmatisches Beispiel dient ein eindimensionaler Roboter zwischen zwei Wänden. In dieser Situation können mit einer homeokinetischen Steuerung deutlich längere Strecken zurückgelegt werden, als mit einem statischen Kontroller. In weiteren Experimenten werden koordinierte Bewegungen, wie Kriech- und Springbewegungen eines schlangenartigen Roboters oder Bewegungen aller aktiven und passiven Körperteile eines unteraktuierten Roboters, generiert. Solche Bewegungsmoden zeigen die Emergenz von sensomotorischer Koordination durch homeokinetische Steuerung und werden durch die Ausnutzung der physikalischen Eigenschaften der Roboter möglich. In Langzeitexperimenten zeigt sich im Einklang mit der Theorie, daß die Parameterdynamiken des Steuerungssystems den Paramterraum fortlaufend explorieren. Die Roboter sind sogar in überfüllten Umgebungen höchstens kurzzeitig inaktiv. Neu angeschlossene oder entfernte Sensoren werden in die sensomotorische Schleife integriert beziehungsweise desintegriert, wobei Sensoren, die zuverlässiger auf Motoraktionen reagieren, bevorzugt werden.

Erweiterungen der homeokinetischen Steuerung durch ein Langzeitgedächtnis und Lernen zweiter Ordnung werden vorgestellt. Das Langzeitgedächtnis erlaubt dem System sich an verschiedene Situationen anzupassen, ohne die internen Parameter umlernen zu müssen. Durch Lernen zweiter Ordnung in einer höheren Kontrollschicht werden bestimmte Aspekte der generierten Verhalten repräsentiert und unter Verwendung von zusätzlichen exterozeptiven Sensoren das System in geeigneter Weise beeinflusst. So kann die Parameteradaption entsprechend der objektiven Funktion beschleunigt oder eine von außen vorgegebene Aufgabe erfüllt werden.

Schließlich wird eine neue Methode zur Anpassung von Prothesensteuerungen an die individuelle Situation eines Amputierten in einer Interaktionsphase zwischen Patient und Prothese vorgestellt. In der Interaktionsphase sollen zum einen die patientenspezifischen Merkmale des myoelektrischen Signals selektiert und zum anderen die vom Patienten bereitgestellten Signale durch Training verbessert werden. Die gewählten Merkmale können effektiv zur Steuerung der Prothese verwendet werden.

Contents

Abstract	i
Zusammenfassung	iii
List of Symbols	vii
1 Introduction	1
1.1 History of Robots	1
1.2 Future Robots	2
1.3 Defining Autonomous Robots	3
1.4 Developing Controllers for Autonomous Robots	4
1.5 Robots as Dynamical Systems	6
1.6 Introducing the Homeokinetic Approach	6
1.7 Embodiment, Situatedness and Homeokinesis	8
1.8 Simulated Versus Real Robots	9
1.9 Thesis Outline	9
2 The Homeokinetic Principle	11
2.1 Introduction	11
2.1.1 Closed-loop control	11
2.1.2 Sensorimotor Loop	12
2.2 Closed-Loop Control in the Sensorimotor Loop	13
2.2.1 Formal Specification of the Sensorimotor Loop	13
2.2.2 An Elementary Sensorimotor Loop	14
2.2.3 Sensorimotor Loop with Several Sensors	25
2.2.4 Systems with Delay	26
2.3 The Principle of Parameter Regulation	27
2.3.1 The Dynamical Systems Approach to Robot Control	27
2.3.2 The Adaptive Systems Approach	28
2.3.3 Defining the Objective Function	29
2.4 Parameter Dynamics in the Elementary Sensorimotor Loop	33
2.4.1 The Generation of Activity	35
2.4.2 Requiring Predictability	35
2.4.3 The Frequency Effect	37
2.5 Parameter Dynamics Using Several Sensors	38

2.6	Discussion	40
3	Applying the Homeokinetic Principle to Autonomous Robots	43
3.1	The <i>lpzrobots</i> Software Package	43
3.2	Homeokinetic Versus Static Controller	45
3.3	Environment Exploration with a Simulated Two Wheeled Robot	50
3.4	Simulated Snakelike Robot	59
3.5	Emerging Sensorimotor Coordination	64
3.6	Discussion	78
4	Extending Homeokinetic Control	81
4.1	Long-Term Memory	81
4.2	Second-Order Learning	87
4.2.1	Obstacle Avoidance for a Wheeled Robot	89
4.2.2	Gripping in a Human-Hand Model	94
4.3	Discussion	96
5	Self-Organization for the Control of Myoelectric Prostheses	99
5.1	Introduction	99
5.2	Self-Organization in Prostheses Control	101
5.3	Equipment	102
5.4	Illustrative Example	104
5.5	Controlling a Prosthesis	108
5.6	Summary	111
5.7	Discussion	112
6	Conclusion and Outlook	115
A	Derivation of the Objective Function for Several Sensors	119
	Bibliography	121

List of Symbols

Symbol	Description
a	Parameters of the internal model.
c	Synaptic strengths of the controller neuron.
g	Output function of the controller neuron with $g(z) = \tanh(z)$.
h	Bias (adaptive threshold) of the controller neuron.
t	Temporal variable.
v	Input shift is the modeling error when using an internal retrospective model. It is the difference between the measured and the reconstructed sensor value at timestep t .
x	Sensor values being input to the controller.
x^M	Predicted sensor value (based on forward model).
x^R	Reconstructed sensor value (based on retrospective model).
y	Output of the controller neuron, being the motor command for the actuators of the robot.
z	Membrane potential of the controller neuron.
E	Objective function.
F	Internal model.
I	Input space (sensor values).
K	Controller.
L	Jacobian of the loop function ψ .
O	Output space (actions of the robot).
R	Feedback strength in the sensorimotor loop.
ε	Update rate.
ζ	Prediction of the modeling error.
ξ	Modeling error when using an internal predictive model. The modeling error is the difference between the measured and the predicted sensor value at timestep $t + 1$.
τ	Time constant.
ψ	Loop function including controller and world model.
s_t	Vector s at time t .
s_i	Element i of vector s .
$s_{t,i}$	Element i of vector s at time step t .

Chapter 1

Introduction

1.1 History of Robots

Long before robots were created, science fiction authors thought and wrote about them. The term robot was introduced in a play by the Czech writer Karel Čapek:

The word ROBOT was first used in Karel Capek's play RUR (Rossum's Universal Robots). The word robot was invented by Čapek's brother Josef. It is derived from the Czech noun *robota* - labour, so a robot is somebody or something that exercises labour. ([Zunt, 2007](#))

In this play, premiering in Prague in 1921 and first performed in New York in 1922, automatic human-like machines are built to work in production halls.

But even before the term *robot* existed people tried to build machines with human like appearance or abilities called *automata*. According to recent findings, the history of the *automata* goes back to the ancient-greece in the 1st century AD ([Sharkey, 2007](#)) or at least to Leonardo da Vinci in 1497 ([Rosheim, 1996](#)).

Nowadays the majority of robots can be found in manufacturing halls as described by Sebastian Thrun:

At present, most robots operate in industrial settings where they perform tasks such as assembly and transportation. Equipped with minimal sensing and computing, robots are slaved to perform the same repetitive task over and over again. ([Thrun, 2004](#))

These machines are usually not man-like, they have a rather practical appearance (see figure [1.1](#)) and their control is just complex enough to accomplish the given task, but not to cope with environmental changes e.g. in the setup of the workplace. Therefore, many of the industry robots are surrounded by a cage to keep the predetermined environment constant in the vicinity of the robot and to protect the human workers.



Figure 1.1: Industrial robots. (a) A typical robot for welding or handling which can be found in many production halls ([Wikimedia Commons, 2008](#)) and (b) autonomous robots for transporting containers on a loading deck (picture from [Bellin \(2008\)](#)).

1.2 Future Robots

For the future development it is beneficial to increase the complexity of the control structures in order to allow the robots to adapt to different environments, situations, or body properties. This way robots will become able to assist people in everyday life. For this task it is also advantageous to have a more man-like appearance which will allow the robots to share our places and manipulate the same things we do. In the future most robots will then probably look like the prototypes in figure 1.2.

The increased complexity of the body on the one hand and the unconstrained number of possible environments make the development of appropriate controllers difficult. (So for example changing from flat linoleum to a fluffy carpet will result in a decrease of the speed of a two wheeled robot. Hence, it will need more time to reach a target and the controller has to somehow deal with this.) Furthermore, changes in the body properties of the robot can lead to different outcomes when applying the same motor commands. All these uncertainties can only be faced when the controller is able to make decisions autonomously, without human intervention. Thus, the ability of a robot to autonomously adapt to new situations and/or body properties is an important aim in the development of future robots (see for example [Lenser \(2005\)](#); [Salomon \(1996\)](#); [Torras \(1995\)](#)). Or as stated by Carme Torras:

At the risk of oversimplification, let me state that the Achilles heel of current robots is their lack of adaptivity, at all levels. ([Torras, 2002](#))

In this work we will focus on the ability of control structures to adapt to the body and the environment. To achieve this ability there are various ways, out of which the three most prominent ones will be shortly discussed in section 1.4, after defining the kind of robots we want to use in the investigations in this thesis.

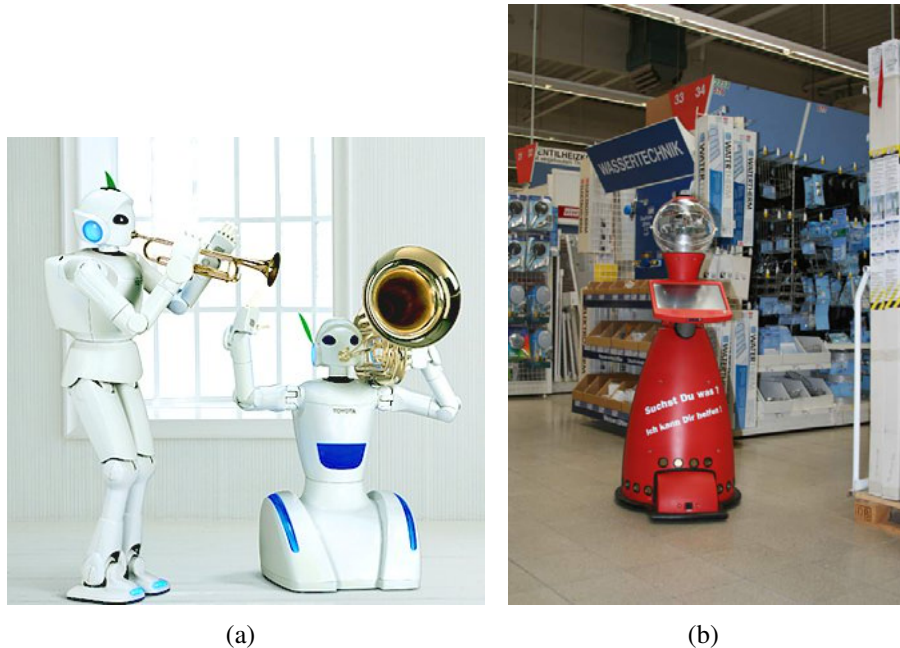


Figure 1.2: Prototypes of possible future robots designed to assist in everyday life. (a) Humanoid and wheel driven “partner robots” developed by the [Toyota Motor Corporation \(2007\)](#). (b) Shoppingrobot developed at the Ilmenau University of Technology ([Neuroinformatics and Cognitive Robotics Lab, 2007](#); [Böhme et al., 2006](#)).

1.3 Defining Autonomous Robots

Before considering the development of controllers, we want to take a more precise look at the term robot. Throughout this work we will consider autonomous robots. Two fundamental requirements for autonomous agents in general, where an agent is an entity who is capable of action, are:

- (1) Autonomous agents structure their behavior on the basis of sensory information that they themselves acquire.
- (2) Such autonomy goes beyond the sensor-driven nature of control systems in that it is flexible. Minimally this means that an agent may change its behavior qualitatively under the influence of sensory information. ([Schöner et al., 1995](#))

This means we will consider robots which are able to change their behavior on their own, without human intervention. Hence, such systems need controllers which are able to make decisions on their own:

The ultimate purpose of any autonomous robot design is to develop systems capable of self-governing in complex and unpredictable environments. ([Goschin et al., 2007](#))

We will discuss three prominent approaches to autonomous robot control in the following section.

1.4 Developing Controllers for Autonomous Robots

1.4.1 Knowledge Based Artificial Intelligence

According to the knowledge based approach to artificial intelligence designers try to list all situations the robot can experience and define rules for adequate reactions of the robot. In general this is done with expert knowledge in order to reproduce the performance of one or more human experts in a specific problem domain. This expert knowledge is then used to make inferences and arrive at specific conclusions for a given question (see [Liao \(2005\)](#) for an overview of expert systems). To do so in the robotic domain the designers have to think of all possible situations during the robot's lifetime already in the design process of the controller. And this must be reconsidered for each change in the body properties of the robot, since the same control command can result in different outcomes, depending on the body properties of the robot. With increasing complexity of the robotic device and the environment, the probability to cover all possible situations is drastically reduced. Furthermore, the definition of a situation is difficult, it can not be described by the use of abstract symbols as we do in our language, since these can not be interpreted by the robot in the context of the real world, as tellingly sketched in figure 1.3 (left). This is the so-called symbol grounding problem as pointed out by [Harnad \(1990\)](#).

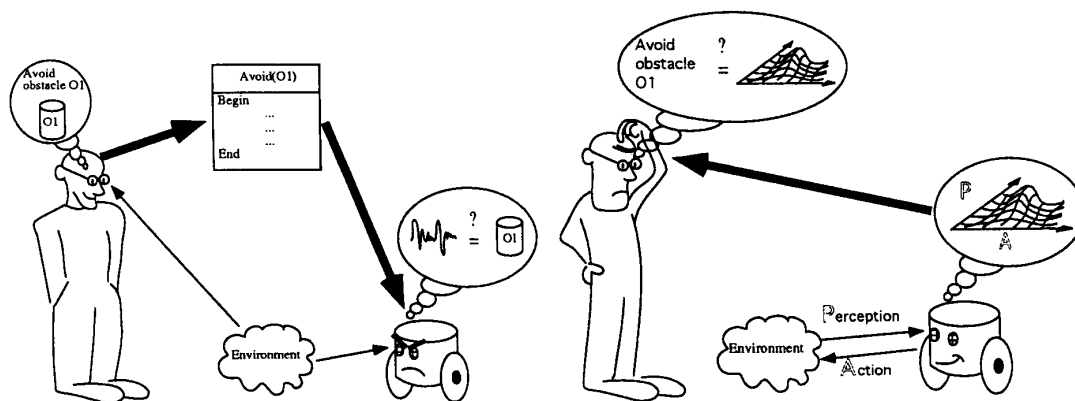


Figure 1.3: The difference between design (left) and self-organization (right) ([Bessiere et al., 1994](#)). While in design it is not clear if the robot interprets the abstract symbols used by the designer correct (symbol grounding problem), in self-organization processes it is not easy to analyze the solution found by the robot.

1.4.2 Reinforcement Learning

Based on reinforcement learning, behaviors of robotic systems can be adapted using (numerical) rewards.

An reinforcement learning agent learns from the consequences of its actions, rather than from being explicitly taught and it selects its actions on basis of its

past experiences (exploitation) and also by new choices (exploration), which is essentially trial and error learning. (Woergoetter and Porr, 2008)

Desired behaviors are favored by positive reinforcement signals, or rewards, whereas unwanted behaviors yield a negative reward/punishment. The controller seeks to learn to select actions that maximize the accumulated reward over time. (For an overview of the principle see Sutton (1992), Kaelbling et al. (1996), Sutton and Barto (1998).) Using reinforcement learning interesting projects could be realized in recent years like a stand-up behavior of a three-link, two-joint robot (Morimoto and Doyab, 2001), the control of an autonomous helicopter (Bagnell and Schneider, 2001) and the dynamic balance of a biped robot using fuzzy reinforcement learning agents (Zhou and Meng, 2003). The downside of this approach is the distribution of the rewards. There is still a designer required to set up the rewards for the specific task and domain in which the robot should learn.

1.4.3 Evolutionary Robotics

Evolutionary robotics (Nolfi and Floreano, 2000; Cliff et al., 1993) is the attempt to develop robots through a process inspired by the natural evolution. An initial population of robot controllers is randomly generated and each of them is allowed to let the robot interact with the environment. The performance of each individual on specific tasks is evaluated. According to the performance, the controllers are allowed to reproduce by copying their genotypes with additional changes due to genetic operations, like mutation or crossover. So a next generation of controllers is generated, can interact with the environment, and will be evaluated afterwards. This process is repeated until an individual is able to fulfill the performance criterion. Using this technique interesting achievements such as a flying, or to be more precise flapping, ornithopter-like robot (Augustsson et al., 2002) and a homing behavior for a Khepera robot (Floreano and Mondada, 1996) could be realized. Komosinski (2000) realized autonomy of design and construction using evolution in a 'limited universe' physical simulation, where controller as well as the robotic body, consisting of bars and actuators, are evolved in order to generate locomotion. Similar work was presented by Lipson and Pollack (2000), where the fittest individuals (defined by their locomotive ability) were physically built using rapid manufacturing technology and reproduced the behaviors of their virtual origin. Maze exploration behaviors in a real and a simulated colony of robots were studied by Nelson et al. (2004). They used rather complex controller networks (multiple hidden layers, time-delayed and recurrent connections) which were evolved to respond to temporal information in order to overcome the inherent perceptual aliasing associated with the simple binary tactile sensors used. However there is also a downside of this approach:

From an engineering perspective, the most serious problem is the amount of time needed to conduct an evolutionary process on physical robots. From a scientific perspective, the most important challenge is the identification of methods of encoding information into the genotype that are suitable to produce incremental and open-ended evolutionary processes (Nolfi and Floreano, 2002).

A further difficulty is the design of the fitness function required for evaluating the performance of each individual. This still has to be done by the experimenter and is significant for the success of the evolutionary process.

1.4.4 Summary

All mentioned approaches have in common that information about the system and the environment is required to define the reactions of the robot, the rewards or the fitness function and genotype. Furthermore, all approaches end with a stable, non-adaptive system. Either the rules are set by the designer or the controller parameters are adapted by reinforcement learning or artificial evolution. However, as soon as the robot is in service no further adaptation takes place.

In this work we try to point out a different approach to robot control which will be introduced in section 1.6. Beforehand the dynamical systems approach to robot control, which is a basis for homeokinetic control, will be shortly presented.

1.5 Robots as Dynamical Systems

According to [Meiss \(2007\)](#) a dynamical system (see also e.g. [Ott \(1993\)](#), [Strogatz \(2001\)](#)) is a rule for the time evolution on a state space:

A dynamical system consists of an abstract phase space or state space, whose coordinates describe the state at any instant; and a dynamical rule that specifies the immediate future of all state variables, given only the present values of those same state variables. ([Meiss, 2007](#))

Examples include mathematical models that describe the swinging of a clock pendulum, the flow of water in a pipe or the spreading of epidemics. In this thesis we will describe the robot in interaction with the environment as a dynamical system. The parameters of the latter will then define the behavior of the robot.

According to the homeokinetic paradigm we will set up a self-regulation of the parameters, without a task, goal, or reference value given from outside, and this way obtain a self-organizing controller for autonomous robots. Activity in this system will be generated by noise amplification and symmetry breaking. Even though we will consider rather simple control architectures complex behaviors will arise, based on the interplay between state and parameter dynamics of the system.

1.6 Introducing the Homeokinetic Approach

The concept of homeokinesis was developed by Ralf Der. He first proposed a system for the adaptation of controllers of autonomous robots in a self-organized way, where the robot still needs a drive for activity given from outside ([Der et al., 1999](#)), ([Der, 2001](#)). In 2002 homeokinesis was proposed in the present form, where the drive for activity also emerges from the general principle ([Der and Liebscher, 2002](#)).

The basic idea of homeokinesis is to provide a way for the adaptation of a controller of a robotic system, which is based entirely on information which is available to the agent itself and does not require a specific environment or robotic body. Nevertheless the controller is expected to generate a rich reservoir of behaviors specific to the robotic device and the environment it is located in. Since the objective function for the adaptation of the controller

parameters is based on internal available information, behavioral competencies will be acquired by self-exploration and interaction with the environment without an external specified goal or task. Hence, as sketched in figure 1.3 (right), the behaviors of the robot are not known beforehand, but they will be explorative, incorporate the properties of the robotic body and depend on the environment the robot is exposed to.

Since the homeokinetic approach, which will be presented here, is based on artificial neural networks (for more information on artificial neural networks see [Rojas \(1996\)](#), [Zell \(1997\)](#) or [Haykin \(1998\)](#)) and fast synaptic plasticity ([Citri and Malenka, 2008](#)), ([Zucker and Regehr, 2002](#)) the robotic systems are models for the behavioral neuroscience, which seeks to understand the neural basis of natural animal behavior:

Such computational models have two purposes: they form the basis for biologically inspired approaches to the design of robot control, while at the same time they yield some insight into the behavior of the brain mechanisms they present. ([Bekey and Goldberg, 1993](#))

In this case we will see that a small fully connected network with fast synaptic plasticity is able to generate rather complex behaviors of robotic devices.

However, these systems, based on the dynamical systems approach to robot control, are also interesting on their own from a dynamical systems point of view due to their rich properties and complex behaviors. These behaviors are generated by self-organization and symmetry breaking due to noise amplification in a rather simple control architecture. The complexity arises from the interplay between the state and parameter dynamics of the systems.

We view homeokinesis as the dynamical pendant of homeostasis ([Cannon, 1939](#)). Like the latter it derives behavior from an entirely internal perspective. So the behavior can be seen as a kind of by-product of satisfying the internal needs of the agent. We give this general idea a constructive formulation in the following way. We suppose that the agent is equipped with an adaptive model of its behavior. A learning signal for both the model and the controller is derived from the misfit between the real behavior of the agent in the world and that predicted by the model (for the significant details see section 2.3.3). In this way, a learning signal for the adaptation of the behavior is derived from a purely internal perspective. Given the constructive formulation above, we will need to define an objective function for the adaptation of the controller and world model parameters corresponding to this idea.

However, since controller and world model start from scratch, that is with small random parameters, the approach faces the so-called cognitive bootstrapping problem, also known as the learning paradox formulated by [Fodor \(1980\)](#) and discussed for example by [Bereiter \(1985\)](#), [Steffe \(1991\)](#), [von Glasersfeld \(1998\)](#). The problem arises from the concomitant learning of controller and world model. On the one hand the controller needs a feasible model to exploit the properties of the device under control. On the other hand the model has to be provided with necessary information about the behavior of the body which requires controller actions making this information available. The proposed principle will give a solution to this problem.

Since the source of the rich behaviors shown by the robots is the parameter adaptation by means of fast synaptic plasticity, the system will (and has to) adapt to changes of the environment or body for the whole lifetime. However, as soon as a specific task or goal

should be achieved or a behavior kept, the system has to be shaped. This brings us to the stability-plasticity-dilemma ([Grossberg, 1987](#)), faced by all learning systems: A balance between stability and plasticity is required to solve the trade-off between the stability required to retain information and the plasticity required for adaptation. However, in this work we are going to approach the dilemma from the adaptive side.

The self-organization process is thought to be a first step in the development of a robot (maybe comparable to newborn vertebrates). It yields to set up the connection between body and controller, and the generation of movements which are appropriate to the specific body without detailed information about the body required. Later on, in an extension of the algorithm or by introducing higher layers in the control structure (like for example in the subsumption architecture proposed by [Brooks \(1986b\)](#)) the emerged behaviors can be used for the execution of specific tasks. The benefit of this way, via the self-organized acquisition of simple behaviors, is to reduce the design effort required to program robots, especially when robots with many degrees of freedom are used in a dynamic environment.

Depending on the realization of this higher order control system this meets the idea of developmental or epigenetic robotics, which tries to understand and model the role of development in the emergence of increasingly complex cognitive structures in interaction with the environment. (For an overview of this field of research see [Asada et al. \(2001\)](#), [Berthouze and Metta \(2005\)](#), [Zlatev and Balkenius \(2001\)](#).)

1.7 Embodiment, Situatedness and Homeokinesis

Since the mid-1980's the term "embodiment" is paid growing attention in the artificial intelligence and cognitive science communities. [Brooks \(1986a\)](#) proposed that system-environment interactions of embodied agents leads to the emergence of intelligence. In this context an embodied agent is a physical device in the real world equipped with sensors and actuators as to interact with his environment. This was opposed to the basic approach of the artificial intelligence that aimed at understanding intelligence as information processing, meaning that computer programs can have cognitive abilities, if the designers are able to write appropriate programs. A popular example that is often cited in this context is Searle's *Chinese Room* (see [Searle \(1980\)](#), e.g. in [Rapaport \(1986\)](#)), showing that understanding means not only to process information but instead to have causal connections between the external world and the internal symbols representing it. This failure of the classical artificial intelligence is discussed by [Brooks \(1991\)](#). In Summary, the idea of embodiment is to use the possibilities of interaction with the environment to allow and simplify the creation of intelligence. An example of the usage of body environment interaction is McGeer's passive walker ([McGeer, 1990](#)), where a purely mechanical system without any actors or sensors walks down a slope in a natural looking way only driven by gravity. When comparing this gait to actual humanoid robots where a lot of engineering and computational effort is used to achieve a human-like walking, the benefit of embodiment is indisputable. Hence, different groups equip passive walkers with sensors and actuators to develop robust walking on flat terrain, up small slopes and in the future maybe even over rough terrain, for example see [Tedrake et al. \(2004\)](#), [Asano et al. \(2000\)](#), [Asano et al. \(2001\)](#). [Pfeifer and Scheier \(1999\)](#) proposed situatedness as a condition for intelligent agents. This means that intelligent agents have to adapt themselves to

their specific actual environment, finding their ecological niche. The opposite will be industrial robots at assembly lines, which are usually put in specific environments with nearly no perturbations.

In this context homeokinesis is a controller paradigm for embodied autonomous robots, since it is based on the interaction with the environment via sensors and actuators. Furthermore, robots following the homeokinetic principle are situated in the sense that they adapt their behavior according to the current environmental situation.

1.8 Simulated Versus Real Robots

In this thesis many experiments were conducted in a physically realistic simulation environment (see section 3.1) because of the higher flexibility offered and the lower manpower required when compared to developing and testing robots in the real world:

Although it is confirmed that control systems developed in simulation will not yield the same good results in reality, simulation allows for much faster testing of new ideas and it can at least provide a starting point for developing real controllers. (Goschin et al., 2007)

The latter point is seen as less important in the work at hand since we are considering systems which are adaptive to environment and body, and hence expect these systems to also be able to adapt to the real world. This expectation was confirmed by experiments with real robots in our group (see section 4.1, figure 3.34 or Der et al. (2006)).

1.9 Thesis Outline

In this chapter we gave an overview of robotic systems nowadays and in the future. The aim of this work is to present a new self-referential dynamical systems approach to adaptive robot control, able to generate behavioral competencies specific to the used robotic device and the environment the robot is located in.

The sensorimotor loop in which these systems work, together with a formal specification, will be introduced in the following chapter. Furthermore, the objective function will be defined and concrete learning rules for the parameters of the controller derived. Additionally the properties of the parameter dynamics will be discussed.

After the theoretical considerations chapter 3 deals with the application of the Homeokinetic principle to different robotic artefacts. In section 3.2 a homeokinetic and a static controller are compared. The environment exploration of a two-wheeled robot is studied in section 3.3. The behaviors generated by a five degree of freedom snake-like robot are investigated in section 3.4. An example for the emergence of sensorimotor coordination is given in section 3.5. Here an under-actuated many degree of freedom robotic device is shown to generate a whole-body motion of all, active as well as passive parts of the body. In section 3.6 the presented results are discussed.

Chapter 4 proposes attempts to extend the so far studied homeokinetic system. A long-term memory is introduced in section 4.1. Together with additional sensory information

it allows the system to adapt to different situations without relearning the parameters. In section 4.2 homeokinesis is extended by second order learning. The behaviors generated by the homeokinetic system are observed from a higher control layer. Together with further (exteroceptive) sensory information the behavior of the system can be shaped to better meet the requirements of the objective function or fulfill an externally specified task.

The homeokinetic approach will be applied to the control of myoelectric hand prosthesis in chapter 5. Here it is used as a new method to adapt the controller of a myoelectric prosthesis to the individual needs of an amputee during the interaction between patient and prosthesis. This training phase is intended on the one hand to select the most appropriate features of the myoelectric signal for that specific amputee and on the other hand to improve the signals provided by the amputee through training. Later on the selected features can be used for the control of the prosthesis.

In chapter 6 the results of this work are summarized and final considerations and remarks on future work are given.

Chapter 2

The Homeokinetic Principle

2.1 Introduction

The homeokinetic principle is a general principle which drives robots to develop body- and environment-related behaviors. It is based on measures which are available to the agent itself. There is no external instance in this approach specifying which kinds of behaviors are favorable. Hence there is no desired behavior, reference value or goal. The only requirements to be fulfilled by the system are that it should be active and able to predict the results of its actions in terms of its sensor values.

In this chapter the homeokinetic approach will be explained in detail, starting with the introduction of the basic terms closed-loop control (section 2.1.1) and sensorimotor loop (section 2.1.2). Based on this, the concrete realization of the system used here can be specified (section 2.2). Furthermore the properties inherent in such systems are investigated to convey an understanding of the effects of the different system parameters. Now we are faced with the very interesting question of how to adapt these parameters in a self-referential fashion in order to achieve active, body- and environment-related behaviors (section 2.3). The resulting properties of the proposed system, equipped with the parameter dynamics, are studied in section 2.4 for the elementary case. These systems are already very interesting as we will see. However we have also investigated the case where several sensors are available in the loop, which leads to the results presented in section 2.5. Section 2.6 will conclude this chapter with a discussion of the described principle.

2.1.1 Closed-loop control

Regarding the question of using feedback, control theory gives at least two possibilities to control a system, namely open loop and closed-loop control (for detailed descriptions see for example Leigh (2004), Warwick (1996) or Günther (1997)). Open loop control is characterized by the lack of a direct connection between the output and controller of a system (see figure 2.1(a)). The reference (or desired) value r is fed into the controller C , which generates an input signal u for the device under control P . The output y of the system is not further observed. Large differences between the desired output and the real output can occur because of the lack of sensitivity to the states and possible perturbances of the system under control. Closed-loop control is realized by introducing feedback, whereby the output

y is measured by a sensor and subtracted from the reference value r (see figure 2.1(b)). The error $e = r - y$ is fed in the controller which tries to minimize it by giving appropriate inputs to the device under control. The used feedback gives the opportunity to react to actual and possibly unpredictable situations of the system. Hence for any kind of a real system with environmental influences feedback is required to achieve a useful control of the system in unforeseen situations.

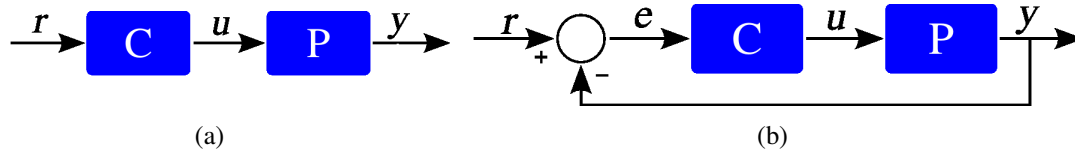


Figure 2.1: Open versus closed loop. (a) Simple open-loop control. No feedback from the system output y to the controller C is used in order to realize the reference value r . (b) Simple feedback loop. The output of the system y is fed back to the reference value r , through a measurement performed by a sensor. The controller C then uses the difference between the reference and the output, the error e , to adjust the inputs u to the system under control P in order to realize the reference value r . The environmental influence, or disturbance acting on the system under control is not explicitly shown here.

2.1.2 Sensorimotor Loop

The term sensorimotor loop highlights the tight relationship between sensory input and motor output in a feedback loop. Already in 1948 Norbert Wiener pointed out the importance of feedback for biological agents (Wiener, 1961):

We thus see that for effective action on the outer world it is not only essential that we possess good effectors, but that the performance of these effectors be properly monitored back to the central nervous system, ... to produce a properly proportioned output to the effectors.

Figure 2.2 shows a more biologically inspired view on feedback systems as in control theory (figure 2.1(b)). The sensorimotor loop describes the information flow in an agent-world-interaction scenario. The agent can be a living being or an autonomous robot which is controlled in a closed loop paradigm. Information from the surrounding world and/or the body is gathered by sensors, and processed in a control unit. The resulting motor command is executed by the agents actuators, which leads to a new situation in the world and thus new sensor values. These new sensor values are again processed by the controller and result in new motor commands to be executed. This loop is repeated until the end of the agent's lifetime.

In recent research in the field of neuroscience, feedback in the sensorimotor loop is studied, for example in the mice or rat vibrissa system (Ferezou et al., 2007; Mitchinson et al., 2007; Nguyen and Kleinfeld, 2005). This was motivation to further study realizations of sensorimotor loops in the robotics. Ahissar and Kleinfeld (2003) give some examples of motor control by closed loops and the following concluding remarks:

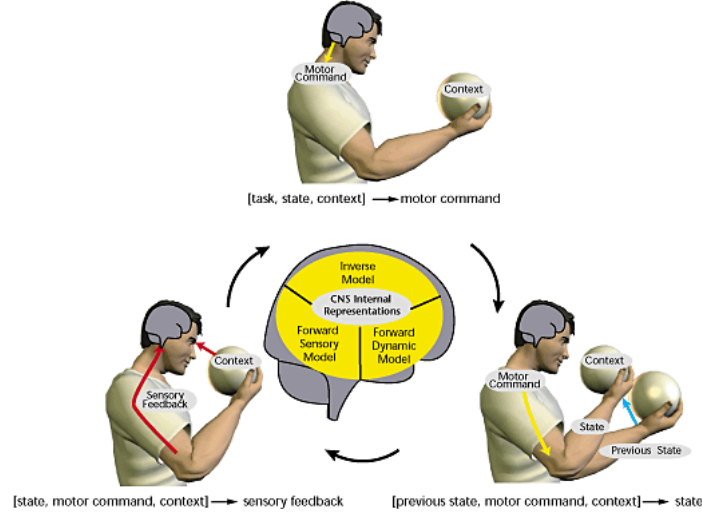


Figure 2.2: The sensorimotor loop, showing motor command generation (top), state transition (right) and sensory feedback generation (left). Center, internal representation of these stages within the central nervous system. (Wolpert and Ghahramani, 2000).

It is possible that closed-loop computation and plasticity are two of the most critical features which make brains so efficient. Achieving an understanding of the interplay between closed-loop computations and plasticity is a further challenge.

This is exactly the point of interest in this thesis where we will investigate closed loop control systems which show a high level of plasticity.

2.2 Closed-Loop Control in the Sensorimotor Loop

The homeokinetic approach realizes closed-loop control in a sensorimotor loop without an external given reference value. The adaptation of the controller parameters is based on measures which are completely internal to the robot. As a result we have a truly autonomous robot, with self-organized generation of behavior. The following section will give a formal specification of the concrete realization of the sensorimotor loop. The properties of this realization are discussed in section 2.2.2 using an elementary sensorimotor loop as an example. In section 2.2.3 the dynamics of the sensorimotor loop in the closed form is derived for the case of multiple sensors. Section 2.2.4 discusses delays between the generation of the motor commands and the corresponding response in the sensor values. The composed controller aggregating the controller and an internal model is defined in section 2.3.3.1.

2.2.1 Formal Specification of the Sensorimotor Loop

In the formal sense a robot is an input-output system which at instances of time $t = 0, 1, 2, \dots$ receives inputs $x_t \in I$ and produces outputs $y_t \in O$. The input space I is the space of input

values (sensor values) and the output space O corresponds to the actions of the robot. These actions are motor commands like the target wheel velocity of the robot or the forces exerted on the joints of some robotic limb.

Closed-loop control means that the controller of the robot is given by a function $K : I \rightarrow O$ mapping the sensor values x to the motor values y

$$y_t = K(x_t). \quad (2.1)$$

This mapping will be realized by an artificial neural network with leaky integrator neurons under the rate coding paradigm.

The change of the sensor values depends on the actions undertaken by the robot. Hence we can introduce a simple function $F : O \rightarrow I$ mapping the actions of the robot to the new sensor values. For generality this mapping could be complemented with sensory and/or further context information in order to increase its predictive abilities. However we are free to keep it at this stage and be aware of the fact that it is restrained here. Using the mapping, the sensor values can be written as

$$x_{t+1} = F(y_t) + \xi_t, \quad (2.2)$$

where ξ_t accounts for the effects not covered by F . So the mapping $\hat{F} : O \rightarrow I$ from motor commands to new sensor values in the world can be written as

$$\hat{F}(y_t) = F(y_t) + \xi_t.$$

With these notions we can write the dynamics of the sensorimotor loop in the closed form:

$$x_{t+1} = \psi(x_t) + \xi_t, \quad (2.3)$$

where

$$\psi(x_t) = F(K(x_t)). \quad (2.4)$$

The function ψ can be visualized as a time series predictor for the time series of the sensor value x . Figure 2.3 shows a sketch of the described sensorimotor loop.

In order to discuss the peculiarities of this realization of a sensorimotor loop we will now study a simple system with velocity control of a wheeled robot.

2.2.2 An Elementary Sensorimotor Loop

Let us consider a closed-loop velocity control of a robot with the sensorimotor loop closed via the wheels alone. For the sake of simplicity we consider the one-dimensional case with $x, y \in \mathbb{R}^1$, i.e. the robot can move only along a straight line. So the controller K consists of a single leaky-integrator neuron under the rate coding paradigm ((Hopfield, 1984); see figure 2.4 for a sketch of the example of the sensorimotor loop).

2.2.2.1 Dynamics of the Loop

In the elementary sensorimotor loop the controller from equation 2.1 is

$$K(x_t) = g(z_t), \quad (2.5)$$

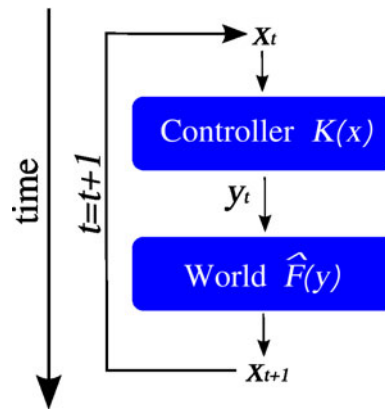


Figure 2.3: Sketch of a sensorimotor loop showing the repeated mapping K from sensor values x_t to motor commands y_t realized by the controller and the mapping \hat{F} from motor commands y_t to new sensor values x_{t+1} happening in the world. (The model F used for the internal prediction of new sensor values is not drawn here.)

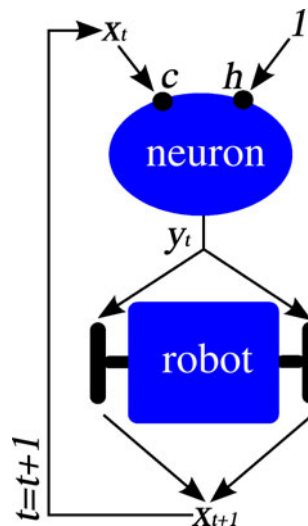


Figure 2.4: Sketch of a simple example of a sensory motor loop. The sensor value x_t (measured wheel velocity) is fed into the neuron weighted with the synaptic strength c . A threshold h is also provided. The neuron output y_t is the desired wheel velocity for both of the wheels. The true (realized) wheel velocity is read back by the wheel counter and used as new sensor value x_{t+1} in the next time step.

with z being the membrane potential of the controller neuron. The output function of the neuron is $g(z) = \tanh(z)$, because in any real system actuators have an upper and a lower limit, which is reflected in the controller by the hyperbolic tangent, setting these limits to ± 1 . In cases with different actuator limits, an adequate factor can be introduced. The membrane potential z of the controller neuron is updated in the time step $t \dots t + 1$ as

$$\tau \Delta z_t = -z_t + cx_{t+1} + h, \quad (2.6)$$

with the controller parameters synaptic strength $c \in \mathbb{R}^1$ and bias (or adaptive threshold) $h \in \mathbb{R}^1$. The bias is always considered explicitly because its role is more like an additional internal state of the neuron. The input to the neuron is the true wheel velocity x as measured by the wheel counter. The update is carried out when the new sensor value x_{t+1} arrives. The output of the neuron

$$y_t = \tanh(z_t) \quad (2.7)$$

is the target wheel velocity of the robot. Figure 2.5 shows a sketch of the neuron model used.

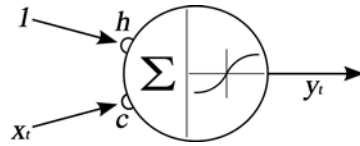


Figure 2.5: Sketch of neuron model used, with input x , synaptic strength c , bias h and output y . The leaky-integrator neuron under rate coding paradigm has a dot product activation function and the output function is the hyperbolic tangent.

The mapping F in equation 2.2 is just linear

$$F(y_t) = ay_t.$$

So the true wheel velocity x_{t+1} as read back by the wheel counter may be assumed to

$$x_{t+1} = ay_t + \xi_t, \quad (2.8)$$

where the parameter $a \in \mathbb{R}^1$ accounts for a (unknown) hardware constant (the response strength of the channel) and $\xi \in \mathbb{R}^1$ incorporates all effects due to slip, friction, discretization noise and so on which make the true velocity deviate from the model assumption $x_{t+1} \stackrel{!}{=} ay_t$. The constant a can be learned by minimizing the error

$$E = (x_{t+1} - ay_t)^2 \quad (2.9)$$

for samples (x_{t+1}, y_t) obtained on-line in each time step $t = 0, 1, \dots$

Using equation 2.6, 2.7 and 2.8 we can write the closed dynamical system as

$$\tau \Delta z_t = -z_t + ca \tanh(z_t) + h + c\xi_t, \quad (2.10)$$

describing the time evolution of the membrane potential of the controller neuron. Together with equation 2.7 this dynamics completely defines the behavior of the robot and will be used in the following section to discuss the stationary states of the system.

2.2.2.2 Stationary States of the System as Solution of the Fixed Point Equation

To get an idea about the properties of the dynamical system (equation 2.10), we want to investigate its stationary states in the ideal case ($\xi = 0$) by way of the graphical solution of the fixed point equation. In a first step we neglect the bias ($h = 0$), while in a second step we discuss the influence of h in detail.

Stationary States of the System Without Bias

For stationary states of the system the update of the membrane potential z is zero, meaning equation 2.6 changes to

$$z_t = cx_{t+1} \quad (2.11)$$

and with equation 2.7 and 2.8 we get the fixed point condition of the system

$$z_t = ca \tanh(z_t), \quad (2.12)$$

or with the feedback strength in the loop $R = ca$

$$z_t = R \tanh(z_t). \quad (2.13)$$

The graphical solution of the fixed point equation can easily be obtained and is shown in figure 2.6. For values of R smaller than one there is one stable fixed point at $z^* = 0$. With

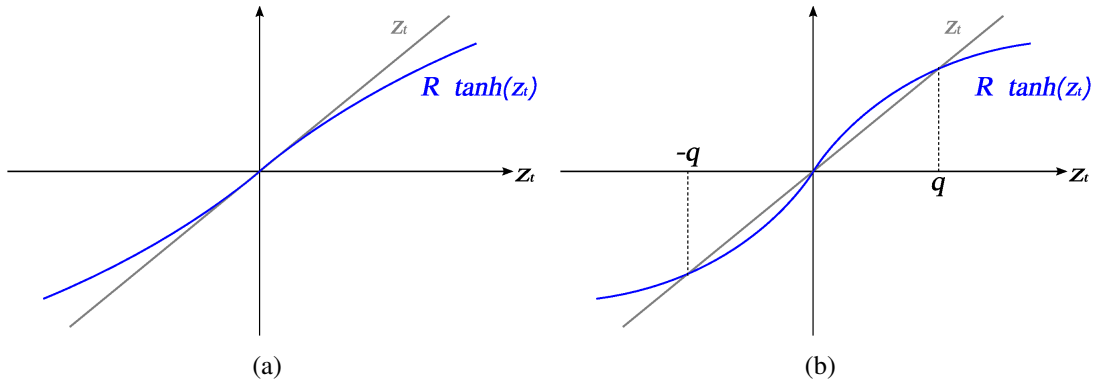


Figure 2.6: Graphical solution of the fixed point equation. (a) With a response strength $R < 1$ there is one point of intersection between z_t and $R \tanh(z_t)$ and hence one fixed point at $z^* = 0$. (b) With a response strength $R > 1$ two new fixed points at $z^* = \pm q$ appear and the fixed point at zero gets unstable.

increasing R this fixed point becomes unstable and two new fixed points with opposite signs at $z^* = \pm q$ appear. A plot of the stationary states of the system as a function of the feedback strength shows a Pitchfork bifurcation (see figure 2.7).

For the robotic application this means that with a feedback strength less than the critical value of $R = 1$ the robot will stall, while with a value above 1 the robot will move either forward or backward. The direction of motion is determined by symmetry breaking at the bifurcation point where the two stable branches with $z \neq 0$ branch off. If an assumed

controller is to generate activity, then the value of the feedback strength must be tuned to values above the critical value. Since $R = ca$ and the response factor a is a hardware constant of the system, the synaptic strength c must be adapted adequately.

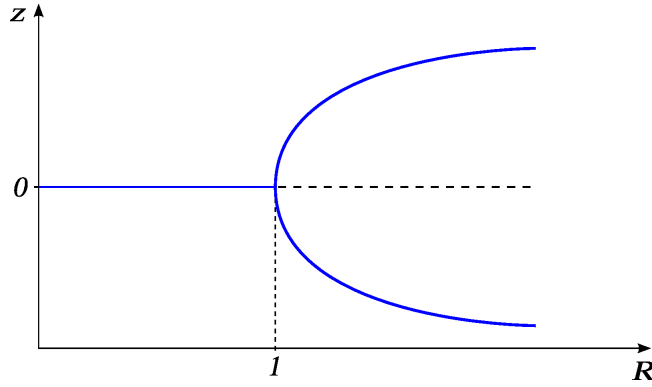


Figure 2.7: The stationary states of the system as a function of the feedback strength R show a pitchfork bifurcation for increasing values of R . The bifurcation point is at $R = 1$.

Stationary States of the System Including Bias

For stationary states of the system the update of the membrane potential z is zero, meaning equation 2.6 changes to

$$z_t = cx_{t+1} + h \quad (2.14)$$

and with equation 2.7 and 2.8 we get the fixed point condition of the system

$$z_t = ca \tanh(z_t) + h, \quad (2.15)$$

or with the feedback strength in the loop $R = ca$

$$z_t = R \tanh(z_t) + h. \quad (2.16)$$

The graphical solution of the fixed point equation is shown in figures 2.8 and 2.9. For values of the feedback strength R smaller than 1 and bias $h \neq 0$ the fixed point at $z^* = 0$ is shifted with the sign of the fixed point defined by the sign of h (see figure 2.8). Hence by increasing the absolute value of the bias, while $R < 1$, an open loop control is realized, neglecting the sensory feedback. Since we are interested in closing the sensorimotor loop and utilizing the sensory feedback this case has no further relevance.

We consider now the graphical solution of the fixed point equation for the case of the feedback strength $R > 1$. For absolute values of the bias h smaller than the critical value h_c the fixed points are slightly shifted as sketched in figure 2.9(a). When $h > h_c$ and $R > 1$ only one stable fixed point z^* with positive sign remains. After decreasing the bias until $h < -h_c$ one stable fixed point with negative sign remains, see figure 2.9(b). Plotting the stationary states of the system as a function of the feedback strength for $|h| > h_c$ shows that the pitchfork disconnects and there is now a globally stable, a locally stable and a locally instable branch (see figure 2.10).

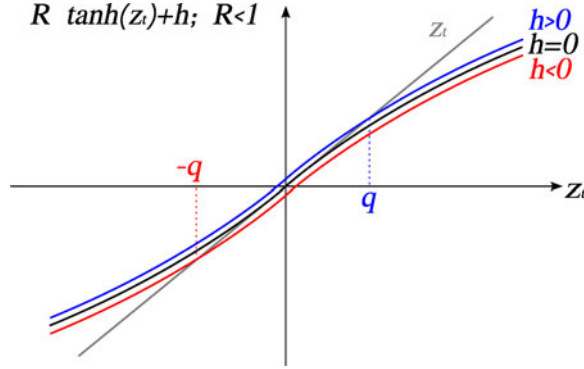


Figure 2.8: Graphical solution of the fixed point equation considering the bias h , with a feedback strength $R < 1$. With increasing absolute value of the bias h the fixed point at $z^* = 0$ is shifted, with the sign of the fixed point defined by the sign of h .

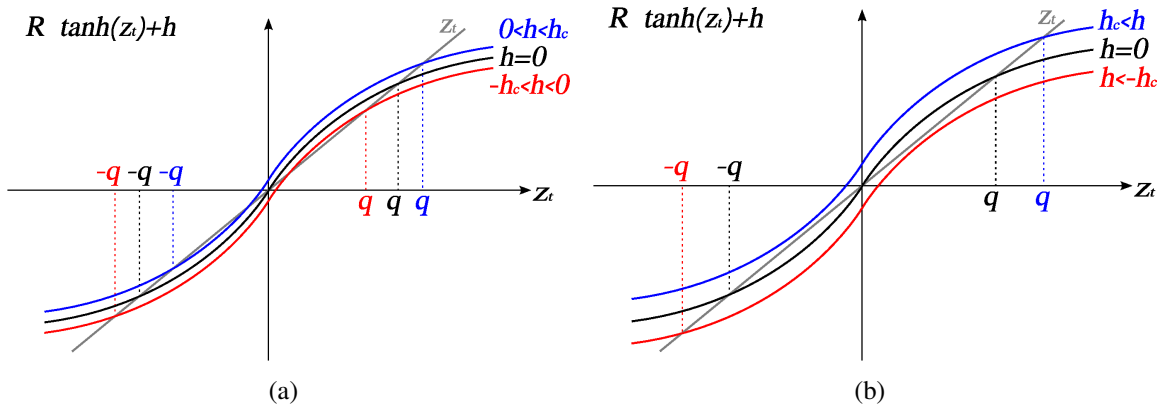


Figure 2.9: Graphical solution of the fixed point equation considering the bias h , with a feedback strength $R > 1$. (a) For $|h| < h_c$ the two stable fixed points remain and are slightly shifted. (b) For $|h| > h_c$ only one stable fixed point remains. If $h < -h_c$ the fixed point has a negative sign, if $h > h_c$ the fixed point has a positive sign.

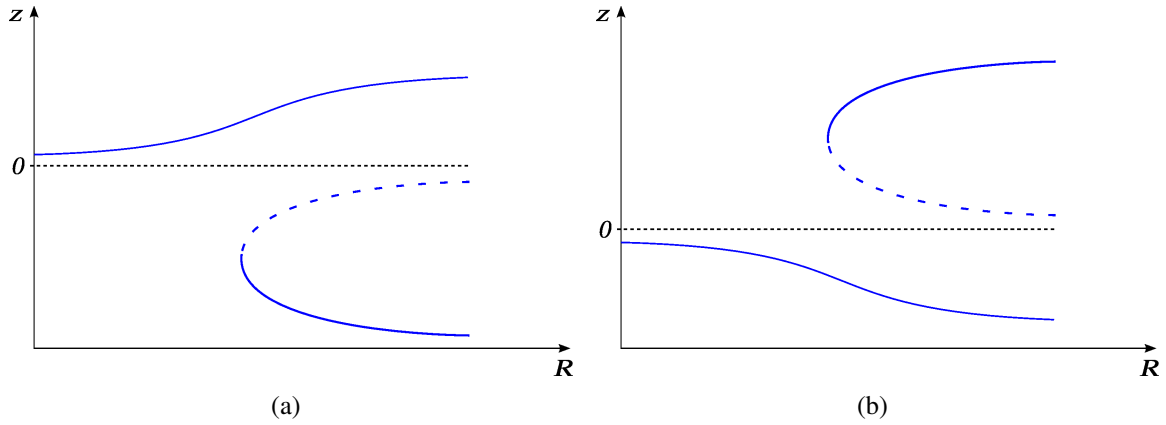


Figure 2.10: The stationary states of the system including bias h as a function of the feedback strength R show a globally stable, a locally stable and a locally unstable branch. For values (a) $h > h_c$ the states of the stable branch have positive sign, while for values (b) $h < -h_c$ they have negative sign.

For the robotic application this means that by modulating the bias h , the actual behavior of the system can be controlled. Since for values $-h_c < h < h_c$ there is a region of bistability, this will be hysteretic control as illustrated in section 2.2.2.4. Keeping $|h| > h_c$ for $R > 1$ realizes again open loop control neglecting the sensory feedback. Hence an assumed closed-loop controller could utilize the possibility to change the actual behavior by temporarily increasing $|h|$ above h_c , but should not keep the bias fixed in this regime.

2.2.2.3 Representation of the Dynamics as Gradient Descent

The concrete behavior of the system acting in the sensorimotor loop obviously depends essentially on the values of the synaptic strength c and the bias h . The essential features of this dynamics are illustrated best by using an alternative view of the dynamics of equation 2.10 obtained by considering the update of the membrane potential z as a gradient descent on a potential V

$$\tau \Delta z_t = -\frac{\partial}{\partial z_t} V(z_t) + c \xi_t. \quad (2.17)$$

Using

$$\frac{\partial}{\partial z} \ln(\cosh(z)) = \tanh(z) \quad (2.18)$$

we find that

$$V(z) = -R \ln(\cosh(z)) + \frac{z^2}{2} - hz, \quad (2.19)$$

again with $R = ca$ being the feedback strength in the sensorimotor loop. As usual, the gradient dynamics of equation 2.17 may be visualized by that of a sphere sliding down on the walls of a vessel filled with a viscous fluid, see figure 2.11. In our case we may use the small z approximation $\tanh z \approx z - \frac{z^3}{3}$ to get the more simple expression

$$V(z) = \frac{1}{2}(1 - R)z^2 + \frac{1}{12}Rz^4 - hz \quad (2.20)$$

for the potential.

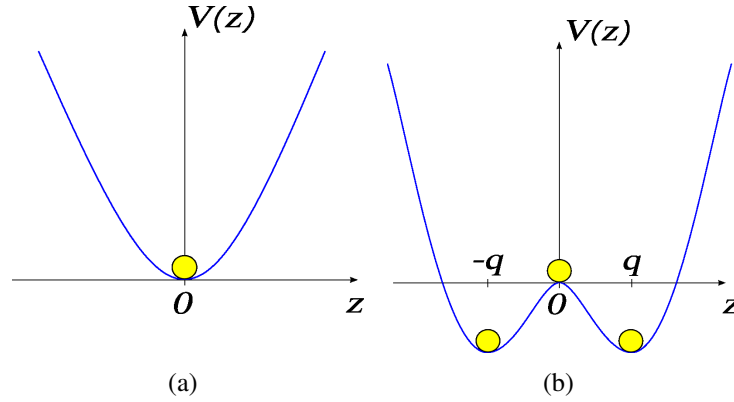


Figure 2.11: The state dynamics of the sensorimotor loop can be considered as a gradient descent on the potential $V(z)$, assuming the bias $h = 0$ here. Maxima of the potential correspond to unstable fixed points, minima to stable ones. The figure shows the potential with (a) $R < 1$ and (b) $R > 1$ where it is a double well potential with two stable and one instable fixed point. Each sphere represents a fixed point of the system.

The fixed points of the system acting in the sensorimotor loop correspond to the extrema of the potential, the stable fixed points being at the minima of the potential. In figure 2.11(a) the potential is plotted for $R < 1$ where the fixed point at $z^* = 0$ is indicated by the sphere. With increasing R we have a pitchfork bifurcation at $R = 1$, and for $R > 1$ we get a bistable system, i.e. we have one instable fixed point at $z^* = 0$ and two stable fixed points at

$$z^* = \pm \sqrt{\frac{3(R-1)}{R}}, \quad (2.21)$$

as indicated by the spheres in the double well potential in figure 2.11(b). The deviation of this approximate value from that for the exact potential in equation 2.19 is a only few percent in the pertinent region $1 < R < 1.3$ (see below).

The gradient dynamics will be used to illustrate the properties of the presented closed-loop control described in the next section.

2.2.2.4 Properties of the Closed-Loop Control

The Noise-Amplification Effect

As described above, the target wheel velocity of the robot $y = \tanh(z)$ is not given by some external description but is adjusting itself as a result of the dynamics of equation 2.10. In particular if $R > 1$ we may initialize the robot with any starting velocity and after some time its velocity will approach one of the two possible fixed point values, i.e. the robot will move either forward or backward with constant velocity. Which of the fixed points is realized depends on the starting value and possibly the noise. Even if the robot is inactive in the beginning, sensory noise will be amplified and bring the robot into motion.

One of the benefits of this closed-loop control system (under the assumption $R > 1$, h being 0) consists in the following. When colliding with an obstacle the wheels are blocked, so that $x = 0$ and z decays. If there is some (small) additional noise in the dynamics of the membrane potential z it will fluctuate around zero. These fluctuations can be amplified if they are of the right sign, i.e. if the robot is moving away from the obstacle. Hence after a short time the robot is found to move away from the obstacle, compare figure 2.12 and figure 2.13, the latter showing the result of an experiment with a real Khepera robot. (Details of the robot can be found in [Mondada et al. \(1994\)](#).) We may say that in this elementary sense the robot is able to survive.

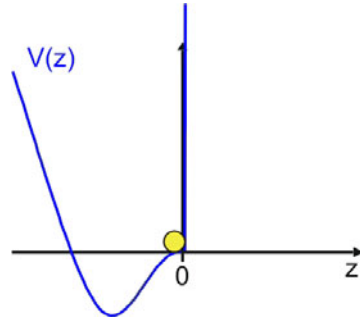


Figure 2.12: The state dynamics as gradient descent on the potential $V(z)$ after a collision with an obstacle. The robot was moving forward, i.e. the state was at the right hand side ($z > 0$) minimum of the double well potential. The impenetrable object corresponds to an infinitely steep rise in the potential so that the state is bound to move to the left hand minimum and the robot starts moving backward.

This holds also true for values of the bias $h \neq 0$ as long as both of the stable fixed points exist ($|h| < h_c$). As soon as only one stable fixed point remains ($|h| > h_c$) the robot will get stuck at the wall, because the direction of motion is determined by the bias alone, as explained in the next paragraph.

The Hysteresis Effect

In the case of finite bias h the fixed points and hence the velocity of the robot are obtained from

$$z_t = R \tanh(z_t) + h,$$

so that they also depend on the value of h and there is a hysteresis effect with respect to the change of h for the case that $R > 1$. The hysteresis effect results from the fact that there is a region $-h_c < h < h_c$ of bistability. Outside we have only a single fixed point which has the same sign as h . When moving the value of h from outside into the region of bistability the fixed point realized depends on which side of the outer region one is coming from, see figure 2.14 for details.

If in an assumed controller the bias would be fixed with $|h| > h_c$ we would obtain an open loop control, not able to react to environmental influences. Since we are interested in an active system showing environment-related behavior this should be avoided.

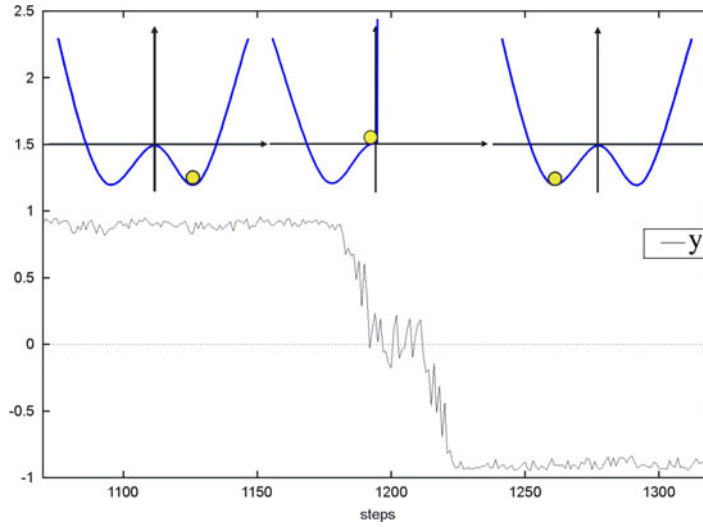


Figure 2.13: Neuron output y of the closed loop control system (with feedback strength $R > 1$) before and after a collision with an obstacle obtained from an experiment with a real Khepera robot. The robot is equipped with a one-dimensional controller, hence it is only able to drive forward and backward. Above the neuron output curve the corresponding fixed points of the system are represented by a sphere on the potential $V(z)$. Before the collision the system is in the fixed point with positive sign (robot drives forward). During step 1185 to 1215 the robot is kept at the unstable fixed point $z = 0$. Around step 1215 the robot starts moving backward because of the noise-amplification effect.

A plot of the state variable in dependence of the bias h as well as the feedback strength R shows a Hopf-bifurcation, see figure 2.15. With $R < 1$ there is only one fixed point $z^* = 0$, but with $R > 1$ we see the hysteresis effect in the (z, h) space which is the larger the larger R . From this diagram we can read the requirement $R > 1$ for activity in the system ($z \neq 0$) and the possibility to direct it by the hysteresis effect of h .

2.2.2.5 Quasi-Equilibrium

In many cases of practical interest one may assume that the dynamics of the membrane potential z is fast compared to the changes in the sensor values, so that Δz is small in equation 2.6 and we may approximately write

$$z_t = cx_t + h, \quad (2.22)$$

so that the controller output y is a direct function of the sensor values. Under this assumption we get a closed update rule for the sensor values as

$$x_{t+1} = \psi(x_t) + \xi_t, \quad (2.23)$$

where the time series predictor for the sensor values is

$$\psi(x_t) = ag(cx_t + h).$$

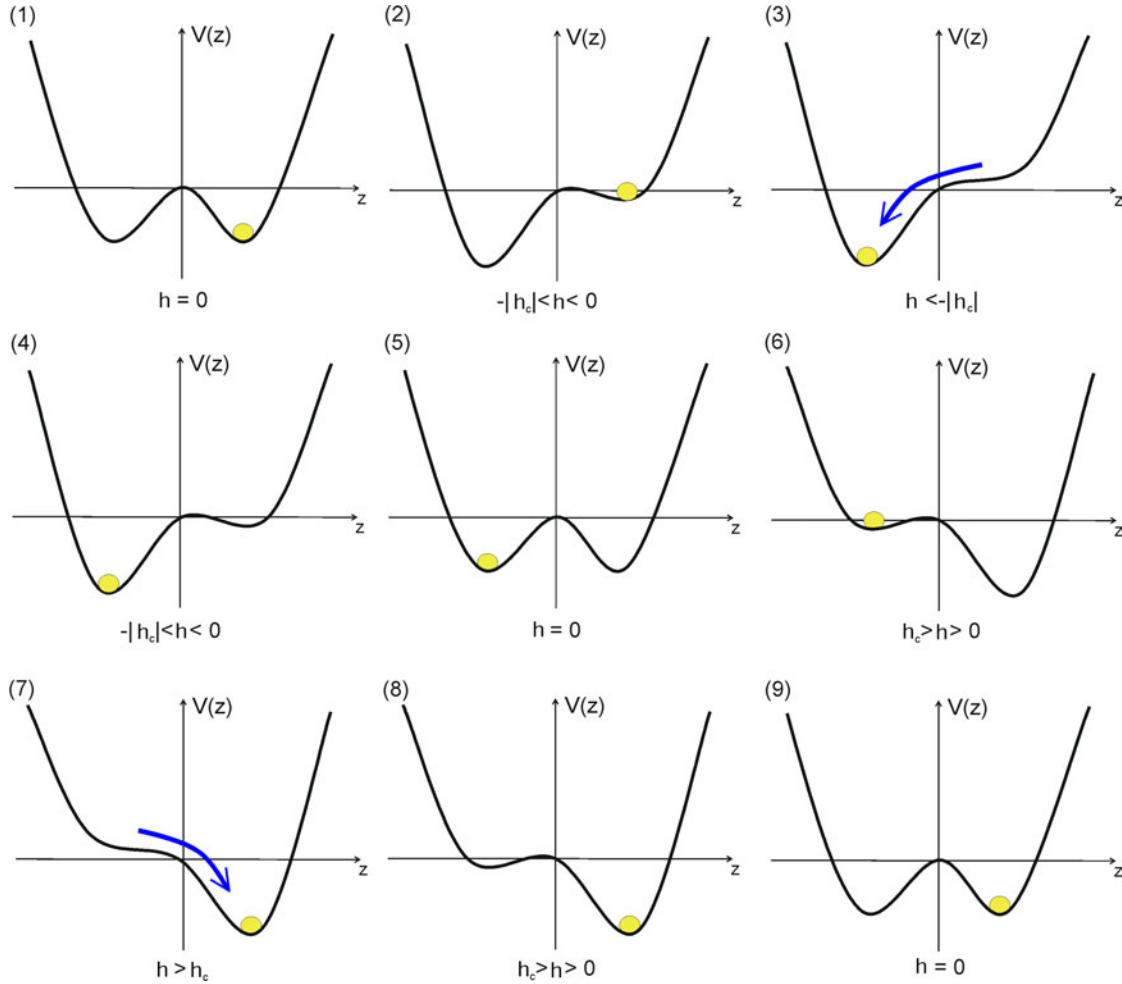


Figure 2.14: The hysteresis cycle. The diagrams show the stages of one hysteresis cycle starting from $h = 0$ (diagram (1)) with the state at $z > 0$ as represented by the sphere. Decreasing h leads to a deepening of the left minimum, while the right minimum becomes flatter, but the state remains at the minimum at $z > 0$, see diagram (2). If $h = -h_c$ both the maximum at $z = 0$ and the right minimum disappear so that the system shifts to the left minimum of the potential (3). Increasing h until $h = 0$ brings us back to the initial situation with the difference that the system changed to the fixed point with negative sign, cf. diagram (4,5). The diagrams (6) and (7) show the switching from the minima at $z < 0$ to the minima at $z > 0$ by increasing h . By decreasing h until $h = 0$ the hysteresis cycle is finished (see diagram (8,9)).

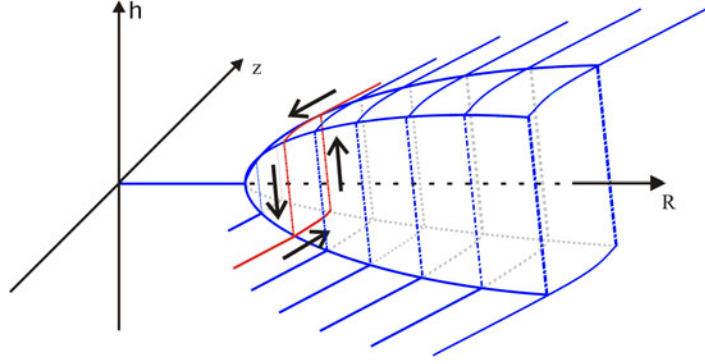


Figure 2.15: The state and parameter dynamics of the system for adiabatic changes show a Hopf-bifurcation. With $R < 1$ there is only one fixed point $z^* = 0$, but with $R > 1$ we see the hysteresis effect in the (z, h) space which is the larger the larger R .

By the same token we may also write

$$z_{t+1} = \text{cag}(z_t) + h + c\xi_t.$$

We will work with these approximations below.

2.2.3 Sensorimotor Loop with Several Sensors

In the more general case a controller neuron should be able to possess several sensory channels as depicted in figure 2.16. The sensor values $x \in \mathbb{R}^{n \times 1}$, with n being the number of

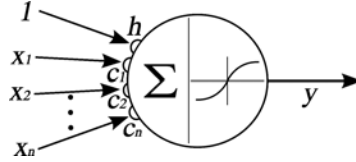


Figure 2.16: Sketch of the used neuron model with multiple inputs x_i , synaptic strength c_i , bias h and output y . The leaky integrator neuron has a dot product activation function and the output function is the hyperbolic tangent.

sensors, may now depend in a more general form on the motor command y . In the following we stipulate as in the one-channel case simple proportionality, i.e. we write the sensorimotor loop as

$$x_{i,t+1} = a_i y_t + \xi_{i,t}, \quad i = 1 \dots n,$$

where n is the number of sensors, or using vector notation

$$x_{t+1} = a y_t + \xi_t, \tag{2.24}$$

where the deterministic part may be considered as the mapping

$$F(x_t, y_t) = a y_t. \tag{2.25}$$

The parameters $a \in \mathbb{R}^{n \times 1}$ are again learned online using any form of supervised learning, where the samples (x_{t+1}, y_t) can be obtained online in each time step. The vector $\xi \in \mathbb{R}^{n \times 1}$ is considered as “noise” describing with each element the misfit between predicted and true sensor value. The controller is now

$$K(x_t) = g(z_t)$$

with $g(z) = \tanh(z)$ being the output function of the neuron. The update rule for the membrane potential z considering n sensors is

$$\tau \Delta z = -z_t + \sum_{i=1}^n c_i x_{i,t} + h$$

or in vector notation

$$\tau \Delta z = -z_t + c^T x_t + h$$

with $c \in \mathbb{R}^{n \times 1}$ being the vector of synaptic strength of the controller neuron, and with the threshold h still a scalar. Using equation 2.24 and $y_t = \tanh(z_t)$ as before we can write the closed loop dynamical system as

$$\tau \Delta z = -z_t + R \tanh(z_t) + h + c\xi, \quad (2.26)$$

where R

$$R = \sum_{i=1}^n c_i a_i$$

is the overall feedback strength in the sensorimotor loop. The fixed points of the system (assuming $\bar{\xi}_i = 0$ for all channels) are given by the solution of

$$z_t = R \tanh(z_t) + h.$$

Under the quasi-equilibrium assumption (section 2.2.2.5) we get the loop dynamics as

$$x_{t+1} = \psi(x_t) + \xi_t, \quad (2.27)$$

where now $x \in \mathbb{R}^n$ and

$$\psi_i(x_t) = a_i g \left(\sum_{i=1}^n c_i x_{i,t} + h \right).$$

2.2.4 Systems with Delay

Delays between the generation of the motor command and the corresponding response in the sensor values were not considered so far. Nevertheless, in practice there is always a finite delay time which can be larger than one time step. Together with Naglaa Hamed the question of delay in the considered system was investigated.

A first possibility is that the physical reaction time of the device under control is larger than one time step and there is no further delay time until the corresponding response to the executed motor command arrives at the sensors. A simple way to deal with this is to

introduce additional sensors which realize different delay times. The response strength a_i of the sensor with the required delay time will be the largest. As will be explained in section 2.5.2 the synaptic strength c_i will be the largest (as compared to the other c_j , $j \neq i$) and sensor i the most contributing one. Hence the system is able to select the adequate delay time, given the proposed setup.

Another possibility is that a system is able to physically react in a shorter time than the delay between the generation of the motor command and the corresponding response in the sensor values. In this case the amplification of noise events with different sign in successive time steps can drive the system into an oscillatory mode of behavior, where it strives for different fixed points in each time step. Assuming, for example, a signal propagation time of $m = 2$ time steps. Supposing the robot is able to realize a motor command in one time step, we obtain two independent dynamical systems

$$\begin{aligned} x_t &= \psi(x_{t-m}) \\ x_{t+1} &= \psi(x_{t-m+1}), \quad \text{with } t = 0, 2, 4, 6, \dots \end{aligned}$$

which are free to go to different fixed points. If this happens the systems comes into an oscillatory mode, switching between these fixed points in every time step. For active autonomous robots such a behavior is not desired. Imagine, for example, a wheeled robot inverting the target wheel velocity in each time step. It will show a tremor-like behavior and will not be able to explore the properties of body and environment. To overcome this we simply average the sensor values over a time window

$$\bar{x}_t = \frac{1}{N} \sum_{i=0}^{N-1} x_{t-i}$$

and use this averaged sensor values as input to the controller

$$y_t = K(\bar{x}_t).$$

In most practical applications $N = 2$ to 4 works fine (if required at all). For a detailed analysis of the oscillatory mode see the section *systems with delay* in [Hamed \(2007\)](#).

2.3 The Principle of Parameter Regulation

After discussing the properties of the closed-loop control in the sensorimotor loop and the effects of the parameters within this framework, we are interested in a regulation scheme for these parameters based entirely on intrinsic principles (section 2.3.3). Beforehand other approaches using the dynamical systems approach for robot control will be mentioned in the next section and the common approach to adaptive systems will be formulated in section 2.3.2.

2.3.1 The Dynamical Systems Approach to Robot Control

The time discrete stochastic dynamic system equation 2.27 is a mathematical description of the sensorimotor dynamics. Our approach is based on the dynamical systems formulation

(see e.g. Ott (1993), Strogatz (2001)) and tends to adapt the controller so that the robot behavior which is the manifestation of the dynamical system has the desired properties. Using the dynamical system as a substrate for the robot behavior has been considered by several authors in varying contexts and with varying success. Related to our subject is the work by Jun Tani (Tani, 2004; Tani and Ito, 2003) and of people around Gregor Schöner, (G. Schöner and Engels, 1995; Steinhage, 1997; Hock et al., 2003). An elaborate behavior-based design system has been developed in the context of dual dynamics. The system has a layered structure of behavioral subsystems realized by ordinary differential equations, each layer having its own time constant. Interactions between the subsystems are realized by specific interaction and “bifurcation-inducing” mechanisms which have to be designed by hand, cf. Bredenfeld et al. (2001).

The authors quoted have mainly tried to design dynamical systems such that they realize prescribed tasks, the smooth navigation through a cluttered environment being a prominent example. The main difference to the work presented here is that we design an objective for the self-regulation of the dynamical system without a concrete task given from outside. The behaviors emerging are therefore contingent, but the interesting point is that in the interplay between activity (through destabilization of the sensorimotor dynamics) and staying nevertheless predictable is the route towards the emergence of environment related behavior.

2.3.2 The Adaptive Systems Approach

The state dynamics of the system depends on the parameters c and h of the controller K . Changing these parameters changes the systems behavior. A system is adaptive if there is an objective function E , measuring the distance from the current to a desired behavior, which is used for the regulation of the parameters. This might be as abstract as measuring the survival properties of the system. One realization of the adaptation is a parameter dynamics as gradient flow

$$\begin{aligned}\Delta c_t &= -\varepsilon S_c(x_t, c_t) \\ \Delta h_t &= -\varepsilon S_h(x_t, h_t),\end{aligned}$$

where

$$S_c = \frac{\partial}{\partial c} E; \quad S_h = \frac{\partial}{\partial h} E,$$

so that we have the combined dynamics

$$\begin{aligned}x_{t+1} &= \psi(x_t, c_t, h_t) + \xi_t \\ c_{t+1} &= c_t - \varepsilon S(x_t, c_t) \\ h_{t+1} &= h_t - \varepsilon S(x_t, h_t).\end{aligned}$$

The problem of learning in robotics consists of finding the objective function E for the generation of a desired behavior. In the usual approach to adaptive systems the function E is provided from outside and is to be designed such that the system establishes the desired reference to the environment. We will consider a self-referential system, i.e. a system for which the objective function is derived from the dynamics of the system itself.

2.3.3 Defining the Objective Function

As already mentioned the homeokinetic principle is intended to generate self-organizing systems which are active and able to predict the results of their actions in terms of the sensor values. This predictive ability requires explicitly the inclusion of an internal model in the sensorimotor loop. Given this requirement there are many different paradigms to define a self-referential system. We want to start with probably the most simple ansatz, maximizing the predictability of the internal world model, but before the composed controller comprising controller and internal model will be introduced.

2.3.3.1 The Composed Controller

The internal model, realizing the predictive abilities of the robot, is seen as part of the controller. Therefore the controller can be considered as a composed controller consisting of a control and a model unit, as sketched in figure 2.17. The control unit requires sensory

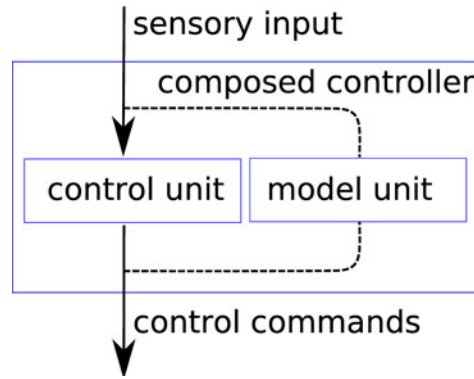


Figure 2.17: Sketch of the composed controller consisting of a control and a model unit. The connections between sensory input/control commands and model unit are plotted as dashed lines, because it depends on the concrete realization of the model which of these values are used.

input and generates the control command. The model unit can access sensory input and/or control commands, depending on the concrete realization of the model (see the following paragraphs). Throughout this thesis the term model refers to the model unit and controller to the control unit of the composed controller.

2.3.3.2 Using a Predictive Internal Model: Homeostasis

We are considering now the mapping F (see equation 2.2) of the motor value y_t to the predicted sensor value x_{t+1}^M for the next time step

$$x_{t+1}^M = F(y_t) \quad (2.28)$$

as the internal model or model unit of the composed controller. The norm of the modeling error $\xi_t = x_{t+1} - x_{t+1}^M$, which accounts for the difference between the true sensor value x and the predicted sensor value x^M , is then used as objective function

$$E = \|\xi_t\|^2 = \xi_t^T \xi_t \quad (2.29)$$

for the adaptation of the model and the controller parameters. A sketch of the control architecture in the sensorimotor loop is shown in figure 2.18. The parameters are adapted by

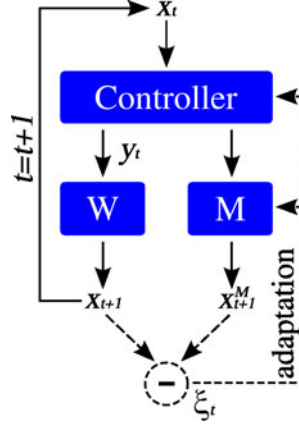


Figure 2.18: Sensorimotor loop with a predictive world model (M) and indicated learning scheme. The modeling error ξ , defined as the difference between the measured and predicted sensor values at time step $t + 1$, is used for the adaptation of world model and controller. The world is represented as box labeled with W .

gradient descent on E (compare section 2.3.2)

$$\Delta p = -\varepsilon_p \frac{\partial E}{\partial p}, \quad (2.30)$$

where p is any of the parameters (controller and world model) on which the loop function ψ depends and ε_p is an adaptation rate. Assuming a linear world model

$$x_{t+1}^M = F(y_t) = ay_t$$

and together with equations 2.7 and 2.22 we find

$$x_{t+1}^M = a \tanh(cx_t + h).$$

Following equation 2.30 we obtain the update rules for the synaptic strength c , the bias h and the model parameter a :

$$\begin{aligned} \Delta c_i &= \varepsilon_c \xi_t a \tanh'(z_t) x_{i,t} \\ \Delta h_i &= \varepsilon_h \xi_t a \tanh'(z_t) \\ \Delta a_i &= \varepsilon_a \xi_t y_t \end{aligned}$$

By gradient descent on E , distracting influences in the sensorimotor loop are prevented by damping the noise in the in the sensorimotor loop. This is done by not utilizing the sensor values.

In a robotic application this setup leads to a situation where the system is stabilized in a state with motor command zero, allowing the best prediction of future sensor values. Hence the most favored behavior of systems following this attempt is the “do nothing” behavior.

We can consider the error function resulting from the predictive model as a realization of the homeostasis principle, introduced by Cannon (1939). It tries to explain the behavior of a system, especially living organisms, as a kind of regulation scheme. There are thought to be several intern variables of the agent that have to be kept in a certain range, such as temperature, blood-pressure or energy (food). If one of these variables leaves its range the organism tries to bring the value back to its set value through specific actions, which in turn can cause other variables to change.

It can be shown, that this regulation scheme works well and keeps the variable in a certain region around the reference value. The problem with this attempt is how to obtain the reference value when creating a new system. This question can hardly be answered in a general and domain-invariant manner.

In the case of equation 2.29 the reference value, which should be realized by the system, is the sensor value x_{t+1} . The problem in generating an active behavior of an autonomous robot becomes obvious. If the robot is in a situation where the sensor values are zero, these sensor values are used as reference value and hence the system has no drive to change the situation. If the robot is started with sensor values different from zero, the objective function leads to a stabilization of these values, but as soon as the sensor values change heavily due to noise effects, collisions or something else happening in the world, the reference value of the system is changed. Since changes like friction or collisions which decrease the sensor values are more likely, these systems will come to the “do nothing” behavior. Imagine, for example, a wheel-driven robot bumping into a wall. The wheel velocity will jump to zero. Since the sensory input x is the reference value in the homeostatic regime the robot will not recover and stay at the wall indefinitely.

We have seen that it is possible to generate a predictive behavior of an autonomous robot using an internal predictive model of the future sensor values. The problem is, that this is not an active behavior. Of course it is possible to achieve active behavior by setting the reference value from outside. But then we lose autonomy and self-organization because the system is guided from outside.

To overcome this problem with the given modules of the architecture, namely the internal world model and the controller, we can build a different setup by using the internal model as a retrospective one.

2.3.3.3 Using a Retrospective Internal Model: Homeokinesis

The following principle was proposed by Ralf Der under the name of homeokinesis, as a dynamical pendant to Homeostasis. Preliminary work on this approach, where an external drive for activity is required, can be found in Der et al. (1999); Der and Pantzer (1999); Der (2000, 2001). The present form, where the drive for activity also emerges from the self-organization paradigm, was proposed in Der and Liebscher (2002); Der (2003). Here the internal model or model unit is a retrospective model. Hence the model is a function mapping the future sensor values x_{t+1} to the controller output y_t . Such a model can be learned with a backed up motor command from the previous time step. The predicted previous motor command is then propagated back through the controller to result in a reconstructed sensor value x_t^R , as depicted in the sketch in figure 2.19. Consequently applying the loop function

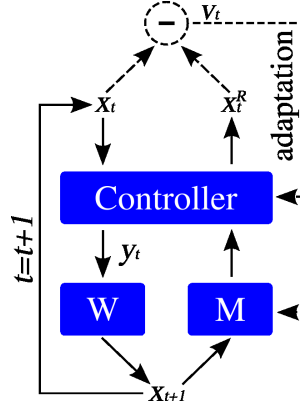


Figure 2.19: Sensorimotor loop with a retrospective world model (M) and indicated learning scheme. The input shift v , measuring the difference between the old sensor value and the reconstructed sensor value at time step t , is used for the adaptation of world model and controller. The world is represented as box labeled with W .

ψ to the reconstructed sensor value, we will obtain the measured sensor value at time $t + 1$.

$$x_{t+1} = \psi(x_t^R).$$

Since ψ is not always invertible, x_t^R can not be exactly calculated in all cases. This can be overcome by a regularization, as described later.

When employing the retrospective model we can formulate the input shift v as the difference between the reconstructed and true sensor values at time step t

$$v_t = x_t^R - x_t. \quad (2.31)$$

With this notion we can define our objective function

$$E = \|v_t\|^2 \quad (2.32)$$

measuring the deviation between the current and the reconstructed sensor values. This deviation arises by propagating x_t through the controller and world, and then propagate it back through model and controller (see figure 2.19). Thus E is the error arising in a time loop and is therefore referred to as the time-loop error, as introduced by Ralf Der (Der and Liebscher, 2002). In equation 2.32 we use the Euclidean norm

$$\|v_t\|^2 = v_t^T v_t.$$

Recalling the sensorimotor dynamics from equation 2.27 we can write

$$x_{t+1} = \psi(x_t) + \xi_t = \psi(x_t^R), \quad (2.33)$$

which can be rewritten as

$$\psi(x_t) + \xi_t = \psi(x_t + v_t). \quad (2.34)$$

If v is small we may use Taylor expansion to write

$$\psi(x_t + v_t) = \psi(x_t) + L(x_t) v_t, \quad (2.35)$$

where the Jacobian matrix L is defined as

$$L_{ij} = \frac{\partial}{\partial x_{j,t}} \psi_i(x_t).$$

Using equation 2.35 in equation 2.34 we find

$$v_t = L^{-1}(x_t) \xi_t$$

and obtaining v means now “only” to find the inverse of the matrix L provided the latter exists. Equation 2.32 may now be written

$$E = \|L^{-1}\xi_t\|^2,$$

respectively

$$E = (L^{-1}\xi_t)^T (L^{-1}\xi_t). \quad (2.36)$$

Hence the objective function only depends on the Jacobian of the system and the modeling error ξ of the predictive model. Since L is based on the response strength a as obtained from the predictive model, the calculation of the objective function E requires only the (forward) loop function $\psi(x)$. So when realizing such a system we can use the predictive model F for the calculation of the objective function based on the retrospective model.

The explicit expression in equation 2.36 displays the main properties of the gradient flow in the parameter space induced by the gradient descent on E . On the one hand E will be small if the vector $\xi_t = x_{t+1} - \psi(x_t)$ is small, i.e. if the robot behaves in a predictable way. On the other hand the Jacobian matrix determines the local stability of the dynamical system defined by ψ . With L in the denominator of E the gradient descent will destabilize the sensorimotor dynamics. This way small changes, such as sensory noise, will be amplified and lead to larger changes in the loop. Hence different behaviors will be actively generated within the sensorimotor loop, although an unlimited increase of sensor values and motor commands is confined by the nonlinearities in the system.

From these simple arguments it can already be anticipated that bestowing the sensorimotor dynamics with a gradient flow in parameter space driven by E will produce a system with very rich properties.

2.4 Parameter Dynamics in the Elementary Sensorimotor Loop

In this section the parameter dynamics for the case of a simple loop with one sensor and one actuator will be derived and the properties of the dynamical system under the presented objective function E will be discussed.

In the elementary sensorimotor loop already introduced in section 2.2.2 we have $x \in R^1$ and $F(x_t, y_t) = ay_t$ so that

$$\psi(x_t) = ag(z_t)$$

with $z_t = cx_t + h$. The Jacobian is

$$L = \psi'(x_t) = Rg'(z_t)$$

with $R = ca$ being the feedback strength in the sensorimotor loop. The error E boils down to

$$E = \frac{\xi_t^2}{L^2}. \quad (2.37)$$

The derivative is written as

$$\frac{\partial E}{\partial p} = -2 \frac{E}{L} \frac{\partial L}{\partial p} - \frac{2}{L^2} \xi_t \frac{\partial \psi(x_t)}{\partial p},$$

where $p \in \{c, h\}$. We assume here that $\bar{\xi} = 0$ so that the ξ term does not contribute to the parameter dynamics in the average over the noise. Using $L > 0$ (see below) and $g'' = -2gg'$ in the case $g(z) = \tanh(z)$ together with $Rg(z) = z - h$ at the fixed point we obtain

$$\begin{aligned} \Delta c &= \mu a - 2\mu x_t (z_t - h) \\ \Delta h &= -2\mu (z_t - h), \end{aligned} \quad (2.38)$$

where

$$\mu = 2\varepsilon E/R$$

(R will be seen to be positive below) is a modified update (learning) rate, and $g' = \tanh'(z) = 1 - \tanh^2(z)$. We will see below that the system goes into a limit cycle in the x, h space. Averaging over a period and using that the amplitude of h is much smaller than that of z we may simplify the parameter dynamics further to

$$\begin{aligned} \Delta c &= \mu a - 2\mu x_t z_t \\ \Delta h &= -2\mu z_t. \end{aligned} \quad (2.39)$$

The parameter dynamics is to be used concomitantly with the z dynamics so that the parameters c and h in equations 2.6 or 2.10 are now time-dependent. As we will see below, the time scale for the change of h in particular is on the level of the behavior so that in other words the behavior is essentially controlled by the dynamics of h . This is different from the usual paradigm of learning where we have a learning and a performance phase or where there is a separation of time scales for learning and behaving.

The dynamics for the synaptic strength c (equation 2.39) consists of a driving and an anti-Hebbian term. The latter is given by the product of the input into the synapse times the membrane potential of the neuron, both quantities being felt directly at the synapse. Neurophysiological evidence for synaptic modifications based on Hebb's postulate (Hebb, 1949) (as a form of long-term potentiation) was found in the 1980s, as discussed in detail in the review articles (Brown et al., 1990), (Bi and Poo, 2001). The driving term μa is given by the response strength a of the sensor x to the output y of the controller. This term can also be obtained empirically by modulating the neuron output with a periodic perturbation and filtering this signal from the sensor values. Hence we may say that the learning rule is a purely local one. This holds also true for the update rule of the bias h .

In the following the most interesting properties observed when using the parameter dynamics equation 2.39 with different kinds of robots will be discussed for the case of the elementary sensorimotor loop.

2.4.1 The Generation of Activity

Let us consider the case with bias $h = 0$ first. The update rule for the synaptic strength c (equation 2.39) consists of the driving term μa and the anti-Hebbian term $-2\mu x_t z_t$. In order to discuss the effects of the two terms we assume that we start the system with the feedback strength in the sensorimotor loop $0 < R \ll 1$ (the tabula rasa condition). Since $R = ca$ the synaptic strength c of the controller is also very small so that the membrane potential z fluctuates around zero. With $y = g(z) = \tanh(z)$ the motor command also fluctuates around zero. Hence a robot will at most make some small movements, if it moves at all. With $z \approx 0$ the anti-Hebbian term is negligible and the driving term is seen to increase the value of R since $\Delta R = \Delta(ca) = \mu a^2$, see figure 2.20. Once a supercritical value of the feedback

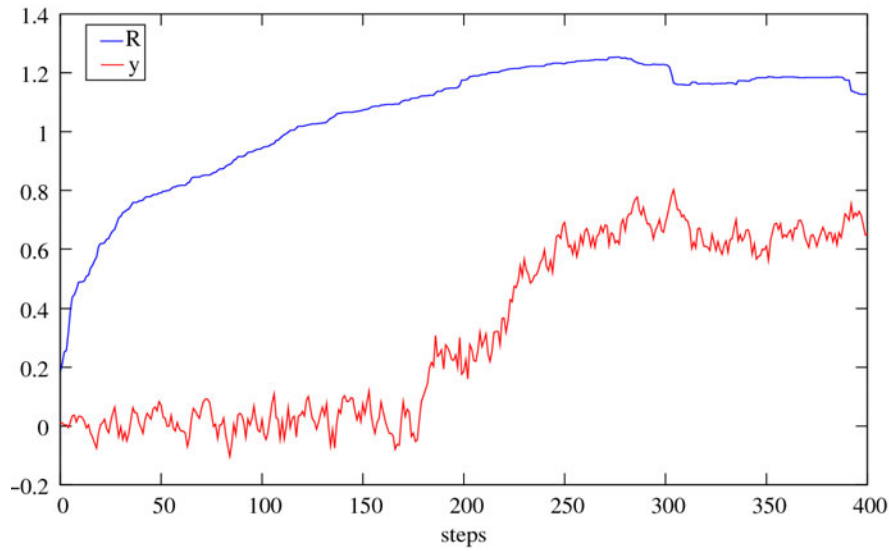


Figure 2.20: The increase of the feedback strength R due to the learning procedure, obtained from an experiment with a simulated Khepera robot using the Khepera Simulator (Michel, 1995). Initially the rise is very steep due to the fact that L is in the denominator. If R is above the critical value the target wheel velocity (motor command y) increases with the sign (direction of motion) determined by the noise amplification effect.

strength is reached ($R > 1$), noise amplification (see section 2.2.2.4) takes place. Hence small sensor values are amplified in the loop. Thus sensory noise is sufficient to generate activity and the velocity will increase exponentially so that the robot starts to move. The direction of the robot (sign of the velocity) is arising from a spontaneous breaking of the $x \rightarrow -x$ symmetry inherent in the complete (i.e. parameter and state) dynamics.

Thus the driving term leads to activity in the system (generate motor commands $y \neq 0$) by increasing the feedback strength in the loop and consequently destabilizing the system (minimize $\frac{1}{L^2}$).

2.4.2 Requiring Predictability

As described above the driving term leads to an increase in the feedback strength R of the sensorimotor loop when the system is not active ($y \approx 0$). An equally important point is that

the feedback strength does not increase unbounded, otherwise, if small changes in the sensor values would lead to huge changes in the motor commands, the behaviors of the system will be hardly predictable. This is realized by the anti-Hebbian term in the update rule which decreases the feedback strength to a slightly supercritical value, as soon as there is activity in the system ($y > 0$). With $|y| > 0$ the anti-Hebbian term becomes considerable and the increase of c is stopped if $a = 2zx$ is reached, i.e. if $1 = 2z \tanh z$ or $1 = 2Ry^2$ which happens at $R \approx 1.2$ corresponding to $y \approx \pm 0.65$. If R was already larger when the robot started to act, c will be decreased until $R \approx 1.2$. Note that R is the feedback strength in the sensorimotor loop so that we observe a self-regulation of the system to a feedback strength which is slightly supercritical.

An interesting interplay between the driving and the anti-Hebbian term is observed if the robot hits an obstacle. We have seen in section 2.2.2.4 that with R fixed the robot will invert the velocity after some time. In the present case this is accompanied by an increase in the feedback strength R due to the fact that with $x = 0$ only the driving term is active in the parameter dynamics. Increasing R means increasing the noise amplification. Hence small sensory noise events showing away from the obstacle will be amplified. Furthermore, as soon as the robot moves away from the obstacle ($|y| > 0$) the feedback strength decreases due to the anti-Hebbian term in the update rule (see figure 2.21 for details).

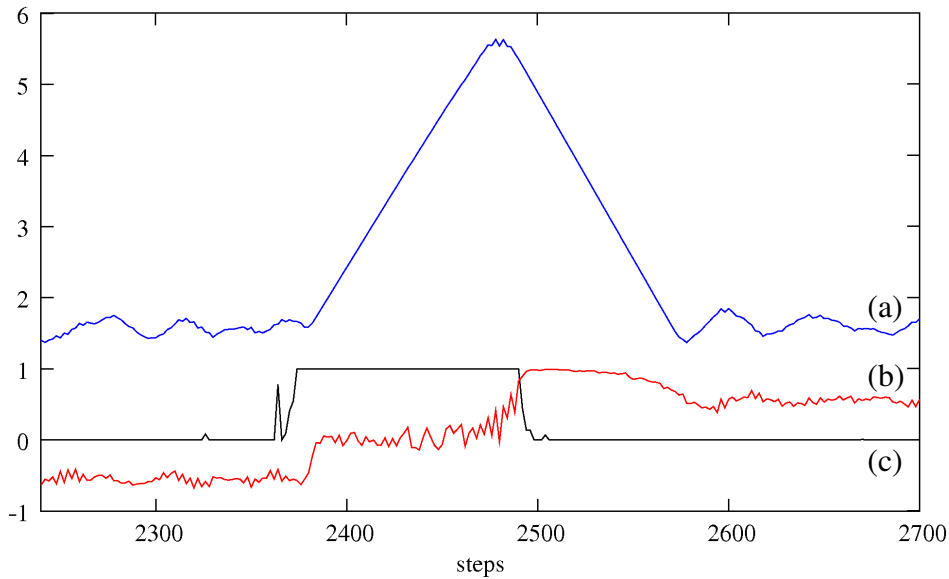


Figure 2.21: Time course of the synaptic strength c (a) during a wall contact of the wheeled robot as indicated by the infrared sensors (c). c and hence the feedback strength is seen to increase during the contact and to decrease again (due to the anti-Hebbian term in the parameter dynamics) when the robot is moving away from the obstacle. The increase in the feedback strength increases the noise amplification and by this the escape probability (see motor command y (b)). The data is obtained from an experiment with a simulated Khepera robot using the Khepera Simulator.

2.4.3 The Frequency Effect

With $h \neq 0$ there is a hysteresis effect with respect to the change of h as discussed in section 2.2.2.4. Equation 2.39 shows that the change of h always aims at destabilizing the actual fixed point of the system ($\Delta h \sim -z$). Hence the system executes the hysteresis cycle shown in figure 2.14. But the system does not exactly reach the fixed points because of the rapid change of h (as compared to the other parameters). Thus the shape in figure 2.15 is washed out and it is easily seen that a smooth limit cycle behavior is obtained. The value of c is seen to slightly oscillate with twice the h frequency, but in the average the feedback strength is self-regulating again to the slightly supercritical value of $\bar{R} \approx 1.2$ (cf. figure 2.22). We may consider the transition to the limit cycle as a self-induced Hopf-bifurcation in the

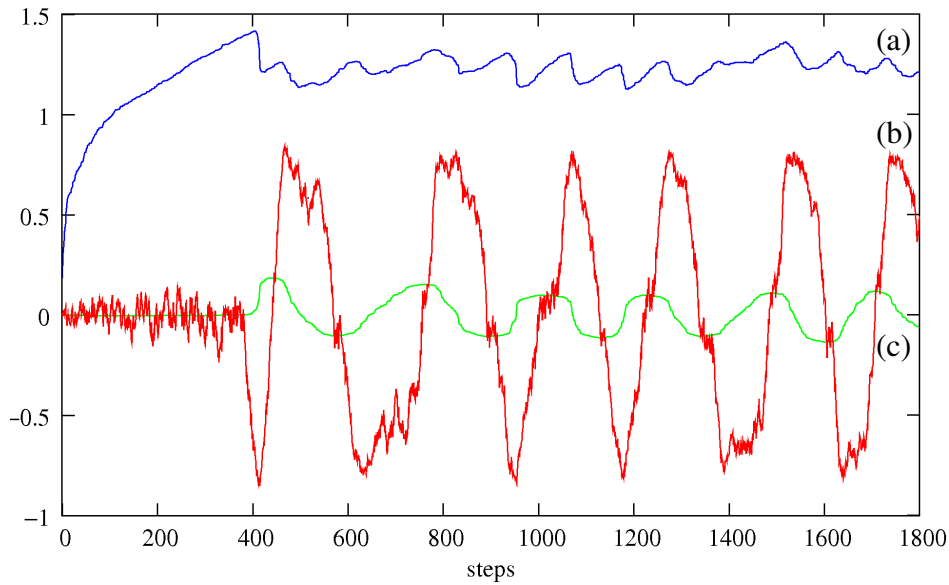


Figure 2.22: The increase and self-regulation of the feedback strength R (a) due to the learning procedure, obtained from an experiment with a simulated Khepera robot using the Khepera Simulator. c and therefore R is oscillating with twice the frequency of h (c) which is determined by the strength of the noise ξ^2 . The phase shift between h and y (b) is a consequence of the hysteresis effect. The Hopf-bifurcation takes place at about step 400.

(x, h) space where the value of c is self-regulating to the regime slightly above the bifurcation point, as described above.

The frequency of the limit cycle oscillation is modulated by the strength of the noise ξ^2 , which we call the frequency effect. With varying noise strength the robot will execute an irregular searching behavior, i.e. the robot will move forward for some time then reverse velocity and move backwards and so on. The most interesting property however is observed when the robot collides with some obstacle so that the wheels get blocked. Then ξ^2 in equation 2.38 is very large so that, in addition to the increase of the feedback strength by the driving term, the rate of change of h largely increases and the robot will almost immediately reverse its velocity (see figure 2.23).

From the elementary case we want to step to the next section, discussing the general properties of the approach when more sensors are used.

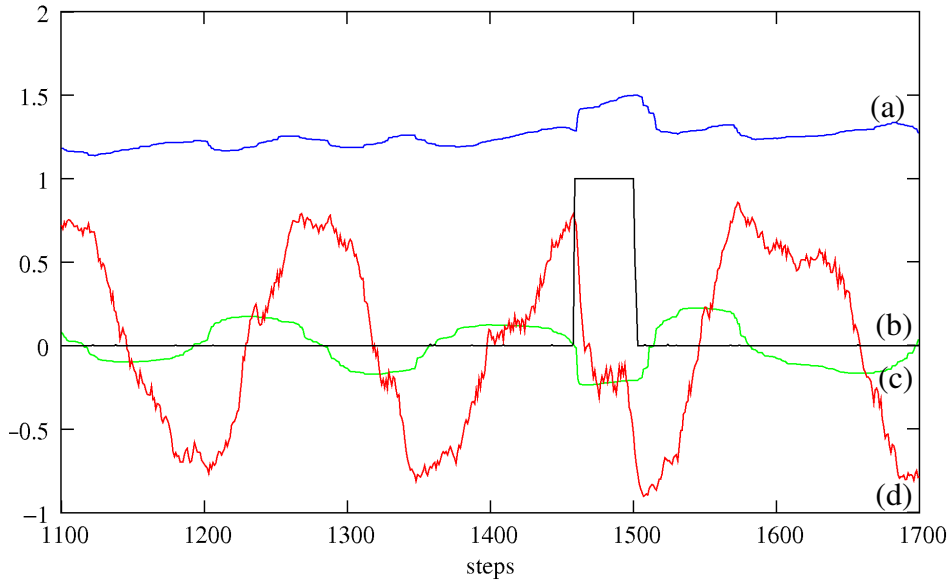


Figure 2.23: Time course of the response-strength R (a) and bias h (c) during wall contact as indicated by the infrared sensor (b). When the wheels get blocked the model error ξ^2 is very large so that the rate of change of h largely increases and the target wheel velocity y (d) is nearly immediately reversed. The data is obtained from an experiment with a simulated Khepera robot using the Khepera Simulator.

2.5 Parameter Dynamics Using Several Sensors

For most applications controller neurons should be able to process information from several sensors in order to generate the motor command. Therefore we investigate the case of a single controller neuron with multiple sensory channels in order to find the parameter dynamics and properties of the homeokinetic approach in this case.

In the sensorimotor loop with n sensory channels as presented in section 2.2.3 we have $x \in \mathbb{R}^{n \times 1}$ and the internal model $F(x_t, y_t) = ay_t$ with the parameters $a \in \mathbb{R}^{n \times 1}$. The loop dynamics under the quasi equilibrium assumption is

$$x_{t+1} = \psi(x_t) + \xi_t,$$

where

$$\psi(x_t) = ag(z_t).$$

The membrane potential z is the weighted sum of the sensory inputs plus bias h

$$z_t = \sum_{i=1}^n c_i x_{t,i} + h.$$

Hence the Jacobian L is

$$L_{ij} = \frac{\partial}{\partial x_{j,t}} \psi_i(x_t) = a_i c_j g'(z_t). \quad (2.40)$$

The principle needs some customizing since the input shift v , required for the objective function (equation 2.32) is not uniquely defined. Formally this is seen by the fact that L is

not invertible. We may remove the ambiguity by making an assumption on the direction of v . In the following we stipulate that v is in the direction of the response strength a given by the internal model. The reason behind this is that the inputs x should be produced by the deterministic part of the sensorimotor dynamics. Hence x is proportional to a apart from the noise. Using the approximation $\xi_t = Lv_t$ the error function used so far is

$$E = \|v_t\|^2 = \|L^{-1}\xi_t\|^2$$

minimizing the input shift v . Considering the projection of ξ on a as the relevant property of the noise we obtain our new objective function as

$$E = \|u_t\|^2,$$

where

$$\|u_t\|^2 = \|\Lambda^{-1}\xi_t \cos \phi\|^2,$$

ϕ is the angle between the vectors ξ and a , and

$$\Lambda = g'(z_t) R.$$

For more details on the derivation of the objective function see appendix A.

Following the lines of section 2.4 we obtain the following rules for the dynamics of the parameters

$$\begin{aligned} \Delta c_i &= \mu a_i - 2\mu z_t x_{i,t} - \gamma \mu c_i \\ \Delta h &= -2\mu z_t, \end{aligned} \tag{2.41}$$

where $\mu = 2\varepsilon u^2/R$ and γ (which is small) was introduced in order to produce a (weak) decay of the weights. This is necessary in order to dampen that part of the initial conditions which is orthogonal to a . We again find that the change of c_i is given by a driving together with an anti-Hebbian term and that the change of the bias is counteracting the membrane potential ($\Delta h \approx -z$).

The discussion of the effects of the terms in the update rule for the multi-channel case will mainly focus on the difference to the mostly similar case with a single sensor, which was discussed in detail in section 2.4. Furthermore, experiments for the single and multi-dimensional case which exemplify the properties of the approach in applications to robotic devices will be shown in chapter 3.

2.5.1 The Generation of Activity

In order to analyze the parameter dynamics we start with an initialization of the synaptic strength c such that the feedback strength is $0 < R \ll 1$. Then the motor command y is fluctuating around zero and the damping term $-2\mu z x_i$ is very small since the membrane potential z is very small. Hence the driving term in the learning dynamics dominates and produces $\Delta c_i = \mu a_i$ so that $\Delta(c_i a_i) = \mu a_i^2$ and hence $\Delta R = \mu a^2$. Obviously the overall feedback strength R increases with channels of higher response strength $|a_i|$ being favored. Once R exceeds the critical value $R_c = 1$, activity in the system can be generated by amplification of sensory noise.

2.5.2 Requiring Predictability

If the system is active the sensor values x_i and the membrane potential z become considerable and hence the anti-Hebbian term of equation 2.41 comes into play. The parameter dynamics becomes stationary (neglecting the bias h here) if

$$\gamma c_i = (1 - 2Ry_t^2) a_i$$

or

$$R = \frac{1}{\gamma/a^2 + 2y_t^2} = \frac{a^2}{\gamma + 2x_{t+1}^2}, \quad (2.42)$$

so that

$$c_i = \alpha a_i. \quad (2.43)$$

Obviously the c_i reach values so that all sensors are integrated into the sensorimotor loop according to their response strength a_i as obtained from the internal model equation 2.25. Thus sensors showing a response to the motor commands are integrated in the sensorimotor loop with a synaptic strength $|c_i| > 0$. Non-responding sensors are barely integrated with $c_i \approx 0$.

The value of α is obtained as

$$\alpha = \frac{1}{\gamma + 2x_{t+1}^2} = \frac{R}{a^2}.$$

Using the fixed point equation in the form $y_t = \tanh(Ry_t)$ together with equation 2.42 we get

$$y_t = \tanh\left(\frac{y_t}{\gamma/a^2 + 2y_t^2}\right).$$

The position of the fixed point is seen to depend smoothly on the value of γ/a^2 for not too large γ . For instance the fixed point is at $y = 0.58$ and $R = 1.15$ if $\gamma/a^2 = 0.2$ which is only slightly lower as compared to $y = 0.65$ and $R = 1.19$ corresponding to $\gamma \rightarrow 0$ and also to the one-channel case.

2.5.3 The Frequency Effect

When including the threshold (h) dynamics a limit cycle behavior is again obtained. The increase of the number of sensory channels has no effect on the update rule of the threshold and hence the properties described in section 2.4.3 remain valid in the average over (at least) one period for the multi-channel case.

2.6 Discussion

We have presented a control paradigm for autonomous robots based entirely on information available to the agent itself. No domain-specific information is required to set up such a system. The resulting parameter dynamics is seen to generate activity in the system and show an explorative behavior which stays sensitive to the environment.

It should be noted that these properties are not the performance of the trained neuron but instead result from the interplay of state and parameter dynamics, i.e. the concomitant effects of equations 2.6 and 2.38/2.41, so there is no separation between a learning and a performance phase or between the timescales of learning and behavior. Hence we obtain systems with closed-loop control and plasticity. Therewith we are approaching the challenge for achieving more understanding of the brain stated by Ahissar and Kleinfeld (2003) (and mentioned in section 2.1.2).

In the case of several sensors it is important to point out that the system is self-regulating into the limit cycle oscillations, independent of the number of channels and the values of the response strength a_i in the channels. Sensors which respond to the executed motor commands are integrated in the control loop according to equation 2.43. So the system is able to close the sensorimotor loop without a (previous) knowledge of the interconnection between sensors and motors. Furthermore, since the feedback strength is self-regulating to a slightly supercritical value activity is generated in the loop. Hence the irregular, environment-sensitive explorative behavior is reproduced also in the case of many channels where each channel is related to a single sensor.

In the presented realization of the homeokinetic principle there is a limitation introduced. So in my work only a simple (in most cases linear) internal model is used, which constrains the prediction ability and hence the ability of the system to adapt to complex situations. This way only a linear response of the sensors is taken into account which favors the use of proprioceptive sensors inside the loop. Nevertheless it is amazing how self-referential systems equipped with a parameter dynamics following the proposed principle can generate activity independent of their concrete physical realization and show body- and environment-related behavior.

Control for robotic devices with multiple degrees of freedom can be easily realized by introducing as many controller neurons as actuators are available. While trying to reduce the computational effort when using such systems it turns out that a rather crude approximation of the updates of the synaptic strength and the threshold is still appropriate and sufficiently fast so that a real time behavior of large systems can be achieved.

The next chapter will describe in detail experiments with real and physically realistic simulated robots under homeokinetic control, differing in shape, sensor modalities, actuators and the number of available degrees of freedom.

Chapter 3

Applying the Homeokinetic Principle to Autonomous Robots

This chapter is intended to show the application of the proposed algorithm for the self-organizing control to different kinds of simulated and real autonomous robots.

In the first section, the framework will be introduced, which was used for the physically realistic simulation of autonomous robots, the *lpzrobots* software package. Section 3.2 points out the main differences between homeokinetic and static control of an autonomous robot. The explorative character of the behaviors generated by the proposed control algorithm will be investigated in section 3.3. Furthermore, the frequency effect introduced in the previous chapter will be analyzed for the case of the simulated two-wheeled robot. A more mechanically complex robotic device and its behaviors generated under homeokinetic control will be described in section 3.4. The final experiment in this chapter (section 3.5) shows a prominent example of the emergence of sensorimotor coordination under homeokinetic control. A discussion of the results presented in this chapter will take place in section 3.6

3.1 The *lpzrobots* Software Package

The *lpzrobots* software package¹ is a framework for simplifying experiments with simulated and real robotic hardware. For example, the possibilities to observe controller or robot parameters online and log them into a file were build in as well as the option to change parameters online. The development of the *lpzrobots* software package was initiated by Georg Martius and the author. Further contributions were made by Ralf Der, René Liebscher, Marcel Kretschmann, Dominic Schneider, Claus Stadler and Frank Güttler.

We decided to conduct experiments in a physically realistic simulation for several reasons. One reason is that there is much more flexibility. We are able to run experiments with many different kinds of robots and with many different (or identical) robots at the same time. Due to the financial efforts and manpower needed, this would not be possible with real robotic hardware. Another point is that the simulations can run much faster than real-time, which gives us the possibility to run more experiments in a given time. Furthermore we can make a variety of changes in the environmental settings, some of which can be hard

¹The *lpzrobots* software package can be downloaded at ([Martius et al., 2008](#)).



Figure 3.1: Screenshots of different simulations realized with the *lpzrobots* software package.

to achieve in reality, such as decreasing the gravity. Screenshots of different simulations realized with the *lpzrobots* software package are shown in figure 3.1.

3.1.1 Modules of the Software Package

The *lpzrobots* software package is modularly organized and consists of three main parts which will be shortly described in the following. For detailed information visit or download the online documentation at [Martius et al. \(2008\)](#).

- The *selforg* package² comprises our controllers for self-organized behavior together with a small framework to ease their use.
- The *guilogger*³ is an application for observing parameters or simulation data in general, online as well as offline.
- The 3D robot simulator *ode_robots*³ is used for experiments in virtual reality.

3.1.1.1 The *selforg* Package

Different kinds of controllers as well as the controllers for the self-organization of behavior developed in our group are integrated in the *selforg* package. Furthermore, different wirings exist which provide the functionality of wires in real robotic hardware, namely connecting sensors and motors with the control structure. Additionally, the wirings allow for simple preprocessing, like the smoothing of sensor values or the adding of additional sensors which could for example provide a derivative of an existing sensor. The wirings are realized independently of the controller and hence can be used in combination with any of the controllers. The usage of the object-oriented programming approach allows extensions by

²The *selforg* Package is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License ([CC License, 2008](#)) for non-commercial use only.

³The *guilogger* and the *ode_robots* simulator are released under the GNU General Public License ([GNU GPL, 2008](#)) as free software.

new controllers or wirings which can easily be plugged in and used in an already existing experimental setup. The controllers can be used in experiments with real as well as with simulated hardware. This simplifies the transfer between simulated robots and their real existing counterparts.

3.1.1.2 The *guilogger*

The *guilogger* application simplifies the task of observing the experimental parameters, including controller and robot parameters, during (online modus) and after the experiment, when inspecting the data offline (offline modus). It is based on the *gnuplot* tool ([Merritt et al., 2008](#)) and allows multiple *gnuplot* windows (showing a diagram of the parameters over time) where the parameters to be shown can be switched on or off in the main window. A configuration file stores the selected parameters of the plot for the next experimental run. In the simulation environment the data is transferred via pipe to the *guilogger*. In the offline modus the data is directly read from file.

3.1.1.3 The Three-Dimensional Robot Simulator *ode robots*

ode robots is a three-dimensional physically realistic simulation environment. It is based on two libraries, one for the physics simulation and one for the visualization. The former is the Open Dynamics Engine (ODE), an open source, high performance library for simulating rigid body dynamics, developed by Russell Smith ([Smith, 2008](#)). ODE is a stable and platform independent library with an C/C++ application programming interface. It has advanced joint types and integrated collision detection with friction. It is used in many computer games, 3D authoring and simulation tools. For visualization we use the Open Scene Graph library, which is an open source 3D graphics toolkit for the development of high performance graphics applications ([Osfield et al., 2008](#)). Written entirely in Standard C++ and OpenGL it runs on a variety of platforms.

We created many robots, with different shapes, sensory modalities and actuators, and environmental setups including different obstacles. Further tools simplify the creation of new objects or robots and ease the experiments, like the possibility to change robot, controller or simulation parameters online or to record a movie of the experiment.

3.2 Homeokinetic Versus Static Controller

3.2.1 Goal

In this first experimental section the properties of the homeokinetic control should be compared with that of a static controller, where the controller parameters do not change over time. Furthermore the importance of the initial conditions for the behavior of the system should be investigated. A one-dimensional controller (equation 2.5) with the parameters synaptic strength c and bias h is considered for these investigations.

3.2.2 Setup

For both controllers the initial value of the bias h is varied in steps of 0.1 from -0.5 to 0.5 , the synaptic strength c in steps of 0.1 from -1.5 to 1.5 . In the case of the static controller these values are fixed until the end of the experiment. For the homeokinetic controller the parameter regulation as described in section 2.3 was applied. The parameter a of the world model required by the homeokinetic controller is initialized with small random values in the range $[-0.01 \dots 0.01]$ and updated by supervised learning in order to minimize the prediction error (equation 2.9), where the samples (x_{t+1}, y_t) are obtained online in each timestep.

The device under control is a simulated one-dimensional robot in the *lpzrobots* software package (see section 3.1), which is placed between two walls as depicted in figure 3.2. The sensor value is the measured wheel velocity. The robot is not equipped with bumpers or proximity sensors. Hence the only possibility to detect a collision is via the measured wheel velocity, due to the blocked wheels. The motor command is the desired wheel velocity of the robot. The distance to each wall is 7.6 length units. Hence the robot can only drive 7.6 length units until it hits a wall. Then the robot has to invert the velocity to be able to further drive.



Figure 3.2: For the simulation of a one-dimensional robot a two-dimensional simulated robot was used, where both wheels get the same motorcommand and the sensorvalue is the mean of the two measured wheel velocities. Hence the robot is able to drive only forward and backward. The robot is placed in the center between two walls, as indicated by the two bars.

3.2.3 Results

For every parameter configuration ten experiments were conducted, each lasting five minutes. In each time step the distance (absolute value) travelled by the robot was measured and summed up at the end of the experiment.

3.2.3.1 Static Controller

For the static controller the diagram in figure 3.3 shows the mean travelled distance over ten experiments. In a small region for values of $|h| < 0.2$ and the synaptic strength $c > 1$ the mean travelled distance reaches values of more than 200 length units. Hence the robot was able to cross the region between the walls up to 15 times. This is caused by the fact that with $c > 1$ and the response strength approximately 1 as given by the system, the overall feedback strength R in the loop is supercritical. Hence sensory noise can be amplified and leads to activity (motor command $y \neq 0$) of the system in the beginning of the experiment. Furthermore, the system will invert the velocity after a collision with the wall due to the

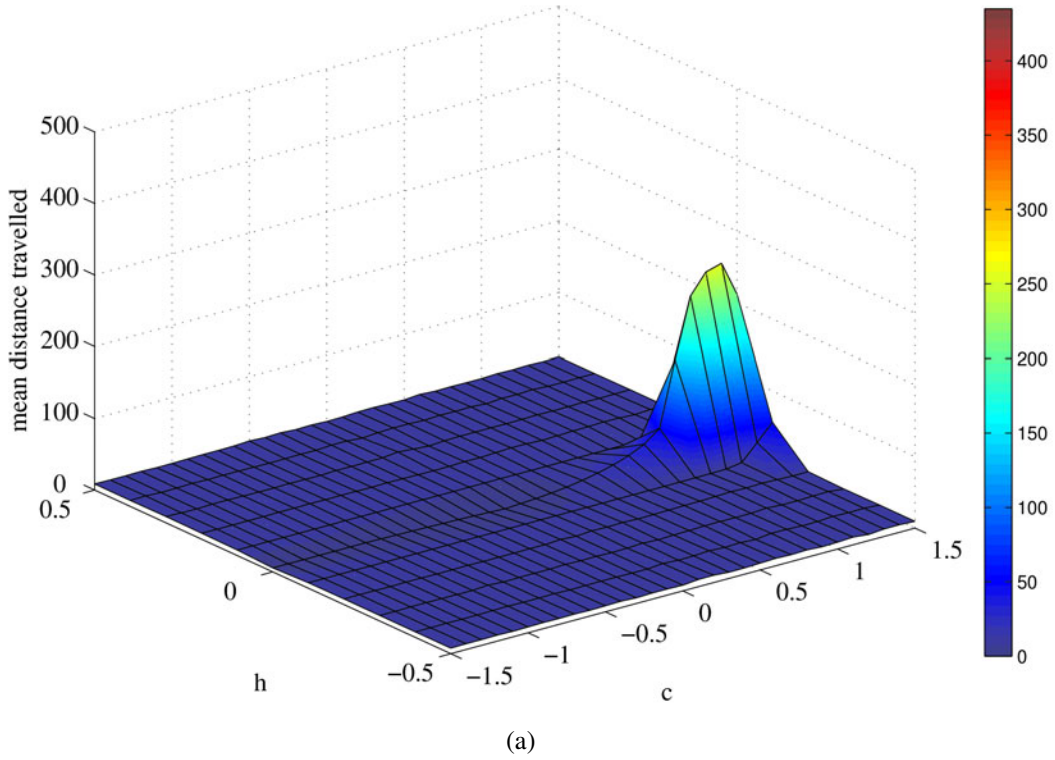


Figure 3.3: Mean distance travelled by a simulated one-dimensional robot over ten experiments for each of the 300 different parameter settings of the static controller. The value of the bias h was varied in steps of 0.1 from $-0.5..0.5$, the synaptic strength c in steps of 0.1 from $-1.5..1.5$. Each experiment lasts 5 minutes of simulated realtime. Only for a small amount of the initial parameter setting the static controller is able to generate movements, which do not end at the first wall contact with a travelled distance of about 7.6 length units. The standard deviation of the travelled distances is circa 30% of the mean travelled distance for all parameter settings.

noise-amplification effect as described in section 2.2.2.4. For values of $c < 1$ and hence $R < 1$ this effect does not take place and the system stays inactive.

For values of the bias h larger than a critical value h_c only one stable fixed point of the system remains, as described under the hysteresis effect (section 2.2.2.4). Based on h the system will generate a motor command $y \neq 0$ and drive either forward or backward, depending on the sign of h . When colliding with a wall it becomes obvious that this is a realization of an open loop control. Since the motor command only depends on h and the sensor values are ignored, it will not change after the collision. This way the system gets stuck at the first collision with a wall which happens after a travelled distance of about 7.6 length units. Only for $h = 0$ (and still $R < 1$) the system does not move at all.

3.2.3.2 Homeokinetic Controller

For the homeokinetic control, the mean travelled distance for the different initial settings of the parameters is shown in the diagram in figure 3.4. The initialization of the synaptic

parameter	mean	standard deviation
c	1.3110	0.0333
h	0.0028	0.0684
a	0.9694	0.0566
R	1.2709	0.0322

Table 3.1: Mean value and standard deviation of the parameters of the homeokinetic controller for all experiments with an initial value of $c > 0$. The mean feedback strength R is seen to be slightly supercritical, representing an active system. The mean value of the bias is very small, since it is only used for switching between the fixed points in dependence of the modeling error, but averages to zero in the mean.

strength with $c = 0$ would lead to a singularity of the objective function (equation 2.37). Thus c was initialized with small random values in the range $[-0.01 \dots 0.01]$ instead of 0. The parameter of the world model was initialized with small random values in the range $[-0.01 \dots 0.01]$ and learned online to minimize the prediction error of the internal model.

For initial values of $c > 0$ and independent on the initial value of h , the system showed a mean travelled distance of around 400 length units. Under this broad range of starting conditions the homeokinetic control is able to travel approximately twice as far as the static controller under the best conditions in the previously presented experiments. Considering the properties of the parameter dynamics in section 2.4 we already know that the synaptic strength will be increased to a value slightly above 1 (assuming the response strength $a \approx 1$). Hence the feedback strength in the loop R will be supercritical and activity (motor command $y \neq 0$) will be generated by amplification of sensory noise. The value of the bias h is kept small most of the time, but will cause a change between the fixed points of the systems if the modeling error is large. Hence the robot is able to change the direction of motion very fast after a collision and is this way able to travel nearly twice as far as with a static controller.

When c is initialized with a negative value the travelled distance is very small, so that the robot does not reach a wall. This is caused by the fact that with $c < 0$ and a (as a hardware constant) being learned to a positive value, the feedback strength in the loop R and hence the Jacobian L of the system are negative. This means the robot tries to invert the target wheel velocity in each time step and is not able to move away. This can not be overcome by the system, since the objective function $E \sim \frac{1}{L^2}$ has a singularity at $c = 0$. So in general we should take care to initialize the systems so that the eigenvalues of the Jacobian are not negative.

The mean value of the synaptic strength c , the bias h and the response strength a in the last time step of all experiments with an initial value of $c > 0$ are shown in table 3.1. In agreement with the theory we find a slightly supercritical mean feedback strength of $R = 1.27$ and a bias which is very small in the mean but allows to rapidly switch between the fixed points if the modeling error is large, e.g. when hitting a wall. This way the robot equipped with the homeokinetic control spent less time at the wall and is therefore able to cover larger distances than the robot with the static controller.

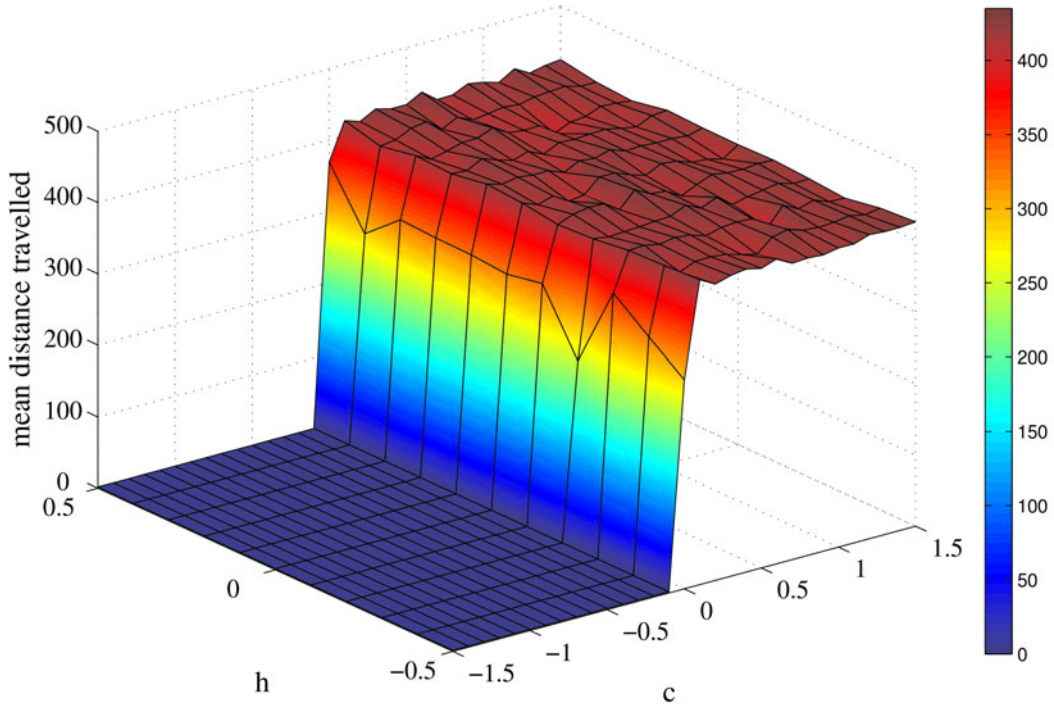


Figure 3.4: Mean distance travelled by a simulated one-dimensional robot over ten experiments for different initial conditions with a homeokinetic controller. The initial value of the bias h was varied in steps of 0.1 from $-0.5..0.5$, the synaptic strength c in steps of 0.1 from $-1.5..1.5$. Each experiments lasts 5 minutes of simulated realtime. For initial values of the synaptic strength $c > 0$ the homeokinetic controller is able to drive a distance of more than 400 length units, which is nearly twice as much as the maximum distance with a static controller. For $c < 0$ (and $a > 0$ as given by the realization of the system) the feedback strength in the loop is negative. Hence the robot tries to invert the target wheel velocity in each time step and is not able to reach the wall. The standard deviation of the travelled distance is nearly 50% of the mean value if $c \leq 0.01$ and below 20% otherwise.

3.2.4 Summary

In this first and simple experiment we have shown that the initial conditions for the proposed control system are less important, except for the requirement that the eigenvalues of the Jacobian must not be negative.

The parameter dynamics is found to reproduce in the mean, the expected slightly super-critical feedback strength which generates activity in the system due to noise amplification. Furthermore the bias is small in the mean, but allows due to the frequency effect a rapid change of the actual fixed point, and hence of the actual behavior, after a collision with the wall (see section 2.4). Equipped with this parameter dynamics the system is able to travel nearly twice as far as the static controller with the best parameter setup in the same time.

3.3 Environment Exploration with a Simulated Two Wheeled Robot

3.3.1 Goal

We will consider now the closed-loop control problem for the velocity of a simulated two wheeled robot to investigate the activity and explorative character of the system. Furthermore the role of the modeling error for the transitions between different behaviors will be analyzed. In a second experiment with a group of robots it will be studied if the active and explorative character of the system remains in the case of heavy perturbations.

3.3.2 Setup

A first group of experiments was conducted with a robot with length 1, which is placed in a circular arena with diameter 13 length units (see figure 3.5). The *lpzrobots* package (see section 3.1) was used for the physical simulation of the robotic hardware.

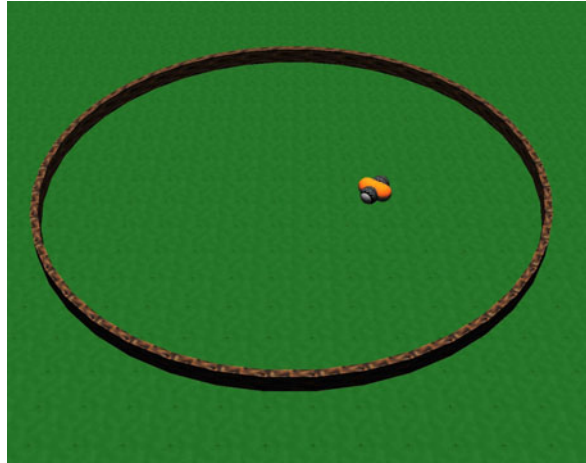


Figure 3.5: Experiments are performed with a simulated two-wheeled robot in a circular arena. The robot is equipped with wheel counters. The length of the robotic body is 1 and the diameter of the arena is 13 length units.

The actual velocity of the wheels of the robot is measured by wheel counters, i.e. by proprioceptive sensors. Hence the vector of sensor values is $x = (\nu_1, \nu_2)^T$ with ν_1 and ν_2 being the wheel velocities of the left and right wheel, respectively. The robot is equipped with two motors controlling the wheels independently. The vector of motor commands is $y = (y_1, y_2)^T$, with y_i being the target velocity of wheel i . The homeokinetic controller is given as a function $K : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ mapping the sensor values x_t at time step t to the motor commands y_t at the same time step.

$$y_t = K(x_t)$$

So the controller for this robot consists only of two neurons, each is connected to the two

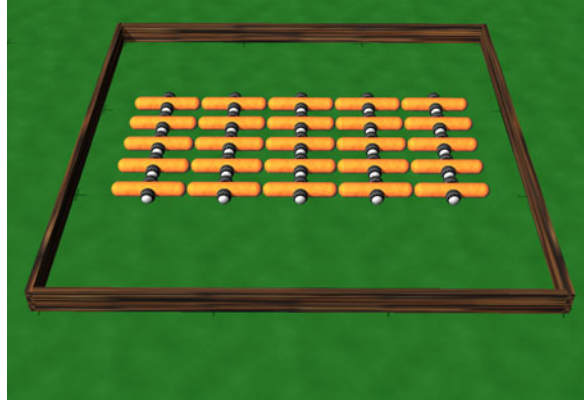


Figure 3.6: Experimental setup with 25 simulated robots with length 2.5 in a square arena.

sensory inputs. Parameters of the controller are therefore the synaptic strength c_{ij} with $i = 1..2, j = 1..2$ and the two bias values h_1, h_2 .

The internal model F is a linear mapping $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ of the motor commands y_t to the new sensor values x_{t+1} of the next timestep, with ξ_t being the modeling error.

$$x_{t+1} = F(y_t) + \xi_t$$

This mapping is realized by the parameters a_{ij} with $i = 1..2, j = 1..2$, weighting the model inputs. The parameters of controller and world model are initialized with small random values, it is only checked, that the eigenvalues of the Jacobian L of the loop function ψ have positive values (compare section 3.2.3). Different experimental runs were conducted lasting one hour and twenty-four hours of simulated realtime, each ten times.

A second group of experiments was conducted with two-wheeled robots with a longer body. The length of the robot is then 2.5 length units. The longer body makes the environment exploration more difficult, it especially increases the possibility to get stuck when a lot of these robots are used in one arena. In the experiment we used 25 robots inside a square arena with a side length of 14.5 length units (see 3.6). All robots are controlled by the homeokinetic approach. Sensors, actuators and controller are as described above. With these experiments we want to investigate the robustness of the proposed algorithm in case of heavy perturbations by other active agents, by a comparison with an experiment with only a single robot in the same arena. The experiments lasted twenty-four hours of simulated realtime and were conducted ten times.

3.3.3 Results

3.3.3.1 Single Robot

A trajectory of the robot in an experiment lasting one hour of simulated real time is plotted in figure 3.7. The figure shows curved and straight paths, and areas of on-site rotation representing the possible behaviors of the robot: straight driving, curved driving, and rotation on the place. By visual judgment of the figure it can be seen that most of the area inside the

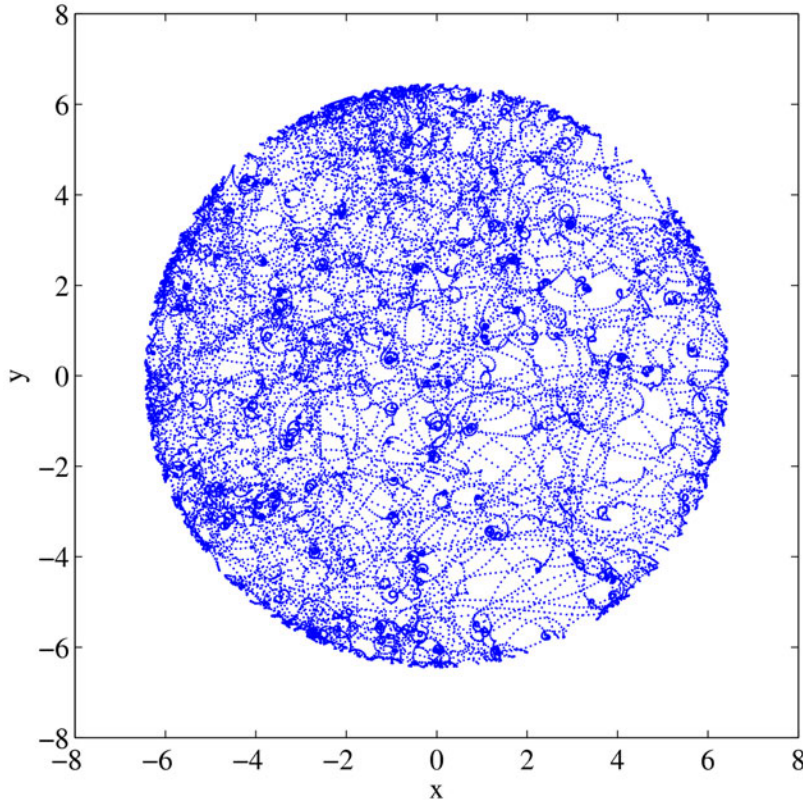


Figure 3.7: Trajectory of the robot in a one-hour experiment in a circular arena using homeokinetic control. The robot's trajectory has covered most areas of the arena. Due to the frequency effect the robot spent more time at the walls than in the inner obstacle-free areas.

arena is already covered. In the inner part of the arena one can see that movements persist for longer periods, hence larger distances are covered here. Near the walls of the arena there are accumulations of robot positions, because the behavior changed more often and only small areas are covered with one and the same movement. This effect is caused by the frequency effect (section 2.4.3) which changes the actual behavior depending upon the modeling error ξ of the internal model. When moving in the inner part of the arena, the behaviors are rather predictable, the modeling error is small and hence the behaviors are kept for quite some time. When colliding with a wall the modeling error is large and the behavior changes rapidly (see figure 3.8 for a detailed plot of a part of the robots trajectory and the modeling error near the wall). Since the robot can not, with only having wheel counters as sensors, infer which behavior leads away from the wall, it just tries different behaviors until it can escape. Thereby the robot spends a relatively long time at the wall. This way the world model is provided with further information to possibly improve the predicting abilities in this region of space. However, due to the limitations introduced by the linear world model used here it is not possible to improve prediction at the wall without reducing the prediction quality in free space.

Figure 3.9 shows the mean absolute value of the modeling error ξ for different distances between the robot and the center of the circular arena for the long run (an one-hour and a twenty-four-hours experiment), clearly stating that the area near the wall causes larger

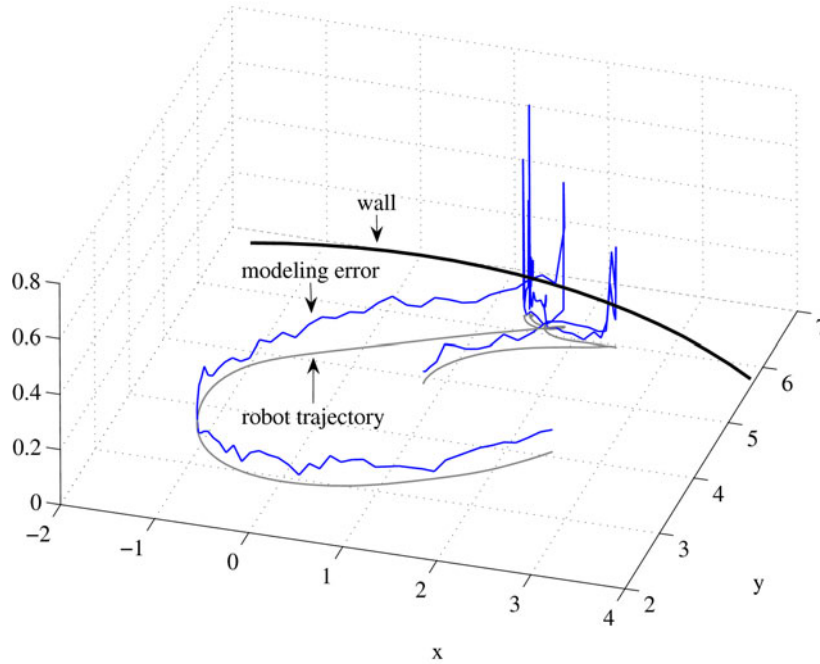


Figure 3.8: A part of a robot's trajectory (center of the robot) and the absolute value of the modeling error near the wall, selected from an one-hour experiment. It can be seen that the robot smoothly changes its behavior in the inner obstacle-free part of the arena, beginning at position (2.5, 4). After finishing the turn the robot drives straight on to the wall, and keeps this motion until it hits the wall. The robot then rapidly changes the behavior until a behavior is found which leads away from the wall. The fast change of the behavior near the wall is caused by the large modeling error in collision situations. Due to the frequency effect (section 2.4.3) and as the case may be the effect of the driving term in the update rule (section 2.4.1), the behavior of the system changes. The absolute value of the modeling error ξ at the respective positions is plotted above the trajectory of the robot. It clearly shows larger modeling errors in collision situations than in free space.

modeling errors as compared to the other locations in this setup. The internal drive of the robot to explore these areas is reflected by the increased probability of staying near the wall, as shown by the histograms of the robot's distance from the center of the arena for a one-hour as well as for a twenty-four-hours experiment, see figure 3.10.

To get an idea about the explorative character of the homeokinetic control the circular arena was divided in 10 000 parts (places), each with an area of about 0.01 square length units. Figure 3.11(a) shows the number of visited places over time. Already after three hours the center of the robot (from which the trajectory was recorded) had visited nearly all places.

In figure 3.11(b) the traveled distance of the robot is shown for the one-hour and the twenty-four-hours experiment. Regions of inactivity, characterized by a slope of zero, are essentially absent. This shows that the homeokinetic principle generates autonomous self-organizing systems which are active without a reference value, desired behavior or goal given from outside.

The parameter dynamics (equation 2.41) which runs concomitantly with the state dy-

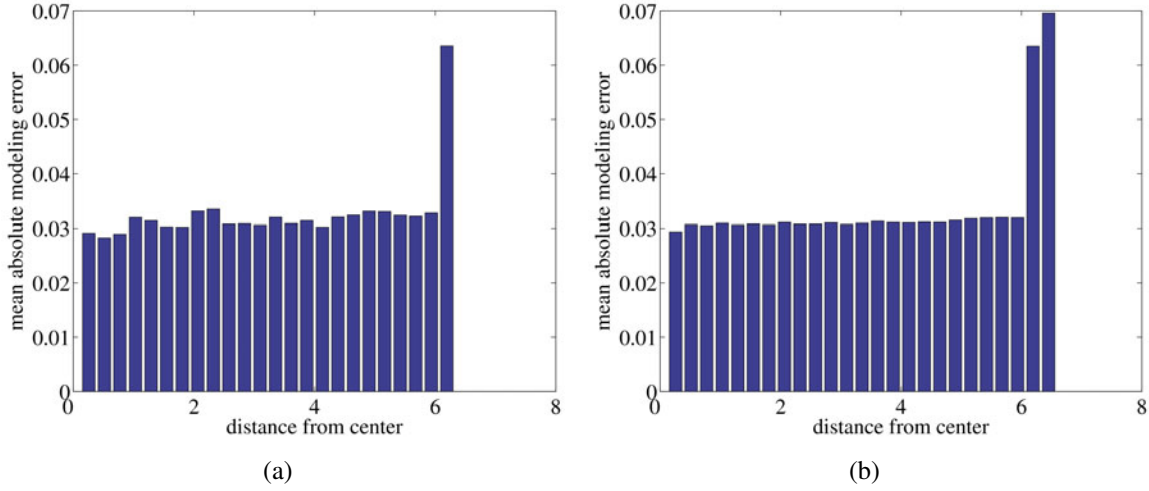


Figure 3.9: Mean absolute value of the modeling error ξ for different distances between robot position and center of spherical circular arena. A distance value of zero means the robot is in the center of the arena, a value of 6.5 means hitting the wall. As it turns out the mean absolute modeling error in the inner part of the arena is around 0.03 while near the wall it is around 0.06. This holds true for averaging over (a) one hour and (b) over one day (twenty-four hours) simulated real time.

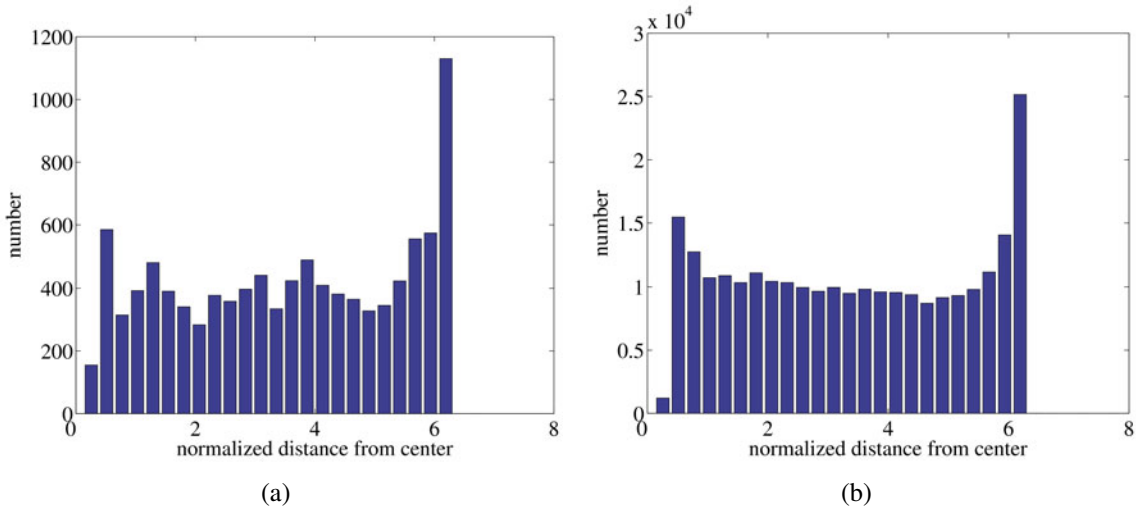


Figure 3.10: Histogram of the robot's distance from the center of the circular arena normalized by the respective areas for (a) an one-hour and (b) a twenty-four-hours experiment. It shows that in short and long term experiments the probability for the robot to stay in the inner region of the arena is smaller than to stay somewhere near the wall. This is caused by the frequency effect, which changes behaviors more rapidly when the modeling error is high, e.g. in collision situations. Therefore the robot is not able to cover large distances and the probability to stay near the wall is increased. In the inner region of the arena the modeling error is smaller, hence behaviors remain for quite some time and the robot is able to cover larger distances.

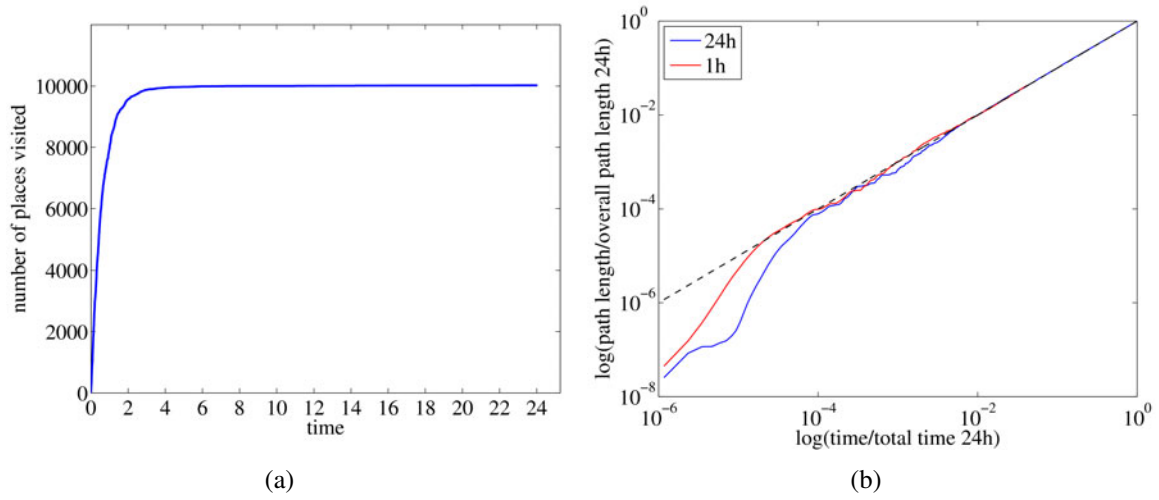


Figure 3.11: (a) Number of visited places over time (only the first visit of a place is counted) during an experiment lasting twenty-four hours of simulated realtime. The circular arena was divided in 10000 parts (places) each comprising about 0.01 square length units. After three hours nearly all of the places were visited by (the center of) the robot. (b) Log-log plot of the cumulative distance traveled by the robot over time using the homeokinetic controller. The monotony of the cumulative distance indicates that the robot is active during the whole time of the experiment. This holds true for the one-hour as well as for longer, e.g. a one-day (twenty-four hours) experiment. For longer times the cumulative distance is a linear function of the time.

namics (equation 2.26), is seen to work as intended in the presented system. During the experiments all parameters stayed at values expected from theory (section 2.5). The driving term in the learning rule leads to an increase of the controller parameters c and hence generates activity in the system. The anti-Hebbian term on the other hand, dampens the increase of the controller parameters, and thereby establishes slightly supercritical feedback strength with mean values of approximately 1.14 in the sensorimotor loop, as can be obtained from table 3.2, showing the mean values and standard deviations of the controller and world model parameters during the twenty-four-hours experiment. The cross-channel response strength ($a[0, 1]$, $a[1, 0]$) is nearly zero while for $a[0, 0]$ and $a[1, 1]$ it is around one as one would expect from a system showing different behaviors, thereby counteracting deprivation of the world model. Der and Martius (2006) showed that the approach explicitly avoids deprivation by producing purposive actions in a natural way. The controller parameters $c[0, 0]$ and $c[1, 1]$ show mean values of 1.14, realizing a slightly supercritical feedback strength. The contribution of the synaptic strength for the cross-channels can be neglected in the mean, but the values of the standard deviation indicate that the system tries to integrate these sensors, and would do so if they would show predictable (in terms of the internal model) responses. The mean values of the biases are very small. Hence the bias values are used to change the actual behavior of the system, via the frequency effect but on average balance to zero.

parameter	mean	standard deviation	parameter	mean	standard deviation
$c[0, 0]$	1.1438	0.0485	$a[0, 0]$	1.0008	0.0146
$c[0, 1]$	0.0069	0.1523	$a[0, 1]$	-0.0053	0.0172
$c[1, 0]$	-0.0115	0.1515	$a[1, 0]$	-0.0029	0.0173
$c[1, 1]$	1.1461	0.0508	$a[1, 1]$	1.0012	0.0155
$h[0]$	-0.0016	0.0576	$R[0]$	1.1462	0.0505
$h[1]$	-0.0012	0.0571	$R[1]$	1.1489	0.0513

Table 3.2: Mean value and standard deviation of the controller and the model parameters, and the feedback strength in the loop R of a twenty-four-hours experiment. The parameter dynamics realize in the mean slightly supercritical feedback strength, which indicate activity in the system. The mean values of the biases are seen to be very small, since the biases are only used to switch between the different behaviors of the system in dependence of the modeling error.

3.3.3.2 Group of Robots

For the experiments with 25 simulated robots, pictures of the experimental run are shown in figure 3.12. As can be obtained from the images the arena is rather crowded, so the robots do not have much free space to explore without hitting one another. That is the reason why the cumulative distance travelled by the robot is lower than compared to a single robot in the same arena without obstacles (see figure 3.13). However, there are no regions of inactivity (slope = 0) showing that the robots do not get stuck and are still exploring the arena even though the exploration takes more time in the crowded arena (see figure 3.14). Hence the behaviors of the robots which were exposed to a lot of perturbations in the crowded arena show the same active and explorative character than the single robot. The mean feedback strength over ten twenty-four-hours experiments is seen to be slightly supercritical (see table 3.3). The diagonal elements of the synaptic strength c are seen to be larger as compared to table 3.2, since they are compensating the smaller response strength $a[0, 0]$ and $a[1, 1]$. The smaller mean values of $a[0, 0]$ and $a[1, 1]$, and the larger mean values of $a[1, 0]$ and $a[0, 1]$, are the result of the “disturbed” samples (x_{t+1}, y_t used for model learning, which do not show the proper relation between motor commands and sensor values, as in the case of the single robot. The larger standard deviation of the biases compared to table 3.2 indicates that the bias was more often used to change the actual behavior, but also shows a very small mean value. For videos of this experiment see Der et al. (2008).

3.3.4 Summary

We have presented an experiment with the homeokinetic principle controlling a realistically simulated two-wheeled robot in a circular obstacle-free arena. The robot was active during the whole experiment lasting for one or twenty-four hours of simulation time. The presented diagrams show that it covered the whole area of the arena without longer periods of inactivity. Different behaviors of the robot appeared and the transition between these behaviors is shown to depend on the modeling error via the frequency effect, as described in section 2.4.3. The driving (confer section 2.4.1) and the anti-Hebbian term (section 2.4.2) in the update rule led

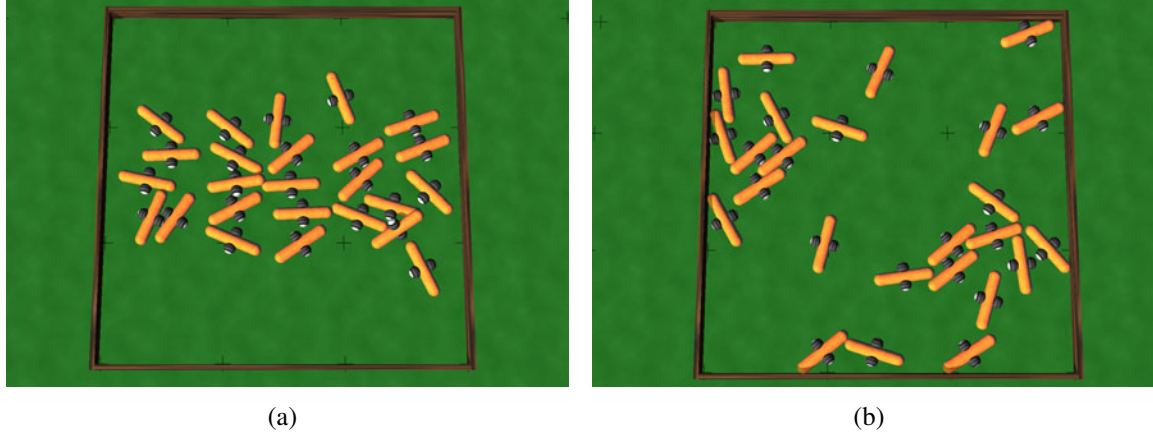


Figure 3.12: Pictures of 25 robots during a twenty-four-hours experiment. (a) After a few steps the order of the initial setup disappears. (b) After half an hour of simulation. Even though the robots build clusters they do not get stuck. The clusters disentangle after a while and new clusters appear at other places.

parameter	mean	standard deviation	parameter	mean	standard deviation
$c[0, 0]$	1.3727	0.1824	$a[0, 0]$	0.7622	0.1503
$c[0, 1]$	-0.1258	0.4973	$a[0, 1]$	0.0901	0.1078
$c[1, 0]$	-0.0833	0.4809	$a[1, 0]$	0.0882	0.1074
$c[1, 1]$	1.3686	0.1961	$a[1, 1]$	0.7640	0.1482
$h[0]$	-0.0012	0.1267	$R[0]$	1.0372	0.1665
$h[1]$	-7.5110e-04	0.126	$R[1]$	1.0425	0.1561

Table 3.3: Mean value and standard deviation of the controller and the model parameters, and of the feedback strength in the loop R of the ten twenty-four-hours experiments. The parameter dynamics realizes in the mean a slightly lower but still supercritical feedback strength than in the obstacle free arena (compare table 3.2). The high number of perturbations by the obstacles is reflected in the lower response of the sensors to the motor commands ($a[0, 0]$ and $a[1, 1]$ smaller). Therefore the synaptic strength $c[0, 0]$ and $c[1, 1]$ are larger to keep the feedback strength supercritical. The mean values of the biases are seen to be very small, since the biases are only used to switch between the different behaviors of the system in dependence of the modeling error. That this happens more often in the overcrowded than in the free arena is seen in the higher values of the standard deviations of h as compared to table 3.2.

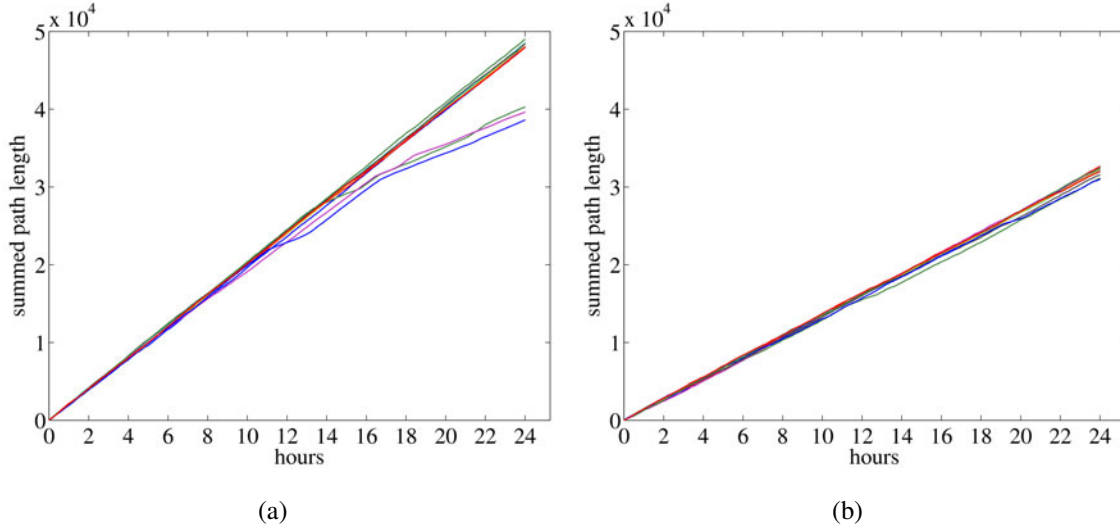


Figure 3.13: Cumulative distances traveled by the robot over time using the homeokinetic controller. The monotony of the cumulative distances indicates that the robot is active during the whole time of the experiment. This holds true for (a) a single robot in the arena as well as (b) the central robot (in the initial position) in the arena with 24 other robots. Both experiments lasted twenty-four hours of simulated realtime and were conducted ten times.

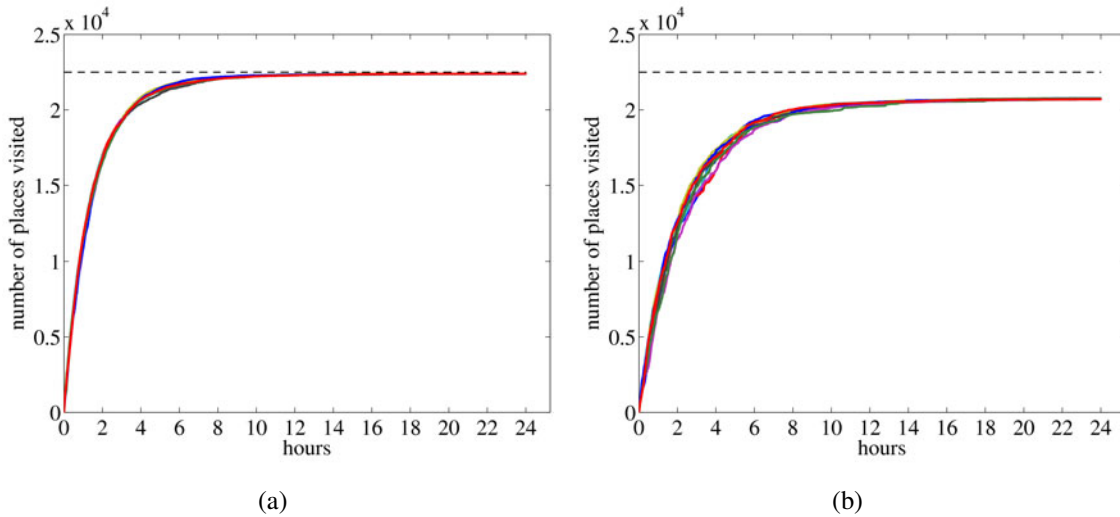


Figure 3.14: Number of visited places over time during 10 experiments, each lasting twenty-four hours of simulated realtime, for (a) a single robot in the arena, (b) the central of 25 robots (regarding the initial position) in the arena. The square arena was divided in 22500 parts (places) each comprising about 0.01 square length units. The robot is able to explore a wide range of the arena under heavy perturbations by the other robots (b) even though not as many places were visited as in the case of the single robot (a).

to desirable parameter values as shown in table 3.2.

In an experimental setup with a lot of robots and hence dynamic obstacles and heavy perturbations, the presented algorithm shows its robustness by the robot exploring the environment while not getting stuck in the crowds.

3.4 Simulated Snakelike Robot

3.4.1 Goal

In this experimental section a robot with more degrees of freedom will be equipped with a homeokinetic controller. This way it can be tested if the realization of the homeokinetic paradigm is able to deal with more complex systems and show different body- and environment-related behaviors, which are active as well as explorative.

3.4.2 Setup

We study a snakelike robot (results are published in Der et al. (2006)), again simulated in the *lpzrobots* simulation package (see section 3.1). The structure of the robot is sketched in figure 3.15. It consists of beam segments connected by hinge joints. In the experiment a robot consisting of six beams, each of them with a length of 1 length unit, was used.

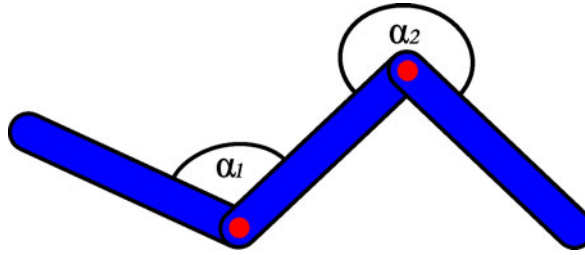


Figure 3.15: A sketch of a snake with two joints. Sensor values sent to the controller are the angular velocities of the joints, the controller outputs are the desired angular velocities.

The vector of sensor values $x = (x_1, x_2, x_3, x_4, x_5)^T$ consists of the measured angular velocities of the five hinge joints. The robot is equipped with five motors controlling the motion of the hinge joints independently. The motor values $y = (y_1, y_2, y_3, y_4, y_5)^T$ specify the target angular velocity of the hinge joints. The homeokinetic controller is given as a function $K : \mathbb{R}^5 \rightarrow \mathbb{R}^5$ mapping the sensor values x_t at time step t to the motor commands y_t at the same time step.

$$y_t = K(x_t)$$

So the controller for this robot consists of five neurons, each having five inputs. Parameters of the controller are therefore the synaptic strength c_{ij} with $i = 1..5, j = 1..5$ and five bias values h_1, \dots, h_5 .

The model F is a linear mapping $F : \mathbb{R}^5 \rightarrow \mathbb{R}^5$ of the motor commands y_t to the new sensor values x_{t+1} of the next timestep, with ξ being the vector of modeling errors.

$$x_{t+1} = F(y_t) + \xi$$

This mapping is realized by the parameters a_{ij} with $i = 1..5, j = 1..5$, weighting the model inputs.

We initialize the matrix a of model parameters as a diagonal matrix with the a_{ii} chosen such that the response of the joints is already coarsely modelled. The matrix c of controller parameters is also chosen diagonal but with very small random values for the c_{ii} (eigenvalues of the Jacobian L of the loop function ψ are checked to be positive, see section 3.2.3) so that in the beginning the joints execute fluctuating motions only. Hence the system starts with 5 decoupled feedback loops due to this diagonal initialization. Each experiment starts with the robot in a compact pose as depicted in the screenshot in figure 3.16(a).

3.4.3 Results

The homeokinetic control of the snake-like robot generates many different forms of behaviors, some of them reminding the observer of different crawling motions of biological agents or call up the impression of purposeful movements like a jumping behavior, see figures 3.16(b), 3.16(c).

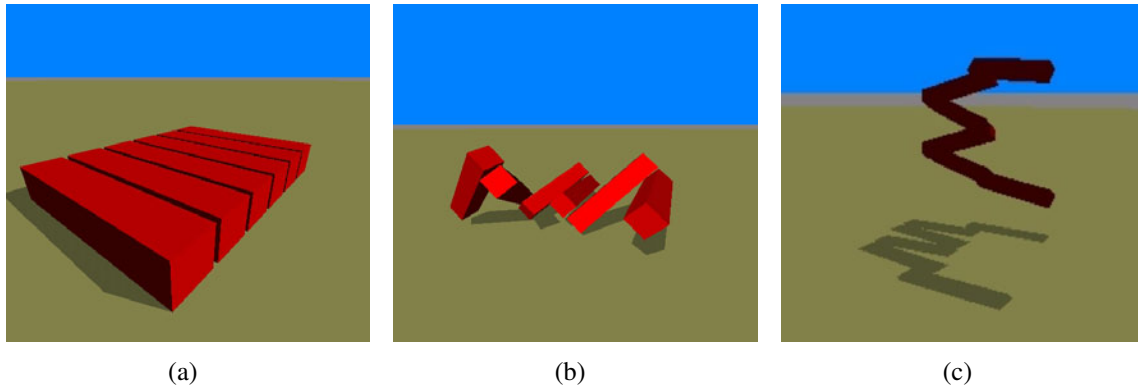


Figure 3.16: Screenshots of snakes on a plane. (a) Shows a snake in the initial position. (b) Crawling and (c) jumping behavior of the snake-like robot as observed during the experiments.

As already known from the theoretical chapter (chapter 2), all behaviors are emerging from the interaction of the parameter and state dynamics of the system. The development of the model and controller parameters associated with the neuron controlling joint 1 is shown in figure 3.17 for the initial phase of the experiment. The diagram 3.17(a) shows at the left-hand side the unit matrix initialization of the model parameters. In the initial phase this relation is preserved by the learning dynamics because sensor 1 shows the best correlation with the motorcommands for joint 1. Diagram 3.17(b) shows a plot of the synaptic strength and the threshold of the neuron generating the motorcommand for joint 1 in the initial phase of the experiment. The diagonal element c_{11} of the synaptic strength is initialized with a very small random value but the learning dynamics rapidly increases the value due to the driving term as described in section 2.4.1. The parameter dynamics rapidly increases all the diagonal elements of c so that the feedback strength in each of the loops increases. This

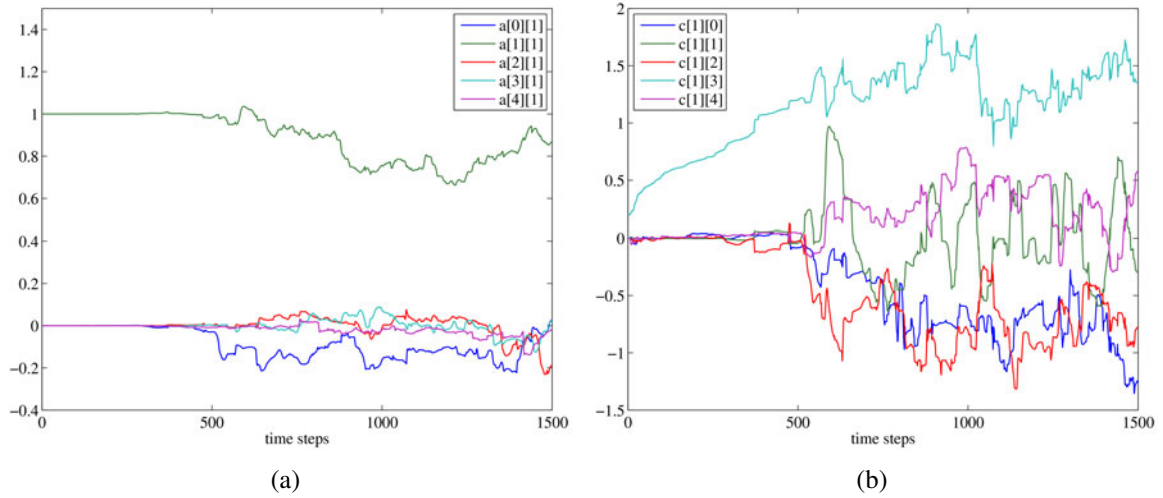


Figure 3.17: Development of the parameters a_{i1} and c_{1j} associated with the neuron controlling joint 1 in the initial phase of an experiment. (a): The world model matrix a is initialized as the unit matrix reflecting the independence of the joints. The learning dynamics preserves this in the initial phase. (b): The diagonal elements c_{ii} are initialized with very small random values and increase until the supercritical feedback strength is reached and the system starts to move (at about time 500). The development of the nondiagonal elements reflects the integration of contributions of the other segments. However, the self coupling c_{ii} is seen to stay dominant (top line).

increase stops if a supercritical feedback strength of the loop is reached and activity in the robotic device is generated.

In this regime the nondiagonal elements of the matrix of synaptic strength are also seen to develop and integrate the contributions of other sensors, so that the dynamics of the joints are coupled. This is caused by the fact that the reactions of the joints to the applied forces are correlated due to collision, inertia, and friction effects, as reflected in the increasing model parameters of these channels on the right-hand side of diagram 3.17(a). However the self-coupling remains dominant since the response strength (model parameter a) in this channel is dominant.

Figure 3.18 shows the development of the model and controller parameters of joint 2 and 3 during a longer period of the experiment. All parameters change during the whole time of the experiment, thereby making up the different behaviors exhibited by the agent. The controller parameters (3.18(b) and 3.18(d)) are substantially changing over time but stay in a certain range, so that the controller neurons remain in a sensitive working regime. This is the result of the two counteracting terms in the update rule: the driving term, which generates the activity in the system by increasing the controller parameters c (section 2.4.1), and the anti-Hebbian term, which demands predictive behaviors of the system by realizing a slightly supercritical feedback strength in the sensorimotor loop (section 2.4.2).

The model parameters a_{ij} describing the observed angular velocity at joint i as the response of the motor action applied to joint j are shown in the diagrams 3.18(a) and 3.18(c). In the diagram 3.18(a) we see that the response strength of the self-coupling stays dominant,

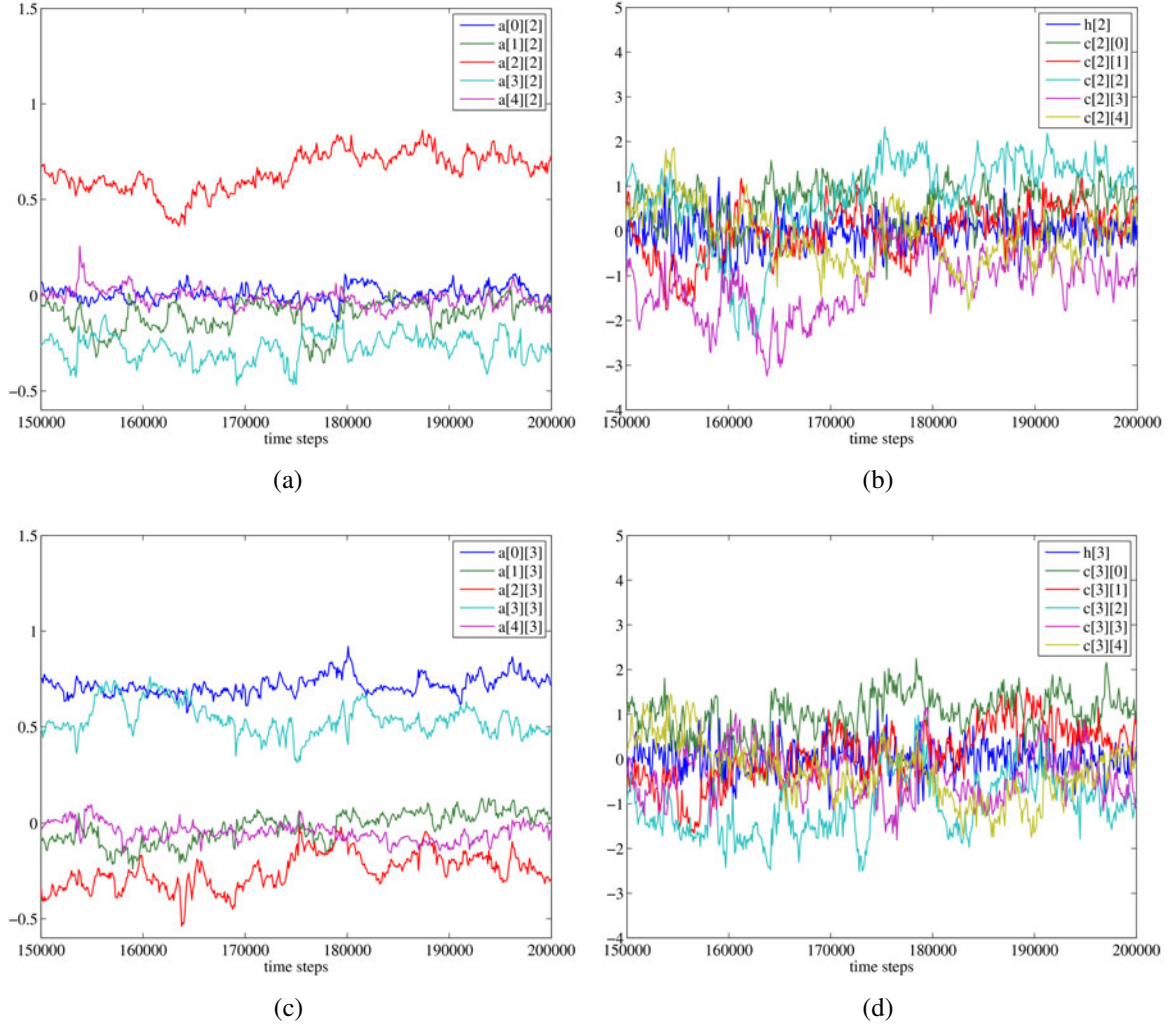


Figure 3.18: Controller and model parameters for joints 2 (top row) and 3 (bottom row) during time step 150 000 to 200 000. (every 100th value plotted). Left: Model parameters; Right: Controller parameters; In accordance with the sensitization paradigm, the controller parameters are substantially changing over time but stay in a certain range, so that the neurons remain in a sensitive working regime. The model parameters a_{ij} describe the observed angular velocity at joint i as the response of the motor action applied to joint j . One would expect a diagonal matrix a (only self-couplings show a large response strength), however some non-diagonal elements are non-zero, reflecting the correlations between different joints, for instance $a[0][3]$ in the lower left diagram.

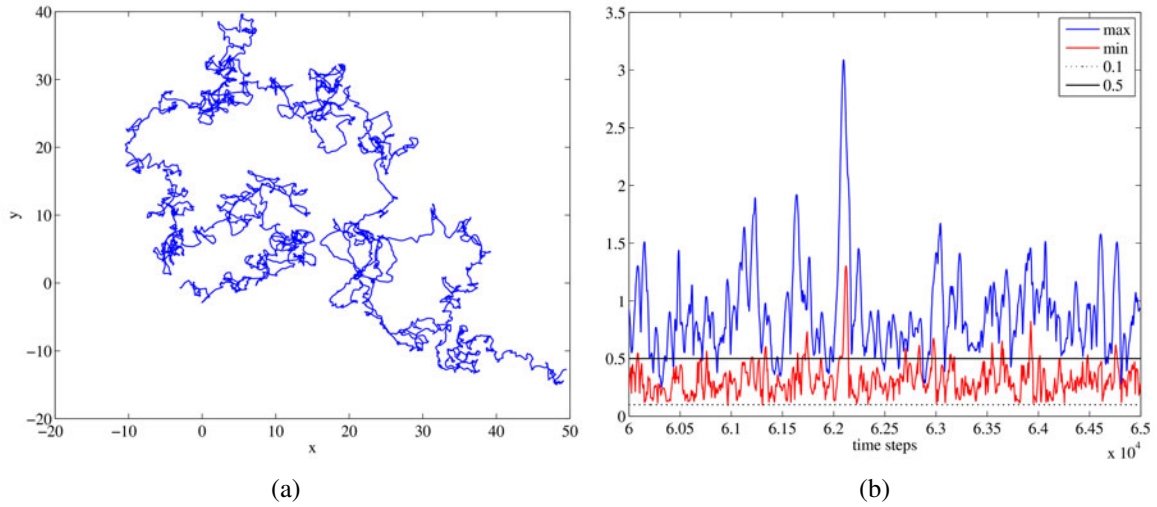


Figure 3.19: Motion of a snakelike robot with 5 joints (6 segments with length 1) during the experiment. Left: Position of the snakes center projected on the plane over 165 000 time steps with starting point at $(-3,0)$. Right: Altitudes of the centres of the highest (max) and the lowest (min) segment from time step 60 000 to 65 000. Segments laying completely on the ground have an altitude of 0.1, standing upright have an altitude of 0.5. One can see that the snake sits up and even jumps so that it exceeds the altitude of 0.5 with the lowest segment.

as expected for independently controlled joints. However, some non-diagonal elements (representing the cross-channel response strength) are non-zero, reflecting the coherence in the motion of the joints, for instance a_{03} in diagram 3.18(c).

The emerging dynamics of the robotic system is quite complex and rather difficult to analyse. Here we want to indicate the degree of organization of the motion by measuring the motion of the center of the snake projected on the plane, see figure 3.19(a). It can be observed that in the beginning the center is more or less stationary (in a time average picture), but after some time the snake covers increasingly larger regions of space. Apart from that, the altitudes of the snake segments also provide information about the type of behavior. For analytical purposes we consider the center of the highest and lowest segment over time. The difference between both can be interpreted as a measure for the current posture. Jumping behavior is characterized by an altitude > 0.5 of the lowest segment. As shown in figure 3.19(b) the snake sits up frequently and occasionally performs jumps. The height of an element laying completely on the ground is 0.1, as indicated by the dotted line in the diagram. Most of the time the lowest segment is above that height, reflecting the active motion of all segments of the body. Note, that even on long time scales qualitative changes in the parameters are observed (figure 3.18), indicating a rich behavior diversity. This is also seen directly when watching the snake over a long time. For videos see [Der et al. \(2008\)](#).

3.4.4 Summary

In this section an experiment with a five degree of freedom snake-like robot controlled by the homeokinetic principle was presented. The parameter dynamics generated activity in

the robotic device by increasing the feedback strength in the sensorimotor loop (due to the driving term in the update rule) as depicted by the diagrams of the parameters in the initial phase of the experiment. It was shown that the development of the parameters goes on during the whole experiment, reflecting the response strength of the sensors in the current environmental situation and thereby generating a rich behavioral diversity. These different behaviors, some of which call up the impression of crawling and jumping motions, lead to an exploration of the environment as well as a generation of behavioral primitives adequate for the robotic device used.

3.5 Emerging Sensorimotor Coordination

3.5.1 Introduction

The following experiment investigates the question of the emergence of sensorimotor coordination in complex robotic systems (as partly presented in [Der et al. \(2005\)](#)). The presented approach to self-organization will be applied to a simulated robot with two active and many passive degrees of freedom (which will be described in detail in section [3.5.2.1](#)).

The robot is underactuated in the sense that its actuators are not powerful enough to set the device (especially the passive parts of the body) in motion if the physical properties of the body are neglected. Thus a whole-body motion can only be excited by exploiting the physical properties of the body.

Hence a controller not provided with information about the physics of the body from outside has to become sensitive to the sensor values (which of course show the effects of the body properties) in order to generate active behaviors of the whole body.

The controller used here is a realization of the homeokinetic principle as introduced in section [2.3](#). Due to the same number of actuators and sensors it is identical to the one applied to the two wheeled robot, where each of the wheels is controlled independently (see section [3.3](#)). The internal model and the controller will be randomly initialized. The only body specific parameters κ are used for the scaling of the sensor and motor values to the appropriate ranges before entering or after leaving the controller (see paragraph [3.5.2.2](#)). No further knowledge about the complex physical behavior of the robotic body is available to the controller. The generation of a whole-body motion in the described system by such a controller would then be a prominent example of the emergence of sensorimotor coordination.

Considering the mentioned problems immanent in the setup of the experiment, it is well suited to investigate the parameter dynamics of the introduced control approach, namely the effects of the driving (see section [2.4.1](#)) and the anti-Hebbian term (see section [2.4.2](#)), the frequency effect (see section [2.4.3](#)) and in general the ability of the algorithm to generate active behavior in a device attached to the controller. But this setup also shows the limitations in the present realization of the approach which is basically seen in the restriction of the complexity of internal model (see section [2.2.1](#)).

In the following section the mechanical configuration of the robot and the setup of the controller will be described before having a closer look at the experimental run.

3.5.2 Setup

3.5.2.1 Mechanical Configuration of the Robot

The robotic device considered here is a snake-like artifact called *skidding snake*. It is simulated in the *lpzrobots* software package (see section 3.1).

The *skidding snake* consists of 10 spheres. The head element (red sphere) is connected to a tail of 9 spheres which are connected to each other by a link (frictionless ball-and-socket joint) so that each of the spheres can rotate passively like in a string of beads (see figure 3.20(b)).

The red sphere is the actuated head of the *skidding snake* to which a force vector

$$f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (3.1)$$

is applied. The force vector is parallel to the ground (lying in the plane). All other parts of the body are passive. The forces available are not enough to enable the head element to pull the tail. Hence the robot is underactuated in the sense that it can generate movements of all (active as well as passive) body parts only by exploiting the properties of the body. In the concrete case a whole-body motion can only be generated if rolling lateral motions are used.

The robotic device provides feedback from two exteroceptive sensors. This is the velocity vector of the head element in the plane with

$$w = \begin{pmatrix} w_x \\ w_y \end{pmatrix}. \quad (3.2)$$

The components of force vector f and velocity vector w are aligned to the x - and y -axis of the global coordinate system, see figure 3.20(b).

The coefficient of friction of the spheres on the ground is set to $\mu = 0.3$. This is a reasonable value if compared to $\mu = 0.01$ for skis on snow or $\mu = 0.5$ for two hands rubbing together. In this constellation of friction and maximal force available the head element is not able to pull the tail. Hence a collective motion essentially requires the passive spheres to roll.

In the real world such a robotic device could be realized by substituting the head element with an omnidrive robot and concatenating the passive spheres of the tail to it. For more information on omnidrive robots see the work of Rojas (2005), Rojas and Förster (2006), Kuppuswamy et al. (2006), Wilson et al. (2001).

3.5.2.2 Controller

The controller consists of a control and a model unit, as explained in section 2.3.3.1. The control unit is a neural network which is self-regulating according to the general paradigm (section 2.5). The network consists of two leaky-integrator neurons. Hence the network provides a two dimensional output vector according to the requirements of the robotic device. The model unit is also realized as a neural network consisting of 2 neurons (identity output function, no thresholds), using the control unit output y to predict the sensor values x of the next timestep.

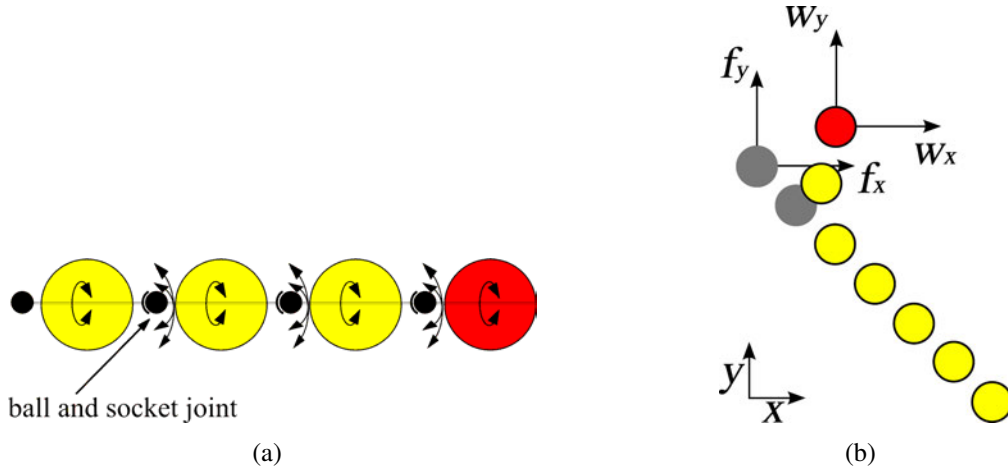


Figure 3.20: (a) Mechanical construction of the skidding snake. Ball-and-socket joints between the spheres are drawn. Arrows indicate the possible rotations: in every direction at the joint and around the axis between joints. (b) Sketch of skidding snake in the global coordinate system. After execution of a time step in which a force vector f was applied to the head element the velocity vector w is read back as feedback of the system. The components of the force vector f and the velocity vector w are aligned with the axis of the global coordinate system.

As sensory input x to the controller the velocity vector w of the head element in the plane is used:

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \kappa_{in} \begin{pmatrix} w_x \\ w_y \end{pmatrix} \quad (3.3)$$

The velocity values are scaled with κ_{in} to the range $(-1, 1)$ before entering the network. This scaling is necessary to allow the adjustment of the ratio of the synaptic strengths of the controller neurons according to the ratio of the response strength in the channels, especially when using different sensory modalities. Furthermore it allows easier handling when employing diverse robotic hardware.

The output vector y determines the force vector f applied to the head element of the “skidding snake” with

$$f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \kappa_{out} y = \kappa_{out} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad (3.4)$$

where κ_{out} is a constant of proportionality. Since the network consists of neurons with a hyperbolic tangent activation function, the output of the network is in the range $[-1, 1]$. To exploit the available forces in the range $[-3, 3]$ this scaling is required.

3.5.2.3 Experimental Setup

During the experiments the “skidding snake” is placed on the ground of a simulated world without borders or obstacles. Hence there are no perturbations by the environment.

In the experiments the model and controller parameters are randomly initialized in the range $[-0.1, 0.1]$, the eigenvalues of the Jacobian L of the loop function ψ are checked to be positive (see section 3.2.3).

With such an initialization the system is set far below the bifurcation point of the Hopf bifurcation (figure 2.15), since the feedback strength $R_i = \sum_{j=0}^1 c_{ij}a_{ji}$ with $i = 0, 1$ in each of the two channels is much smaller than the critical value 1.

3.5.3 Results

The description of the experimental run is divided into three paragraphs, one for each of the three phases shown in the course of the experiment (compare figure 3.21). The first paragraph will describe how the controller with the tabula rasa initialization adapts to the environment and to the body in which it is embedded. The controller has no prior knowledge about the body or its properties, only the number of sensors and motors is known. The second and third paragraphs describe the emergence and decay of a whole-body motion.

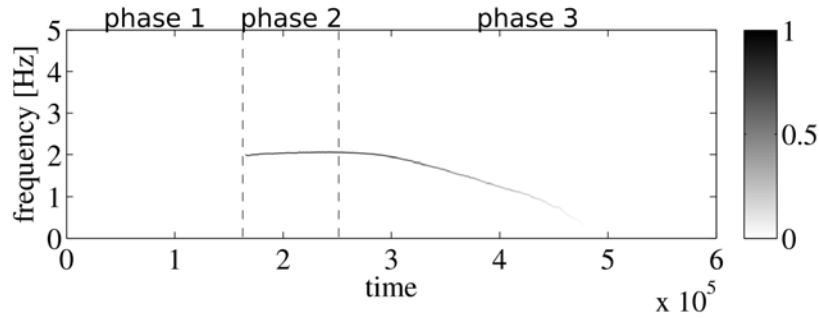


Figure 3.21: Power spectra of the sensor value x_1 (measured velocity) during the experiment. Each column represents the lower frequency part of the discrete Fourier transform of a 5 second sliding window of the sensor values. Every 0.1 seconds a power spectrum is plotted. Bright pixels correspond to low energy and dark to high energy in the corresponding frequency band. High energy in a certain frequency band means that changes between positive and negative velocity in the sensor/motor value occur at about that frequency. The energy is normalized by the maximal energy in the power spectrum. In the beginning of the experiment (phase 1) no favorite frequency of behavior can be identified in the system. The start of the rotational mode of behavior is indicated by the appearance of the dark line showing the frequency of rotation (around step 165000). During the rotational mode (phase 2) this frequency stays almost constant. The energy of the frequency of rotation is much larger as compared to the frequency components before this motion sets in. Hence the diagram shows only bright dots before and after the rotational mode. In phase 3 the frequency of rotation decreases, indicating the end of the rotational mode of behavior. Around step 480000 the dominating frequency disappears. Now only small lateral random motions occur. The frequency components of these motion are very small as compared to the frequency of rotation during the skidding behavior.

3.5.3.1 Adaptation to body and world

In the beginning of the experiment the controller and internal model parameters are randomly initialized in the range $[-0.1, 0.1]$ (see figure 3.22). Hence the feedback strength in each of the two channels has a maximal possible value of 0.02 ($R_i = c_{i0}a_{0i} + c_{i1}a_{1i}$, see left hand side of figure 3.23(b)). This leads to a strong suppression in the sensorimotor loop.

Since the “skidding snake” is placed on the ground without any initial motion or force applied, the sensor values are zero except for some sensor noise (see left hand side of figure 3.24(a)). Consequently a correlation between the sensor and motor values is missing. This is reflected in the the model parameters a which stay around their initial values, see left part in diagram in figure 3.22(a).

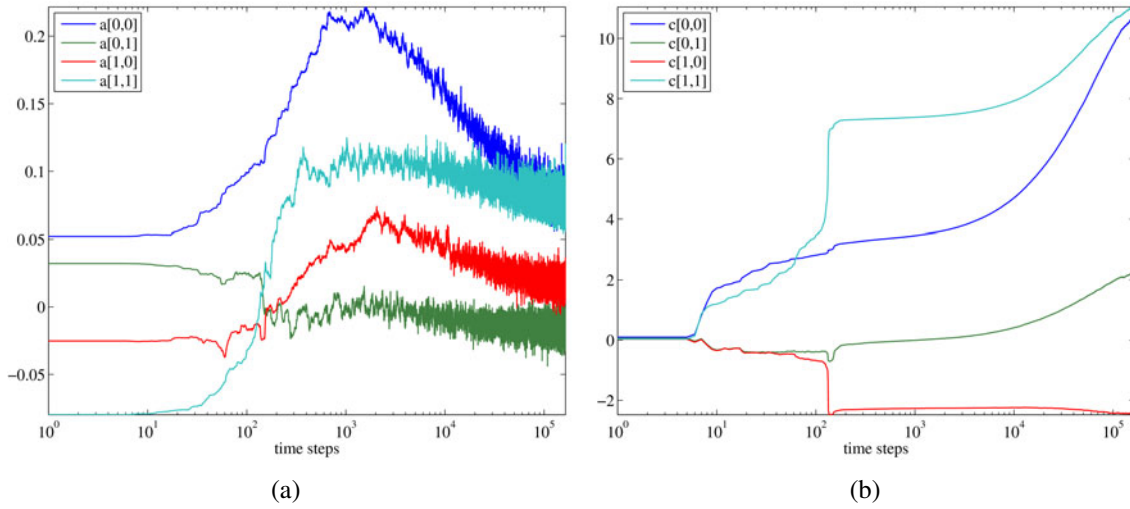


Figure 3.22: Model and controller parameters (without threshold h) from the beginning of the experiment until the skidding snake enters the rotational mode of behavior (step 0 to 160000). A logarithmic time axis is used to provide a detailed view of the parameters in the very beginning of the experiment. (a) The model parameters in-channel (a_{00} , a_{11}) and cross-channel (a_{10} , a_{01}) response strength show only weak correlations between sensor and motor values, while in-channel correlations are stronger. (b) The synaptic strengths of the neurons controlling the output y are seen to increase (due to driving term in the update rule) in order to generate activity in the system. The diagonal elements of c show a steeper rise since channels with higher response strength ($|a_{ij}|$) are favored.

In this situation the parameter dynamics comes into play. The small values of x and z cause a domination of the driving term μa_i in Δc_{ij} (compare equation 2.41, section 2.5.1). This leads to an increase of the matrix elements of c , with channels of higher response strength ($|a_{ij}|$) being favored, according to equation 2.43. Figure 3.22(b) shows the development of the controller parameters in the beginning of the experiment (the first 200000 steps).

This increase of the synaptic weights leads to increasing motor values and small movements of the head element, but only weak reactions of the passive tail. Hence the maximal amplitudes of the sensor values stay around an absolute values of 0.1. Figure 3.24 shows the

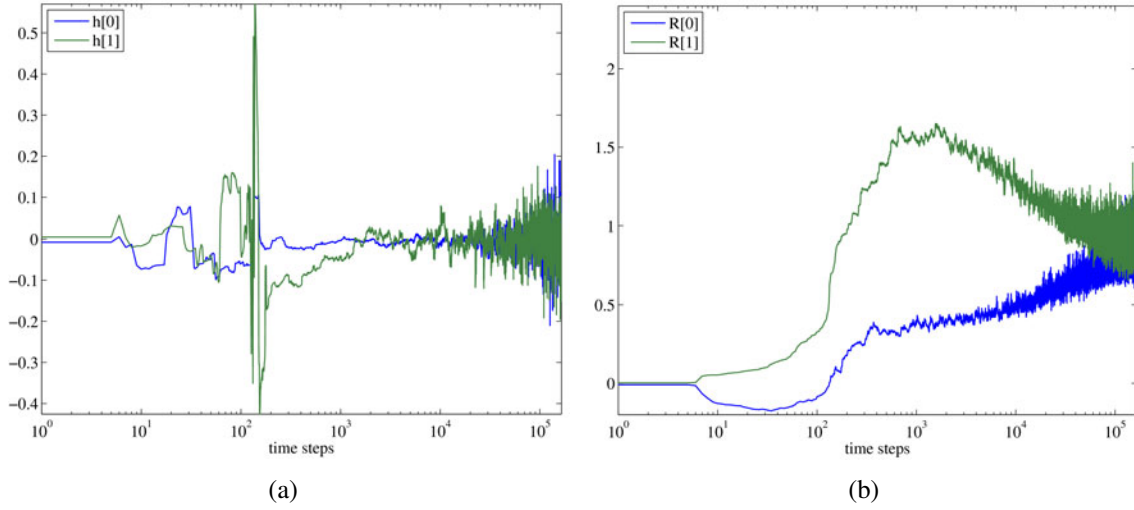


Figure 3.23: Biases (adaptive thresholds) and feedback strength R of the controller neurons from the beginning of the experiment until the skidding snake enters the rotational mode of behavior (step 0 to 160000). The thresholds (a) oscillate around zero and show increased values only around step 140 when also a steep rise in the synaptic strength occurs. The system is seen to increase the feedback strength (b) in both loops to supercritical values. In this regime it is possible to generate activity and hence motions by noise amplification.

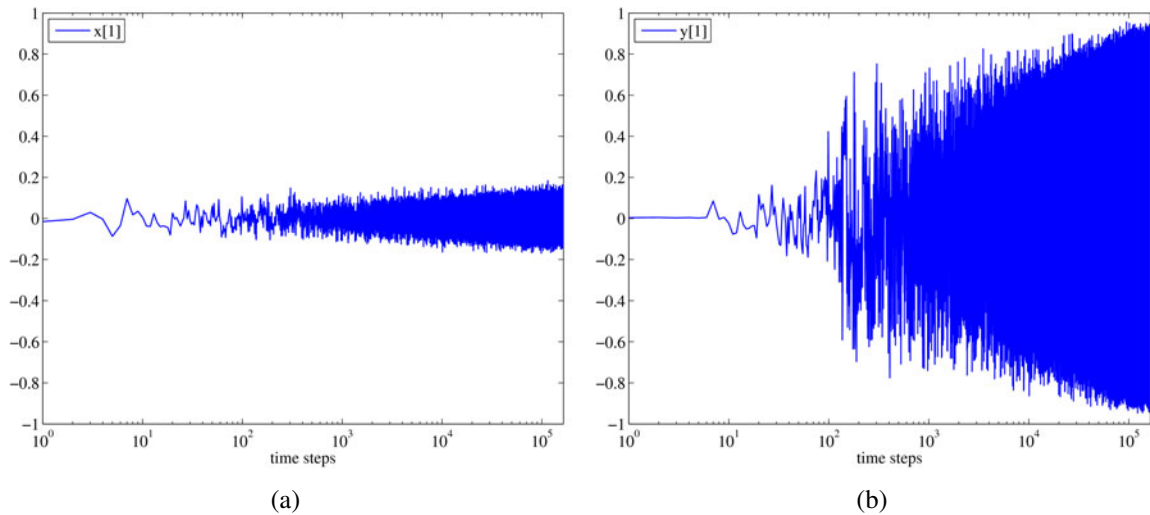


Figure 3.24: Sensor and motor values of channel 1 from the beginning of the experiment until the skidding snake enters the rotational mode of behavior (step 0 to 160000). A logarithmic x -axis is used to provide a detailed view of the values in the very beginning of the experiment. (a) The sensor value x_1 is mainly determined by the sensor noise and fluctuates around zero. (b) The motor value y_1 shows an increasing amplitude. This is caused by the driving term in the update rule of the controller parameters and will lead to the excitation of a rotational mode of behavior.

small sensor values compared to the increasing motor values for one of the channels.

Beginning around step 200 the model parameters in figure 3.22(a) show that the correlation between motor and corresponding sensor (the in-channel response strength, represented by the diagonal elements of matrix a) is slightly stronger than between a motor and the sensor of the other motor (the cross-channel response strength, represented by the non-diagonal elements of matrix a). But in general the values remain small, reflecting the fact that mainly the noise of the sensor values is amplified and the body shows only weak reactions to these random signals, seen as a swaying of the body.

The peaks in the amplitude of bias h_1 around time step 140 (compare figure 3.23(a)) indicate a large modeling error probably caused by the underactuatedness of the system. Since the modeling error also has an effect on the learning rate of the synaptic strength, c_{01} and c_{11} show a rapid change around step 140.

A frequency plot of the sensor value x_1 is shown in figure 3.21 but no favorite frequency of behavior can be identified in the first phase of the experiment.

The feedback strength R in each of the channels is seen to increase and reach super critical values in the right part of figure 3.23(b), as expected from the driving term in the update rule.

3.5.3.2 The rotational mode of behavior

With the feedback strength (respectively the synaptic strength) and hence the sensitivity increasing more and more the controller starts to respond to the swaying of the body. Around step 165000 the controller manages to amplify these swayings. Thereby a rolling motion of all, active and passive, elements of the body is generated. This motion is characterized by a rotation of all spheres around their axis (connection between the joints), where the first half of the spheres rotate in the opposite direction than the other half. Hence a collective rotational mode of behavior of the underactuated many degree of freedom robotic device was excited by the skidding motion. While the whole snake rotates around its center, it becomes stiff like a stick due to centrifugal forces and gyro effects.

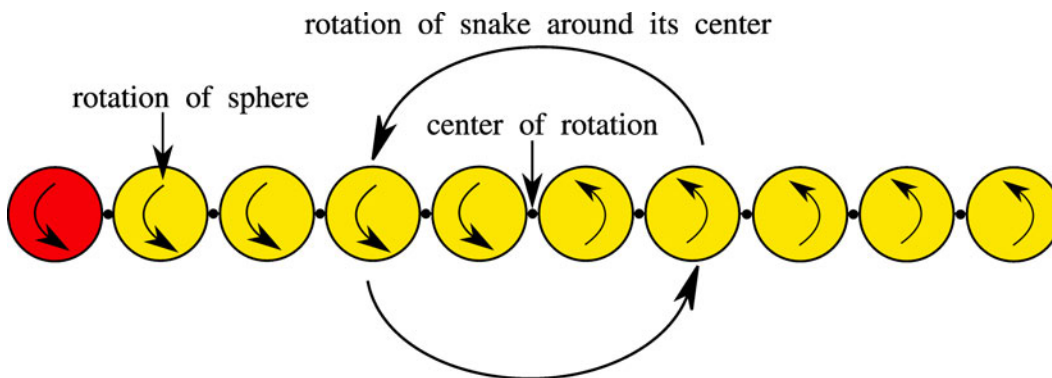


Figure 3.25: The rotational mode of behavior is characterized by a rotation of all spheres around their axis (connection between the joints), where the first half of the spheres rotate in the opposite direction than the other half. Hence the whole snake rotates around its center, at which point it becomes stiff like a stick.

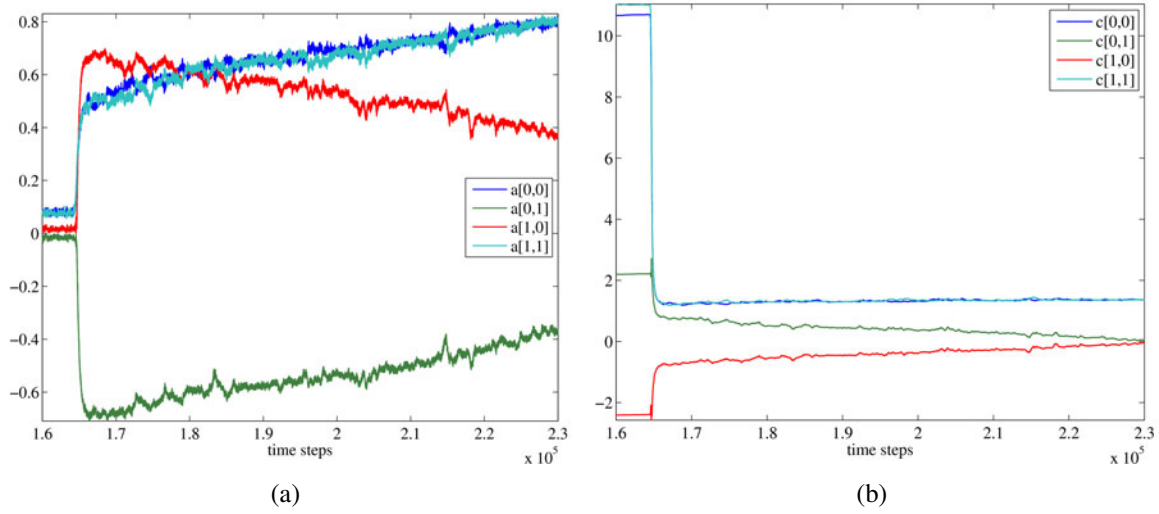


Figure 3.26: Parameter values during the rotational mode of behavior (steps 160000 to 230000). (a) Model parameters in-channel (diagonal elements of a) and cross-channel (non-diagonal elements of a) response strength. With the onset of the rotation around step 165000 the model parameters increase rapidly, representing the fact that the sensors now give a reasonable response to the motor commands. Later on the in-channel response strength is seen to increase while the cross-channel response strength decreases. (b) The synaptic strength of the controller neurons. Due to the increased response strength after the emergence of the rotational behavior the anti-Hebbian term of the learning rule (equation 2.41) comes into play and the strength of the synaptic weights is reduced. So the system regulates itself back to a slightly supercritical response strength. The relation between diagonal and non-diagonal elements of c is determined by the proportionality to the internal model (see section 2.5.2).

The emergence of the rotational mode of behavior generates a large modeling error due to the different response of the body (and hence the sensors) to the motor commands as before. This can be seen in the peaks of the threshold amplitudes (figure 3.27(a)).

The behavior is self-supporting since the emerging collective motions respond to the controller in a much more systematic way. This can be seen in the model parameters (figure 3.26(a), beginning with step 165000) which quickly adapt to the new situation since the systematic part in the generated behavior can now be modeled.

Due to the large increase of the model parameters the feedback strength also increases largely around step 165000, as depicted in figure 3.27(b). Through the large sensor values x (figure 3.28 shows the envelopes of the sensor and motor values) and hence large membrane potentials z the anti-Hebbian term in equation 2.41 comes into play and the synaptic strengths decrease very fast, see left part of figure 3.26(b). Thereby the slightly supercritical feedback strength is re-established as can be obtained from figure 3.27(b), beginning around step 166000. The feedback strength is seen to remain slightly supercritical during the whole mode of rotational behavior.

In the course of time the non-diagonal elements of a decrease while the diagonal elements further increase. This reflects the assumption that there is basically a correlation in and not across the channels, as one would expect for independent channels.

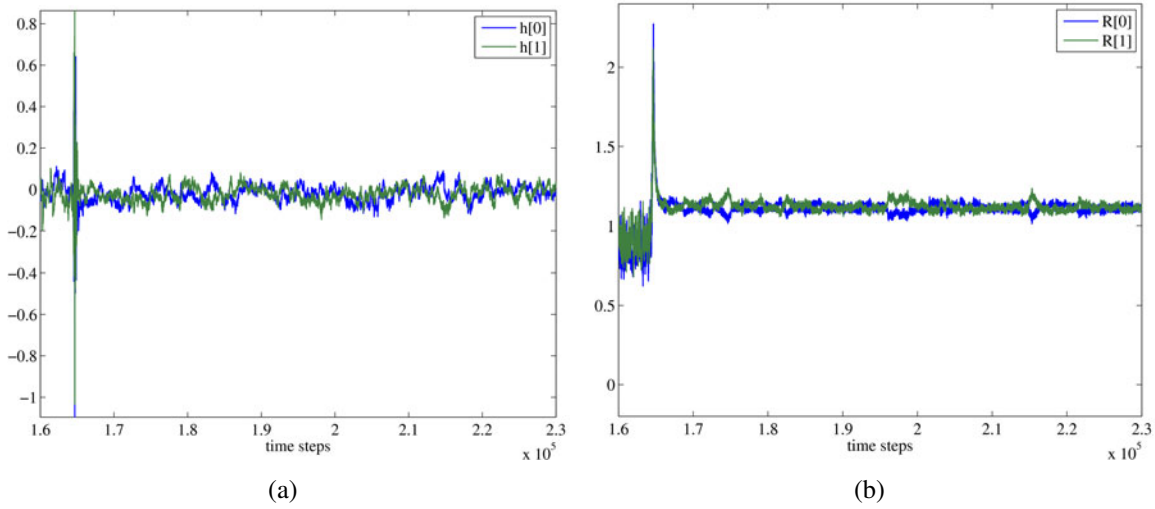


Figure 3.27: Biases (adaptive thresholds) and feedback strength of the control unit during the rotational mode of behavior (steps 160000 to 230000). Except around step 165000 the thresholds (a) oscillate with small amplitudes around zero. Due to the emergence of the rotational behavior around step 165000 the sensor values immediately reach large amplitudes which can not be predicted by the model. The resulting model error leads to large values of h since the amount of change of the thresholds is proportional to the modeling error. While the rotational mode of behavior sets in, the feedback strengths (b) increase rapidly. Due to the nonlinearity in the system an unlimited increase of the motor values is confined. The anti-Hebbian term of the update rule (see section 2.4.2) regulates the feedback strengths down to a slightly super-critical value. This value is retained during the whole mode of rotational behavior.

According to the proportionality between model parameter and synaptic strength of a channel (equation 2.43) the non-diagonal elements of c also decrease, (see figure 3.26(b)).

Despite of this changes in c and a , the feedback strength in each channel is seen to stay at a slightly supercritical value around 1.1, as can be obtained from the right hand side of figure 3.27(b).

The diagram in figure 3.29 show the sensory and motor space, where transitions to temporary stable limit cycles represent the rotational mode of the *skidding snake*.

Figure 3.21 shows the power spectrum of the sensor value x_1 during the experiment. It can clearly be seen that around time step 1650000 a main frequency appears which is the frequency of rotation of the *skidding snake*. After a short time of increase the frequency of rotation stays almost constant.

3.5.3.3 The end of the rotational mode of behavior

As described above the controller was able to generate motions of all elements of the *skidding snake* which is seen at first as meandering and eventually leads to a rotational mode of the body. In this rotational mode each of the spheres is rapidly rotating. Due to the mass of the spheres this results in strong inertia which keeps the snake in motion. However these

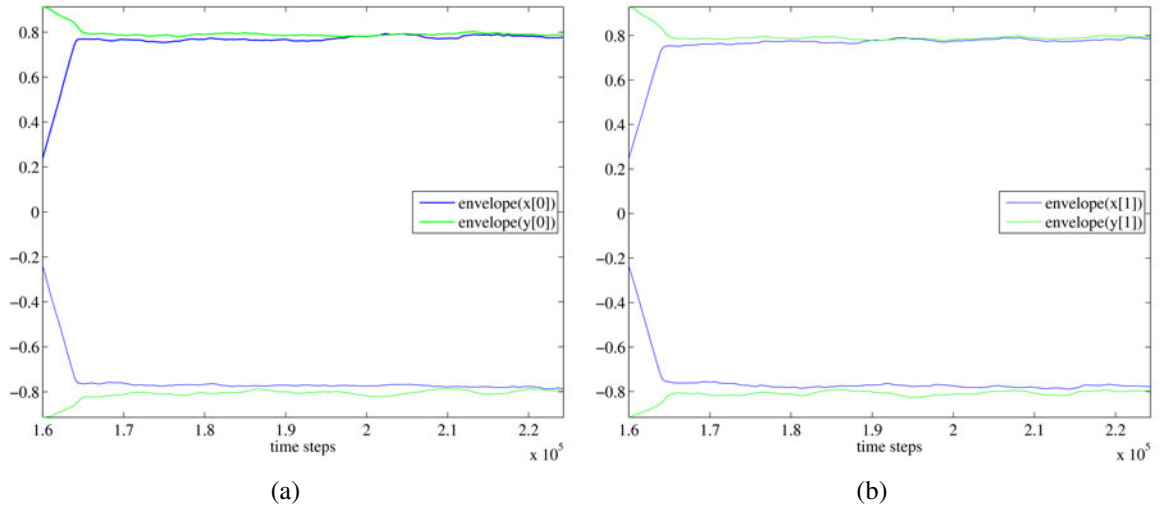


Figure 3.28: Envelopes of the sensor and motor values of (a) channel zero and (b) channel one during the rotational mode of behavior (steps 160000 to 230000). The figures show the envelope of the signal, since due to the fast oscillations also in a signal plot only the envelopes are identifiable. In the left part of the figures the emergence of the rotational mode of behavior can be seen. The sensor values x increase since the body shows a reasonable response to the motor commands. This leads to a situation of higher activity in the sensorimotor loop. The algorithm reacts by reducing the synaptic strengths of the controller neurons and consequently the motor values decrease. During the rotational mode the amplitude of the sensor and motor values remains almost constant at 0.8.

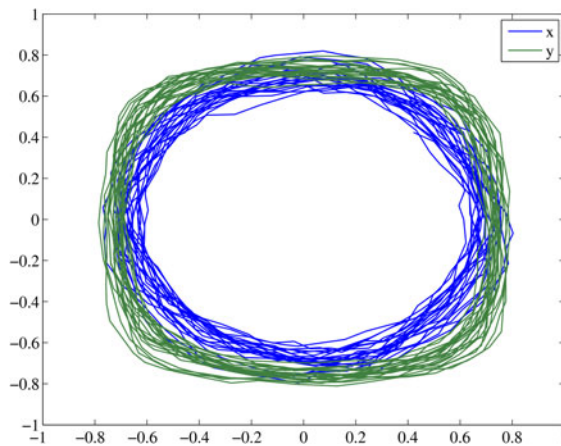


Figure 3.29: Motor space (blue line) and sensor space (green line) show the temporary stable rotational mode of the skidding snake (1000 time steps plotted).

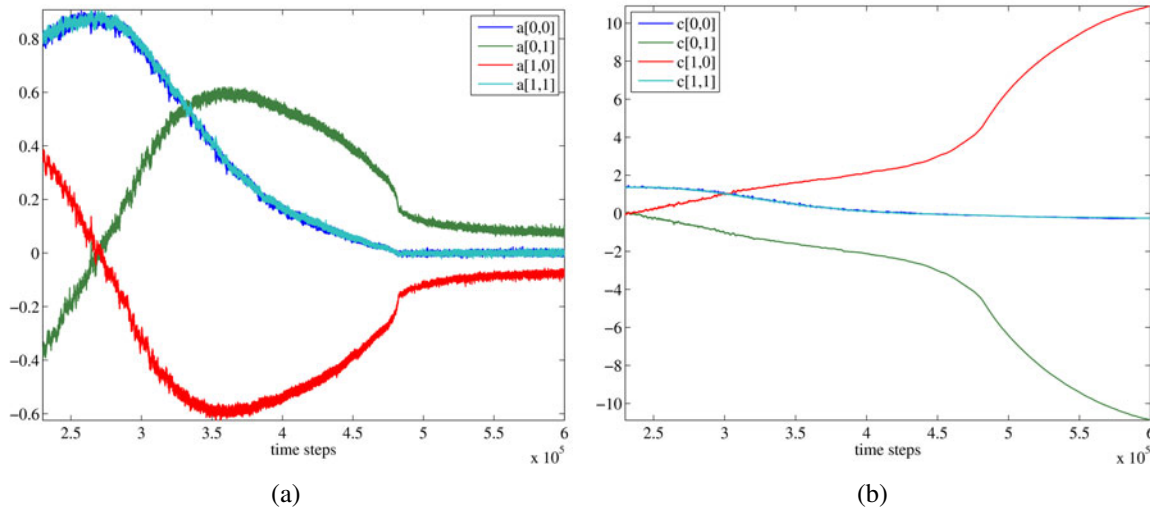


Figure 3.30: Controller and model parameters at the end of the rotational behavior (timestep 230000 to the end of the experiment). (a) Inertia effects, which can not be modeled by the system used here, lead to an incorrect model where the cross-channel response strength is large and the in-channel response strength goes to zero. (b) According to the proportionality between the synaptic strength c and the model parameters a the diagonal elements of c go to zero while the non-diagonal elements increase. Hence the sensor value of a channel mainly determines the output of the other channel. The so generated motor commands disturb the rotational mode of behavior which ends around step 480000. Later on the small model parameters represent the missing correlation between sensor and motor values. The increasing strength of the non-diagonal controller elements shows the intention of the approach to regenerate activity in the sensorimotor loop.

inertia effects can not be predicted by the simple (linear) internal model. Starting from around step 280000 the model learns a strong correlation across the channels, while the diagonal elements of the model matrix a (which represent the in-channel correlations) decrease (see figure 3.30(a)). Since the algorithm maintains the proportionality between model and controller parameter of a channel, the controller parameters show a similar behavior in figure 3.30(b). This change of the controller parameters leads to different motor commands which do not support the rotational mode of the body any more and even disturb it. The diagrams in figure 3.32 show a decrease in the amplitude of the sensor values and the frequency of the oscillation decreases as depicted in figure 3.21.

The amplitudes of the motor values stay constant while this occurs (see figure 3.32). The model has to adapt to this situation and hence all model parameters a decrease. The diagonal elements go to zero, but the non-diagonal elements stay stronger. At about step 480000 the rotational mode of behavior ends and the passive parts of the robot stop movement. The sensor values are in the range from -0.1 to 0.1 as before the skidding mode of behavior. The feedback strength breaks down to sub-critical values (figure 3.31(b)).

The thresholds h show a normal behavior, as depicted in figure 3.31(a).

Due to the driving term in the update rule the controller weights increase, so that a super-critical feedback strength can be re-established in order to regenerate a more active behavior

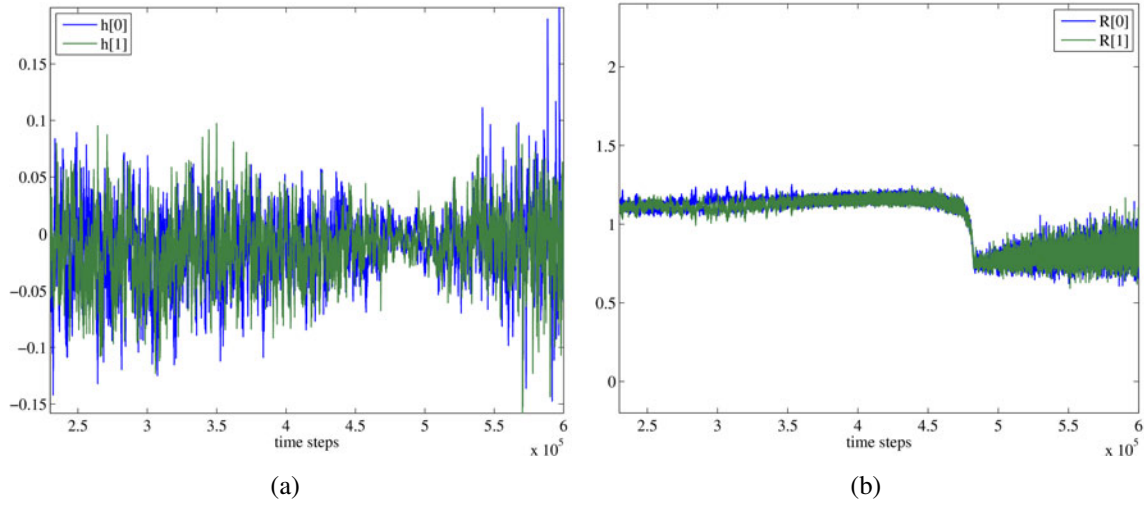


Figure 3.31: Thresholds and feedback strength at the end of the rotational mode of behavior (timestep 230000 to the end of the experiment). The thresholds fluctuate at all times around zero. The only noticeable differences are somewhat smaller amplitudes when the skidding mode of behavior ends. During the decrease of the sensor values until step 480000 the feedback strength (b) is seen to stay super-critical. A breakdown of the feedback strength is observed when the skidding behavior ends. However, due to driving term in the update rule (see section 2.4.1) R increases again and reaches super-critical values, although it strongly oscillates and is effected by the noise of the sensor values.

of the system. However, since the model keeps the non-diagonal elements of a larger than the diagonal ones (right hand side of figure 3.30(a)), the controller parameters also keep this ratio, see right hand side of figure 3.30(b). This leads to a behavior where a large sensor value in one channel generates a large activity in the other channel. Furthermore the activity is inverted in channel zero since c_{01} is negative. So if activity will be generated it is suppressed two time steps later⁴. Usually the homeokinetic control would be able to generate further active behaviors. Due to the complexity restriction of the model in this realization of the homeokinetic principle the system is not able reenter a rotational mode of behavior.

3.5.4 Summary

In the presented experiment we have shown that a controller based on the homeokinetic principle is able to generate whole-body motions of all, active as well as passive elements of the underactuated snakelike robot. This is seen at first as meandering and eventually leads

⁴ If we assume a positive sensor value $x_0(t) \approx +1$ the large positive cross-channel weight leads to a positive motor command in the other channel $y_1(t) \approx c_{10}x_0(t) \approx +1$. Defined by the physics of the system we obtain a positive sensor value in this channel $x_1(t+1) \approx +1$. The large negative synaptic strength of the cross-channels leads to an inversion of the activity $y_0(t+1) \approx c_{01}x_1(t+1) \approx -1$, which generates a negative sensor values in the next timestep $x_0(t+2) \approx -1$. This negative sensor value now leads to a motor command with the opposite sign as in time step t : $y_1(t+2) \approx c_{10}x_0(t+2) \approx -1$ and for this reason motions that were possibly generated in step t are suppressed in step $t+2$.

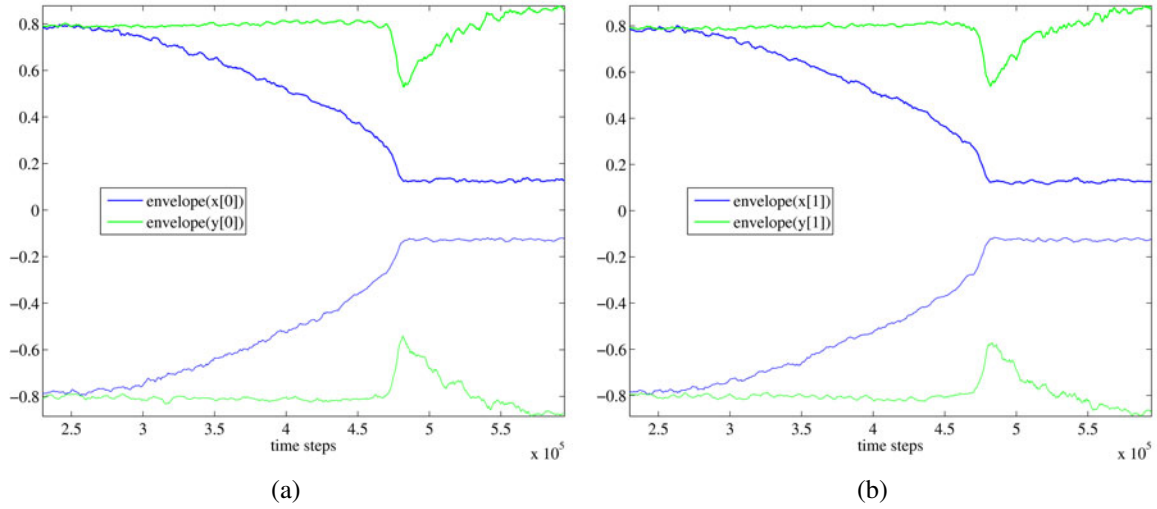


Figure 3.32: Envelopes of sensor and motor values of (a) channel zero and (b) channel one at the end of the skidding behavior (timestep 230000 to the end of the experiment). The left hand side of the diagrams shows a decrease of the sensor values x_0 and x_1 , indicating the end of the rotational mode of behavior. Around step 480000 the sensor values are mainly determined by the sensor noise. The motor values y_0 and y_1 oscillate with constant maximal amplitude until the end of the skidding behavior, followed by a short breakdown. The driving term in the update rule then leads to an increase of the synaptic strength (in this experiment the non-diagonal elements of the controller matrix c increase). Hence the feedback strengths and controller outputs increase at which point the latter reach larger amplitudes than during the skidding behavior. However, due to the actual constellation of the controller parameters (caused by the incorrect internal model) the controller is not able to excite a second phase of hurling behavior (with the realization used here!).

to a rotational mode where the body becomes stiff like a stick due to the centrifugal force of the rapidly rotating spheres of the body, (see video [Der et al. \(2008\)](#)). The generation of such a mode of behavior is only possible by exploiting the physical properties of the device. Since these properties are not known to the controller this is an example for the emergence of sensorimotor coordination.

The complexity restriction of the world model (section 2.2.1) is seen as the cause for the incorrect internal representation and hence the inability of the system to initialize a second phase of rotational behavior. However in experiments with different environmental influences (walls, other agents, see figure 3.33) the snakes are seen to reenter the rotational mode of behavior if their rotation was stopped by a collision, before the internal model learned an incorrect representation (see video [Der et al. \(2008\)](#)).

So even with the restricted internal model the algorithm was able to show the emergence of sensorimotor coordination by generating a whole-body motion in the underactuated snakelike robot.

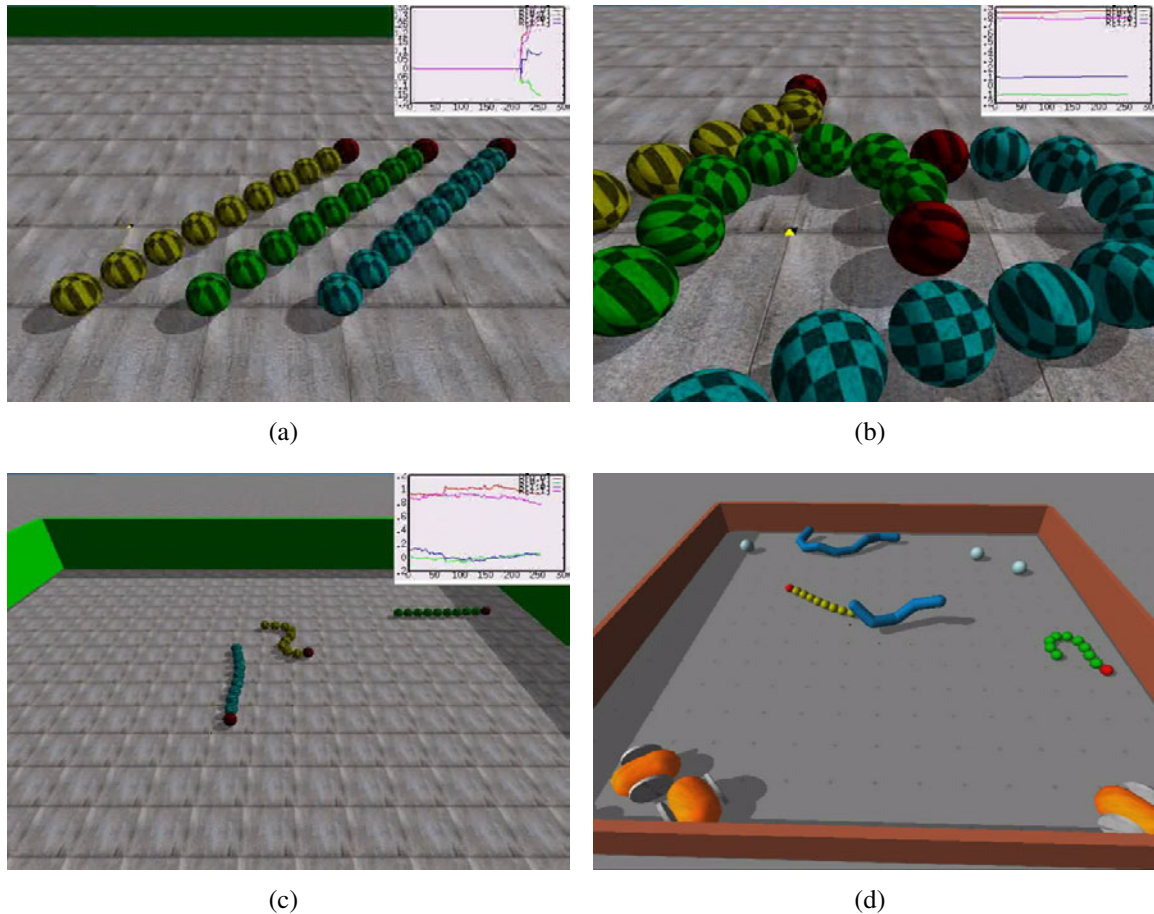


Figure 3.33: Pictures of skidding snakes. (a) Initial position in an experiment with three skidding snakes. (b) Close view. (c) In the run. Two snakes in rotational mode where the bodies become stiff like sticks due to the gyro effects of the rapidly rotating spheres of the body. The third is in z-shape with the tail passively swaying around. (d) A group of interacting self-organized creatures controlled by the presented approach (except the three passive balls). The systems try different modes of behavior (including the rotational mode of the skidding snake) even under these heavy perturbations. (See also the videos [Der et al. \(2008\)](#).)

3.6 Discussion

In this experimental chapter we have demonstrated that the homeokinetic principle, as formulated mathematically in the objective function E (confer equations 2.32 and 2.36) in applications to completely different agents, yields in each case to an environment-related active behavior. The emerging behaviors are dictated by the body of the agent.

In the first experiment (section 3.2) we have shown that the parameter dynamics of the homeokinetic controller (equation 2.39) leads to parameter values which would also generate active behaviors in a static controller. However, the proposed algorithm shows faster reactions to environmental influences and hence was able to travel nearly twice as far as the static controller with the best parameter setup in the same time. Furthermore, it was shown that the initial conditions are less important, except for the requirement that the eigenvalues of the Jacobian have to be positive.

The experiment with a two wheeled simulated robot (section 3.3) pointed out the explorative character of the homeokinetic control. A further interesting property shown is that the parameter dynamics never gets stuck in the saturation regions of the neurons or that the activity of the agents decreases for a longer time. The correlation between a large modeling error and a change of the actual behavior of the system (as a result of equation 2.41) was described in detail and found to be the key in generating rich sets of behaviors. These provide the internal model with information about body and environmental properties which can be used to improve the prediction ability.

The control of a simulated five degree of freedom snakelike robot was presented in section 3.4. In this complex device various coordinated behaviors were generated which led to an exploration of the environment as well as the body properties. Some of these modes of behavioral organization remind the observer of motions of biological agents like the crawling or jumping behavior. Thus mechanically completely different devices like the thus far considered wheeled robots can be controlled by the proposed paradigm at which point the resulting behaviors depend on both the body and the environment.

The last presented experiment (section 3.5) was an example of the generation of body related behaviors by homeokinetic control. The specific properties of the device under control could be exploited to generate a behavioral mode of whole-body rotation, where all, active as well as passive, parts of the robot are involved. The complexity restriction of the internal model in the presented realization of the homeokinetic principle caused an incorrect internal representation and hence the inability of the system to reenter new behavioral modes. This shows a point of extension for future realizations. However even with the restricted model the controller was able to show the emergence of sensorimotor coordination in this complex robotic system.

In the robotic applications presented in this chapter, exploration takes place in the action space as well as in the one, two, or three-dimensional environment. However exploration of the environment does not only mean passing each location over time, but the approach furthermore prefers to explore regions which cause problems in modeling. So large errors of the world model led the system to gather more information about that area, which can be used to improve modeling. Such a mechanism, where *by coordinating sensory and motor processes organisms can select favorable sensory patterns and thus enhance their ability to achieve their adaptive goals* (Nolfi and Marocco, 2002), is referred to as active perception.

Biological evidence is found for example by [Dill et al. \(1993\)](#) and [Franceschini \(1997\)](#), where the visual systems of flies were investigated.

To give an impression of the diversity of robots controlled following the homeokinetic approach and show that we do not only rely on simulated devices, figure 3.34 depicts three of the real robots used in our group.

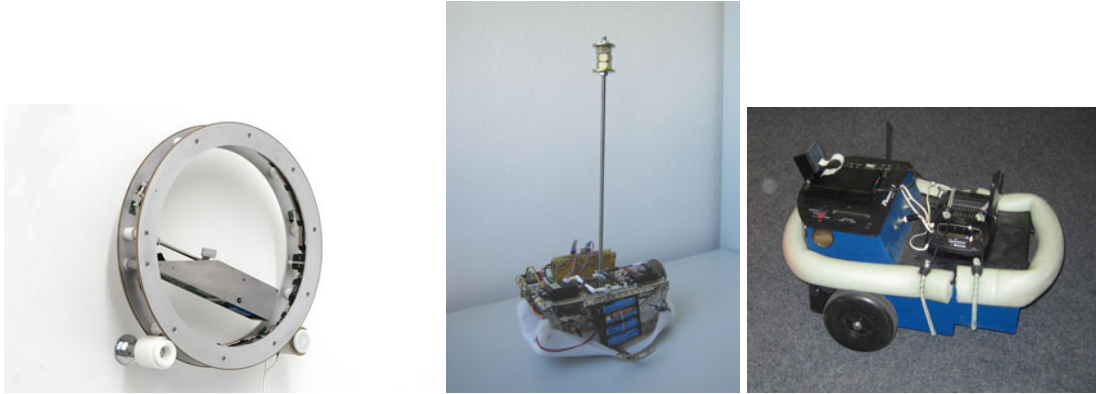


Figure 3.34: *Micro.adam* (left) is a circular robotic device designed by the artist Julius Popp ([Popp, 2008](#)) and equipped with a control following the homeokinetic paradigm. The rocking robot (center) was assembled by Georg Martius (see [Der et al. \(2006\)](#)). The Pioneer robot ([MobileRobots Inc, 2008](#)) (right) is also used for tests of the homeokinetic approach.

At the present step the behaviors, although related to the specific bodies and environments, are without goal. What we achieve so far is the concomitant learning of both explorative behaviors together with their forward model from scratch. This bootstrapping task is solved under the proposed paradigm in a more or less natural way, invoking phenomena like noise amplification and spontaneous symmetry breaking.

Chapter 4

Extending Homeokinetic Control

The homeokinetic controller (section 2.3) generates simple reactive behaviors that are interesting because of their flexibility, as we have seen in the experiments presented in the previous chapter. We are now going to modify the controller such that, in addition, prospective information can be exploited in order to allow the controller to adapt to different situations or generate preventive actions. This information may be available from more complex sensors and predictors, and is referred to as context information, the corresponding sensors as context sensors. We propose to interpret such information in terms of the low-level control which may be advantageous if no background information can be referred to for the interpretation of the high-level information.

4.1 Long-Term Memory

In this section the homeokinetic controller will be extended with an additional long-term memory, provided with context information (as presented in [Der et al. \(2004\)](#)). This memory allows the controller to adapt to different situations without relearning the parameters, supposed that context information qualifying the current situation is available. We will consider this extension for the case of sensors showing fast switching activity and characteristics, which thus can be integrated in the sensorimotor loop.

Control realized on the basis of the homeokinetic principle is achieved in tight sensor motor coupling, denoting that all sensors responding to the motor activities are automatically integrated into the generation of the motor command. In the theoretical considerations in chapter 2 was assumed that the sensor response is essentially proportional to the velocity of the robot. This is not the case for a proximity sensor. In this case one could use a preprocessing and consider the change of the sensor value between time steps. However, the problem is that this sensor characteristics (the proportionality between wheel velocity and sensor value) is valid only if the sensor is “on”, i.e. the obstacle is within the range of the sensor. Otherwise the response of the sensor is zero, even if the robot moves. Figure 4.1 shows as an example a wheeled robot with a sensor measuring the wheel velocity and a proximity (infrared) sensor in two different situations. On the left side, in a free space situation, the wheel sensor shows a larger response strength than the infrared sensor and therefore obtains a larger synaptic strength, as indicated by the thickness of the arrows from the robots

sensors to the controller. The right side depicts a “close to collision” situation where the infrared sensor response is larger than that of the wheel sensor. Hence the controller has to adapt to the different situations as fast as the switching occurs in order to make use of the sensors and react properly. Similar problems arise when new sensors are installed for some time or sensors temporarily break down.

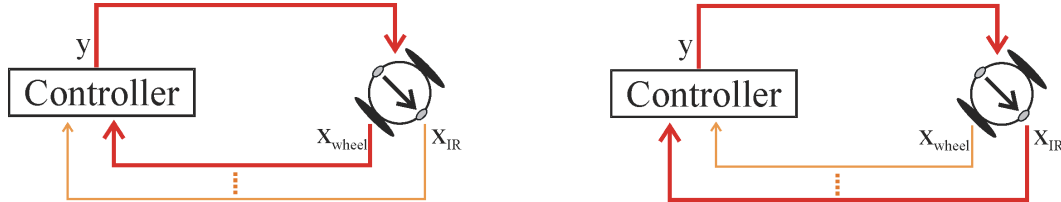


Figure 4.1: Sensorimotor loop closed over wheel and infrared sensor of a wheeled robot. If the response factor a of the wheel sensor is larger, the synaptic strength of the input from this sensor is greater than from the infrared sensor, as represented by the thickness of the arrows from robot sensors to the controller (left). If in a “close to collision” situation the infrared sensor shows a greater response than the wheel sensor, the synaptic strength corresponding to the infrared sensor will be larger (right).

Before starting the experiments let us discuss the parameter dynamics for the case of several channels (equation 2.41), which in principle may cope well with the situation. Assume a new sensor k is switched on and the value of the coupling is $c_k = 0$. In the beginning we have $\Delta c_k = \mu a_k (1 - 2Ry^2)$ since the damping term $-\gamma \mu c_k a_k$ in this channel is negligible as compared to the other channels, because of the small value of c_k . Obviously the only channel-specific term in this equation is the response strength a_k of the channel, which of course has to be adapted to the new situation by continuous model learning. So due to the driving and the anti-Hebbian term in the update rule (see section 2.5.1) c_k will adapt until $c_k = \alpha a_k$ is reached (equation 2.43). In the concrete case c_k will rise and thereby integrate the new sensor in the sensorimotor loop.

Concomitantly the couplings of the other sensors and hence the value of α is readjusted so that the global balance is reestablished. Hence a newly switched on sensor is automatically integrated into the sensorimotor loop, according to its response strength. The switched-off situation is dealt with in the same fashion.

The problem however is that the processes of readjusting the coupling vector c takes some time. In practical applications (see below) the switching on and off of sensors may take place in very short time intervals. It is therefore not of interest to relearn the couplings but instead to have a kind of long-term memory where the couplings are stored and read out appropriately. This is possible either on the basis of direct information on the state of the sensors or on context information which is able to qualify the sensor situation. We will study the latter case in the following.

Setup

The experiments have been chosen to show in a simple case that the derived learning rules generate an explorative behavior of the robot which is highly sensitive to the reactions from

the environment. This means that it will move more or less tentatively as long as the predictive ability is insufficient, i.e. the modeling error is large (the model parameters a_i are still erroneous) and with increasing predictive ability a more and more explorative behavior will originate, while staying sensitive to the environment. Furthermore, we want to investigate the situation of switching sensor activities. For the experiments a Khepera robot is used, which is placed in a moveable box which on its hand is confined in a larger area with fixed walls as borders, see figure 4.2. Details of the robot were described by Mondada et al. (1994).

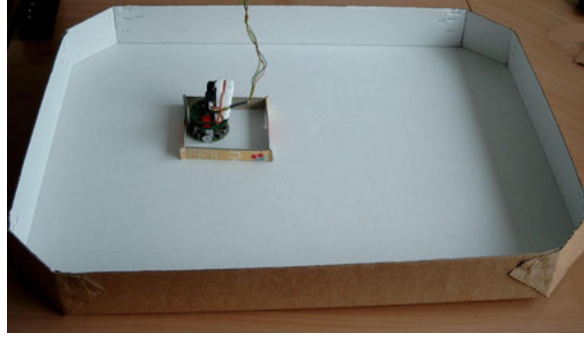


Figure 4.2: Khepera robot inside a moveable box which is situated in a larger box with fixed walls.

The Basic System

In the experiments two sensors are used, one measuring the velocity of the wheels $x_1 = v$ and a pseudo-infrared sensor. The output p of the pseudo-infrared sensor is proportional to the target velocity y of the robot so that it can be used immediately as sensory input x_2 . The pseudo-sensor is introduced to retain the simple case of linear dependencies to ease understanding, while showing the properties of the approach. The pseudo-sensor is triggered by the physical infrared sensors with outputs r_t (the moveable box is large enough to contain both *on* and *off* regions of the pseudo-infrared sensor)

$$p_t = \begin{cases} by_{t-1} & : \max_{i=0}^7 r_{t,i} > r_{min} \\ 0 & : \max_{i=0}^7 r_{t,i} \leq r_{min}, \end{cases} \quad b > 0, const \quad (4.1)$$

so that the sensor p is active as long as there is at least one infrared sensor with an activity larger then the threshold r_{min} , else the value of this sensor will be zero. Thus we have the case of a fast switching sensor.

The controller of the robot consists of a single neuron with the update rule for the membrane potential as given in equation 2.39. Its output $y \in [-1, 1]$ is the forward velocity of the robot. Due to the very fast switching of the sensor values some precautions against divergencies are conducted. Each update is pushed through a squashing function

$$\Delta c \leftarrow \eta \tanh \left(\frac{1}{\eta} \Delta c \right),$$

which introduces a maximum step width given by η .

Finally, the model parameters a_1 and a_2 are learned on-line by gradient descending the model error equation 2.9. However, in the model the measured wheel velocity depends linearly on the controller output - that is the model is appropriate only if the robot moves without problems. When colliding with the wall the model is not valid any longer and the learning should be switched off. This is achieved by multiplying the learning rate by a kind of reliance factor which is chosen as

$$f_j = \exp(-\beta \xi_j^2)$$

for channel j where $\xi_j^2 = (x_{t+1,j} - a_j y_t)^2$ is the error of the model in this channel.

The Extension by a Long-Term Memory for the Parameters

For many robotic applications the problem of switching sensor activity plays an important role. In particular, the pseudo-infrared sensor p_t can switch frequently between $p_t = 0$ and $p_t = by_t$ (see equation 4.1) according to the robot's position in the moveable box (figure 4.2). So good predictions are obtained only if the model parameter is switched according to the situation.

The sensor situation depends on the output of the physical infrared sensors r_t which can be transformed in a context m_t with the values

$$m_t = \begin{cases} 1 & \max_{i=0}^7 r_{t,i} > r_{min} \\ 0 & \max_{i=0}^7 r_{t,i} \leq r_{min}. \end{cases}$$

To solve this problem we train a neural network (with the context m_t as input) to set the value of the model parameter $a_{t,2}$ according to the context. The learning signals are directly given by gradient descending the model error.

Moreover, different model parameters a_i produce different synaptic weights (see the discussion above). Therefore the controller parameters are also represented by a neural network (with the context m as input), except for the bias h which is changing rapidly all the time (compared with the other parameters) and therefore does not need to be memorized. The long-term memory is adapted according to the parameter changes of the homeokinetic controller. The architecture used is shown in figure 4.3. With the incorporation of this long-term memory the controller is able to handle very fast switching sensors.

Results

In the experiments the parameters of long-term memory and controller are initialized with small random values (the eigenvalues of the Jacobian L of the loop function ψ are checked to be positive, see section 3.2.3). The model parameter for the wheel channel $a_1(t)$ was conveniently initialized by hand (figure 4.4(b)). The initial value of the model parameter for the pseudo-infrared channel $a_2(t)$ was set to zero and the learning of the model parameter was disabled for the first 1000 steps (figure 4.4(c)). As a result the model error is large (figure 4.4(d)) if the pseudo-infrared sensor is active which leads to an almost immediate change of the value of the bias h (figure 4.5(a)) so that the robot changes its direction of motion. The effect is that the robot avoids collisions with the walls of the moveable box it is enclosed in and hence makes only small movements (figure 4.4(a)).

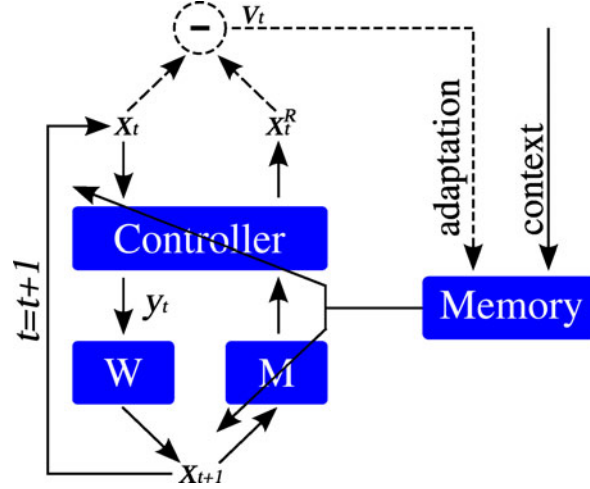


Figure 4.3: System architecture used in the experiment. The homeokinetic controller is supplemented with a long-term memory using the context information to recall the appropriate controller and world model parameters for the current situation. The long-term memory learns this parameters by observing the parameter changes of the homeokinetic controller.

When after step 1000 learning of the long-term memory takes place the model parameter $a_2(t)$ becomes more and more adapted (jumping between $a_2 = 0$ and $a_2 = b$, cf. Figure 4.4 (c)) which decreases the model error. Hence when approaching the wall of the moveable box the relearning of h (figure 4.5 (a)) does not take place and the robot starts pushing the box around.

Eventually when the robot reaches the wall of the arena (wheels get blocked) the model error is large so that the rapid relearning of the bias h and hence the velocity reversal takes place at this collision event. In this way the robot now explores the full region of the arena (see figure 4.4 (a)) by exploiting context information and long-term memory .

Approximately with step 4400 the learning of the model parameter a_2 is disabled from outside and its value is set to zero again. This leads to a large model error and fast relearning of h when the pseudo-infrared sensor is active. Hence the robot moves only in a short range of his environment like in the beginning of the experiment, so one can see that if the model for some reason is not able to make good predictions any more the robot returns to its uncertain behavior.

As for the controller parameters we observe that at the beginning and the end of the experiment the pseudo-infrared channel is not included in the sensorimotor loop ($c_2 \approx 0$, figure 4.5(b)) because a_2 is set to zero. Hence the sensorimotor loop is closed only over the wheels with $c_1 \approx 5$ (figure 4.5 (c)) and with $a_1 \approx 0.23$, leading to a feedback strength of $R \approx 1.1$ ($\alpha \approx 21.7$).

In the middle part the pseudo-infrared sensor is included in the sensorimotor loop, but only when it is active. Then the model parameter $a_2 \approx 1$ leads to the increase of the appropriate weight until $c_2 \approx 1$. The readjusted factor $\alpha \approx 1$ can also be seen in the wheel channel ($c_1 \approx 0.2$, $a_1 \approx 0.23$) so that R is around 1.1 again. When the pseudo-infrared sensor is not active the parameters should be $a_2 \approx c_2 \approx 0$ and $c_1 \approx 5$. This is not the case ($c_2 \approx 1$, $c_1 \approx 3.3$) so the feedback strength is smaller than 1 for a short time. The cause is

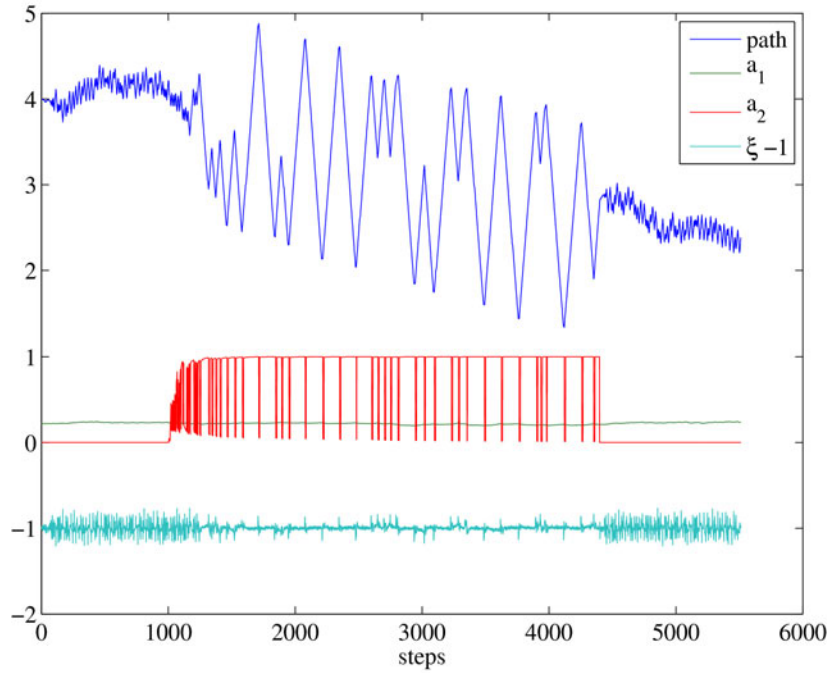


Figure 4.4: The path traveled by the robot (with odometry error). For the first 1000 the steps model learning was disabled so that the robot moves only very cautiously not pushing the moveable box. Then model learning is allowed and with increasingly predictive ability the robot starts moving the box and later explores the full range of the arena while moving the box around. In the end learning was disabled (with a_2 set to 0) again so that the cautious behavior reappears. Model parameter a_1 of the wheel channel is already learned at the beginning of the experiment. Model parameter a_2 unlearned at the beginning, then learning to jump between 0 and 1 depending on the activity of the infrared sensors and reset to zero at the end. The difference between predicted and measured pseudo-infrared sensor values (ξ ; $\xi - 1$ plotted for visibility) gets smaller when the model learns to predict (middle) and rises with resetting the model parameter a_2 to zero.

seen in the very fast switching of the pseudo sensor together with the timescale of z which is comparable to the activity time of the sensor. Hence z can not reach the fixed point value, so the theoretically derived values can not exactly be realized.

This experiment shows the uncertain, tentative behavior (changing direction of motion very often) of the robot in areas the model can not predict properly, the “brave” behavior (covering large areas, which are predictable for the model) and the change between these two behaviors through the exploitation of context information and long term memory. In doing so the robot stays sensitive to the reactions of the environment and additionally the switching sensor is integrated in the sensorimotor loop, as long as it is active.

Summary

In the presented experiment we could show that the homeokinetic approach is able to deal with the integration of newly attached sensors or removed/damaged ones. The same holds

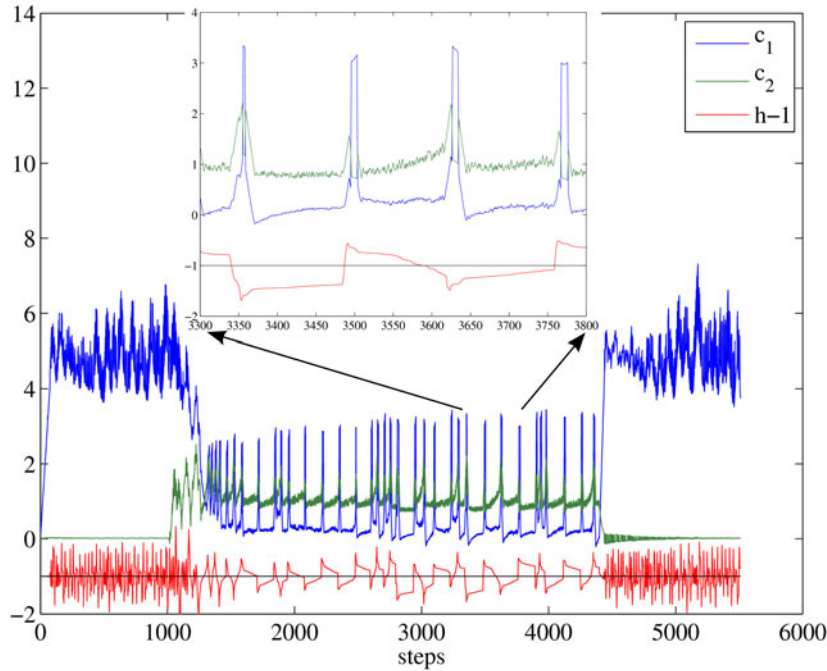


Figure 4.5: The bias h (shifted for visibility to $h - 1$), the weights for the pseudo-infrared input c_2 and the wheel input c_1 of the homeokinetic neuron used as controller. The input weights converge to different values depending on the value of the model parameter for the pseudo-infrared channel. The frequency of the changes of the bias h and therefore the time the robot travels in one direction depends on the model error.

true for sensors switching between an *on* and an *off* state. As soon as this switching happens very frequently a long-term memory is required to allow the controller to adapt fast enough to the new situation. In the presented case the additional memory for the parameters of controller and world model, provided with context information, allowed the system to cope with fast switching sensors.

In the experiment was also demonstrated that in the case of multiple sensors those which show a reasonable response to the motor actions are predominantly integrated in the sensori-motor loop. Furthermore we were able to show how sensitive the robot reacts to situations it can not properly predict. In this example the Khepera robot did not push the small moveable box it was located in as long as it did not properly predict the change of the proximity sensors when approaching a wall (as can be seen in the video [Der et al. \(2008\)](#)).

4.2 Second-Order Learning

In the following a different kind of extension than the long-term memory described in the previous section will be introduced. We combine the homeokinetic controller with an additional learning mechanism (as presented in [Hesse et al. \(2007\)](#)) which brings about the avoidance of situations that are hardly predictable by the internal model. For this purpose, a mechanism is required that is able to predict the (large) modeling error in such situations. Here we in-

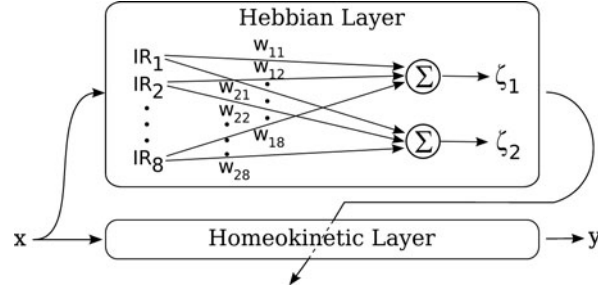


Figure 4.6: Sketch of the extended control structure. A Hebbian layer affects the homeokinetic controller (now encapsulated in the homeokinetic layer) by the predicted modeling error ζ which complements the low-level time-loop error.

roduced the prediction ζ of the error of the state estimation which is provided by a different input channel. This additional error will be associated with the low-level learning process by an additional Hebbian layer (confer figure 4.6). The error function (equation 2.36), minimized by the homeokinetic control layer, is extended by the predicted modeling error ζ which can be represented by

$$E = (L^{-1}(\xi + \zeta))^T (L^{-1}(\xi + \zeta)), \quad (4.2)$$

where the vectors ξ and ζ have the same number of elements, which is equal to the number of sensors of the homeokinetic layer. In addition to the minimization of the state estimation error ξ , the robot minimizes the prediction ζ of the state estimation error. ζ is defined to be small when context information is unavailable, such that in these cases the actual behavior is produced by the low-level controller. Otherwise the behavior will be changed such that the large modeling error ζ is reduced. The extended objective function (equation 4.2) applies only in the update rule of the threshold h , because we are interested in an extension of the frequency effect (section 2.4.3) but want to maintain the effects of the driving (see section 2.4.1) and the anti-Hebbian term (section 2.4.2) of the update rule.

The Hebbian layer is realized by a leaky integrator neuron with a linear output function for each of the sensor values x , i.e. for each x_i the homeokinetic layer provides a predicted sensor value \hat{x}_i and a specific modeling error ξ_i , the Hebbian layer provides the predicted modeling error ζ . All context information x^c is used as input to each neuron of the Hebbian layer and weighted by the synaptic strength w_{ij} according to

$$\zeta_i = \sum_{j=1}^m w_{ij} x_j^c, \quad i = 1..n, \quad (4.3)$$

with m being the number of context sensors available to the Hebbian layer and n the number of sensors available to the homeokinetic layer. The update rule for the synaptic weights is

$$\Delta w_{ij} = \varepsilon \xi_i x_j^c (1 - w_{ij}^2), \quad (4.4)$$

where ε is a learning rate and $\xi_i x_j^c$ realizes Hebbian learning between the modeling error ξ_i of the homeokinetic layer and the input x_j^c of the Hebbian layer. The update rule will produce a

positive synaptic strength w_{ij} if ξ_i is mostly positive by the time $x_j^c > 0$. A negative w_{ij} will be produced if mostly $\xi_i < 0$ when $x_j^c > 0$. A regularization term $(1 - w_{ij}^2)$ is added, which restrains the weights from unlimited increase. Nevertheless, a sensory input of 1 weighted with a synaptic strength of nearly 1 would result in a predicted modeling error of about 1, which can realize an immediate change of the actual behavior as intended. By adding the (via the Hebbian layer) predicted modeling error to the actual modeling error, the homeokinetic controller can thus avoid situations which lack low-level predictability.

4.2.1 Obstacle Avoidance for a Wheeled Robot

As a first example we consider a two-wheeled robot, where the low-level controller receives the measured wheel velocities as input. In addition, infrared sensors are available as context sensors.

The modeling error describes differences between predicted and measured wheel velocities. The predicted modeling error is used to modulate the homeokinetic layer in order to change the actual behavior before arriving at situations with a large modeling error, which refers to collision situations in the example.

In the experiments we will show that an obstacle avoidance behavior of a two-wheeled robot equipped with infrared sensors can be obtained, solely based on the intrinsic properties of the system. The effectiveness of the obstacle avoidance is not perfect since the system tries occasionally to also explore the regions near the boundaries. Nevertheless, the time the robot spends near obstacles will be drastically reduced.

4.2.1.1 Setup

The initial setup of the experiments consists of a simulated two-wheeled robot with infrared sensors, placed in a circular arena with diameter of 14 length units (see figure 4.7). For the simulations the *lpzrobots* software package (section 3.1) was used.

The Hebbian layer is provided with proximity information from eight infrared sensors with a sensor range of three length units. In order to suppress small noisy activity in the infrared sensors, only sensor values larger than 0.15 are considered. The synaptic strengths w_{ij} of the Hebbian layer are initialized with zeros. The parameters of the homeokinetic layer are initialized with small random values, the eigenvalues of the Jacobian L of the loop function ψ are checked to be positive (see section 3.2.3).

4.2.1.2 Results

In a first experiment only the homeokinetic layer was used. Experiment 2 was done using the extended controller. Each experiment runs for one hour simulated real time in the *lpzrobots* simulation package. To obtain some information about the long-term stability a third experiment was conducted that lasted 24 hours.

The trajectory of the robot in the first two experiments is plotted in figure 4.8. The positions of the robot concentrate increasingly on the inner obstacle-free region when using the Hebbian control layer as compared to pure homeokinetic control. The histogram of the robot's distance from the center of the arena illustrates the effect of the learning scheme (see

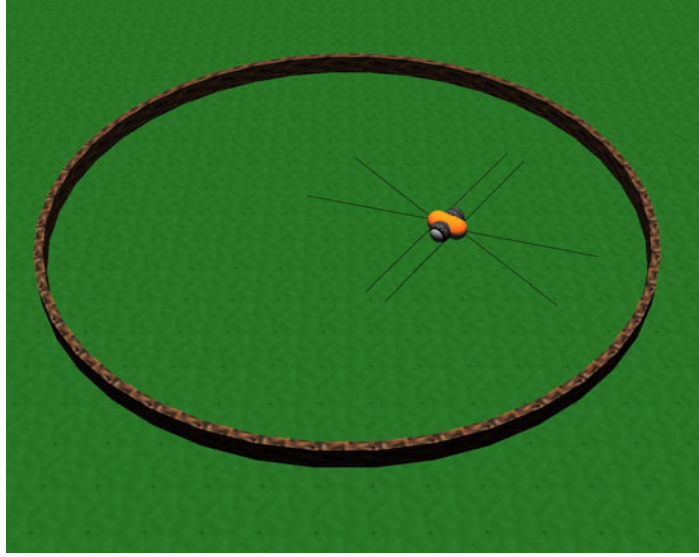


Figure 4.7: Experiments are performed with a two-wheeled robot in a circular arena. The difference to the setup presented in figure 3.2 is that the robot here is equipped with wheel counters and eight infrared sensors. The black lines indicate the IR sensor orientation and range. The sensor range is 3 and the diameter of the arena is 14 length units.

figure 4.9). During the first part of the experiment (top row) the Hebbian layer started to adapt but shows hardly any effect on the robot's behavior yet. Hence the histograms show similar distributions. The bottom row of figure 4.9 shows histograms of the robot's position during a later part of the experiment where the influence of the Hebbian control layer is dominant. Without access to the Hebbian layer the robot's probability of staying near the wall is approximately three times higher than being at any other distance from the center (confer figure 4.9 (bottom left)). This is caused by the fact that in the central obstacle-free region of the arena behaviors are more stable due to the small modeling error and hence larger distances covered by the robot, whereas in the region near the wall behaviors change more often due to a larger modeling error and the robot is not able to cover large distances. Therefore, the robot's probability to stay near the wall is higher. When enabling the Hebbian layer the robot's probability of being near the wall is drastically reduced, and the highest probability is now shifted towards the center of the arena (see figure 4.9 (bottom right)).

The predicted modeling error of the Hebbian layer leads to a change of the actual robot behavior before the collision region is reached. Since the selection of the following behavior is not constrained the robot can still reach the collision area, but with much less probability. This can be interpreted as a flexibility of the system which continues to explore the collision area.

The usage of the predicted modeling error in the homeokinetic layer leads to pre-collision changes of the robot's behavior rather than to the trivial solution where the robot stops somewhere in the central region of the arena. In figure 4.10 the traveled distance of the robot with and without usage of the Hebbian layer is shown. Regions of inactivity are essentially absent. Also, the total traveled distance is not reduced by incorporating the Hebbian layer.

In the long-term experiments the parameter dynamics works as in the pure homeokinetic

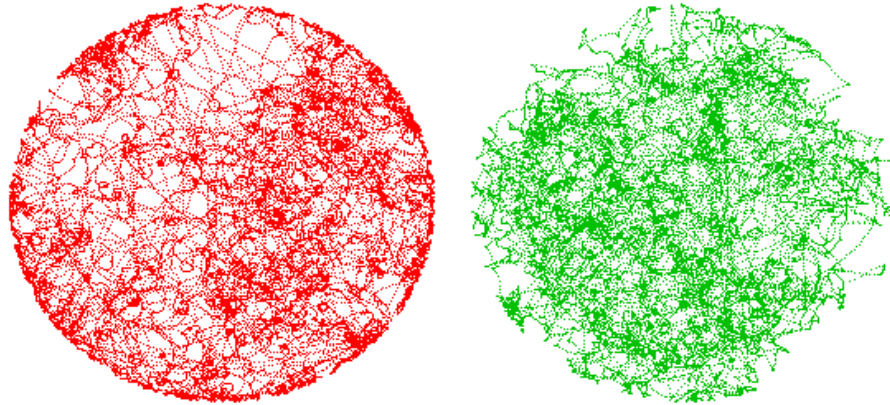


Figure 4.8: Trajectory of the robot using pure homeokinetic control (left, also depicted in figure 3.7) and the extended controller (right). An increasing concentration of the robot's positions in the inner obstacle-free part of the circular arena can be identified when using the Hebbian control layer, as compared to pure homeokinetic control.

system, and realizes a slightly supercritical feedback strength as expected from theory (see section 2.5.2). The comparison of table 4.1 and table 3.2 shows no remarkable differences, except for the standard deviation of the bias h , which increased by a factor of nearly 4. This shows that the frequency effect, which is responsible for the rate of change of the bias (see section 2.4.3), is now more active, as intended by this setup.

The weights of the Hebbian layer during the 24 hour experiments show that the learned correlations are indeed based on the behavior of the robot (confer figure 4.11). The two front infrared sensors are included with a negative sign. Note that, if the wheel counters indicate forward motion by $x > 0$, then the predicted velocity will typically also be positive $\hat{x} > 0$. Near a wall the front infrared sensor will be active, but after collision the velocity sensor will yield $x = 0$, while the prediction is still $\hat{x} > 0$. Hence, ξ will be negative. So by converging

(a)			(b)		
parameter	mean	standard deviation	parameter	mean	standard deviation
$c[0, 0]$	1.2130	0.0921	$a[0, 0]$	0.9728	0.0311
$c[0, 1]$	0.0132	0.1478	$a[0, 1]$	-0.0044	0.0191
$c[1, 0]$	0.0024	0.1615	$a[1, 0]$	-0.0040	0.0216
$c[1, 1]$	1.2301	0.1097	$a[1, 1]$	0.9636	0.0377
$h[0]$	-0.0090	0.2073			
$h[1]$	0.0027	0.2247			

Table 4.1: Mean value and standard deviation of (a) the controller and (b) the model parameters of a twenty four hour experiment.

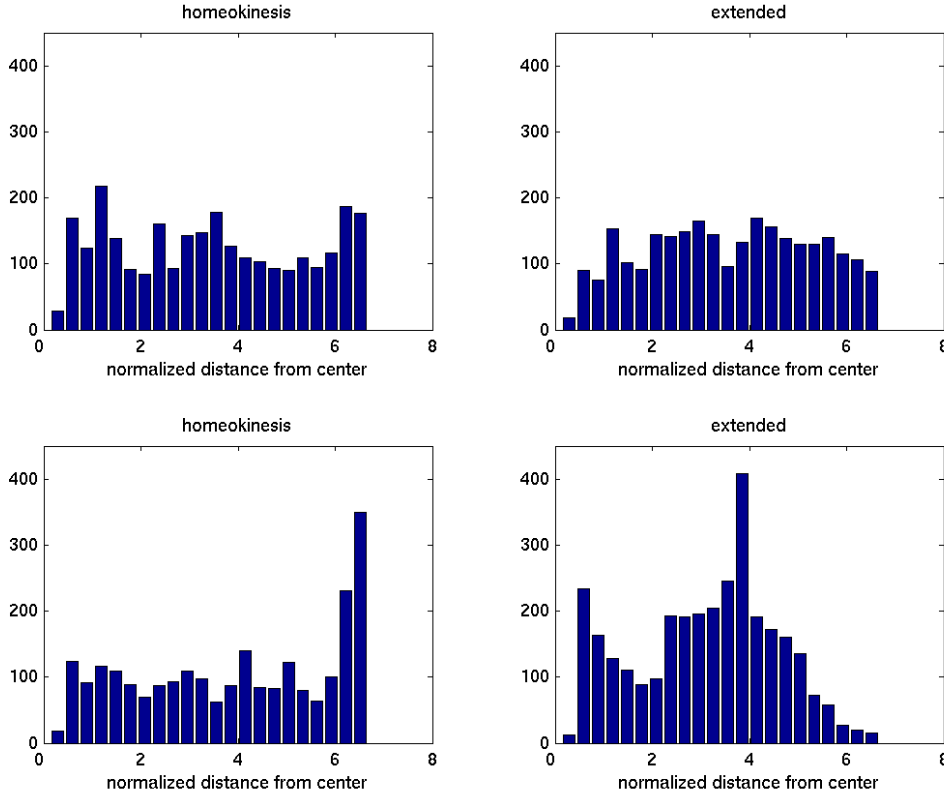


Figure 4.9: Histogram of the robot's distance from the center normalized by the respective areas for pure homeokinetic control (left column) and the extended controller (right column) of the first 15 minutes (top row) and last 15 minutes (bottom row) of the experiments with a total time of 1 hour. In the initial phase the Hebbian layer is not yet functional and both controllers show comparable results. In the later part of the experiment (bottom row) the mean occupancy has shifted away from the wall towards the center of the arena in the case of the extended controller.

to negative weights for the front infrared sensors the Hebbian layer extracts this correlation and is able to predict a negative future modeling error ζ . The same holds true for the rear infrared sensors with inverted sign for weights and modeling error. For the sideward sensors the correlations are not significant.

4.2.1.3 Summary

In this experimental proof of principle we could show that combining the homeokinetic controller with a second order learning mechanism based on Hebbian learning can shape the behavior of the agent under control, while keeping the properties of the self-organizing system. In the concrete realization a collision avoidance behavior could be realized, based on the minimization of the predicted error of the state estimation.

If the effect of the additional error term is inverted the robot will move only in the vicinity

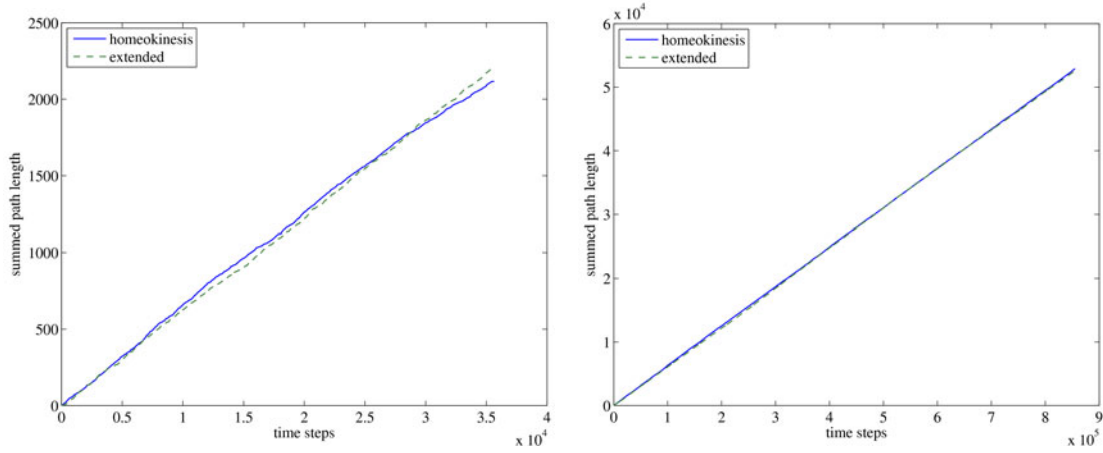


Figure 4.10: Cumulative distance traveled by the robot over time using pure homeokinetic control and the extended controller. The traveled distances in the two experiments are comparable, indicating that the Hebbian layer did not reduce the activity of the robot. This holds true for (a) the one hour as well as for (b) the twenty four hour experiment.

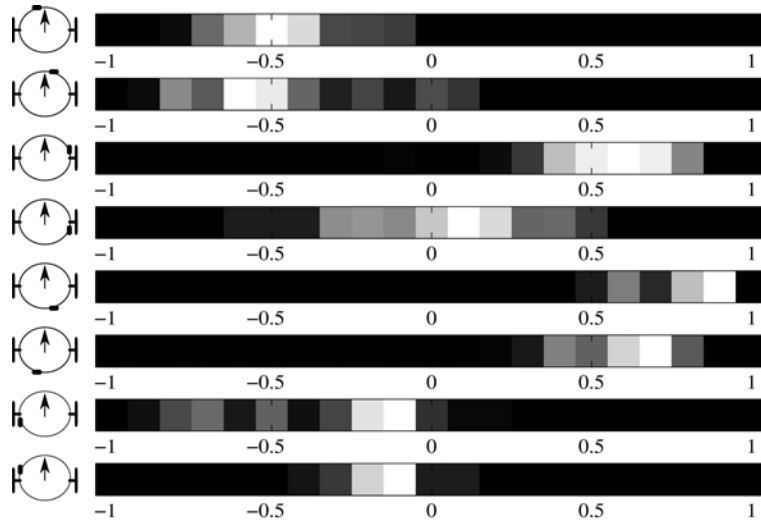


Figure 4.11: Histogram of the weights of the Hebbian layer contributing to ζ_1 for a long-term experiment (24 hours simulated real time) with extended controller. Bright color corresponds to many, dark color to zero (or only a few) counts in that interval. The labels on the left show the locations of the infrared sensors, the arrow indicates the forward direction. Front and rear sensor weights have negative and positive sign, respectively, indicating the ability of the Hebbian layer to correctly extract the correlations between modeling error ξ and IR sensor activity.

of the wall. In this way the robot increases its opportunity to adjust its internal parameters such that it is able to move freely near walls. The robot's preference for wall in this modified scheme suggests it as a model for a foraging rat, confer e.g. the work of [Tamosiunaite et al. \(2008\)](#). We will study the modified principle in a more complex hardware set-up in the following section.

4.2.2 Gripping in a Human-Hand Model

The proposed combination of second-order learning and homeokinetic control should now be further investigated in application to a more complex robotic device – a simulated human hand model. Besides the question of the transferability of the presented principle to different devices, it should be investigated whether the behavior of the system can also be shaped in a different way than to avoid situations lacking low-level predictability.

4.2.2.1 Setup

For the following experiment a model of a human hand with 5 degrees of freedom was programmed in the *lpzrobots* software package (section 3.1) (see figure 4.12). All joints

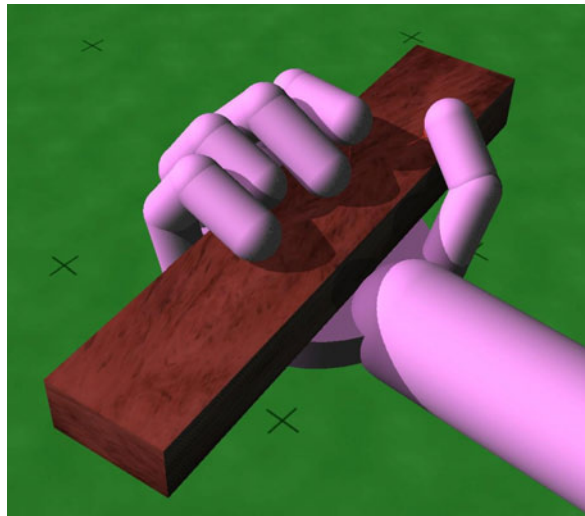


Figure 4.12: Simulation of a human hand with multiple degrees of freedom. The hand is equipped with motion sensors at all joints and infrared sensors at the finger tips. It is operated in a fully exploratory mode with or without a manipulated object.

are controlled by bidirectional motors that mimic the interplay between flexor and extensor muscles. The effect of a motor action is measured by motion sensors, which serve as input to the low-level homeokinetic controller. Each finger is controlled by an individual controller such that interactions between the fingers are possible only via the environment. If no object is present for manipulation the fingers become quickly engaged in vivid movements which can be interpreted as an exploration of the dynamical range. In the presence of an object the modeling errors increase considerably when the fingers touch the object, because this is not

predicted by the internal model. Context information about objects in the hand is provided by infrared sensors in the finger tips.

In this experiment we exploit the directionality extracted by the Hebbian layer (see figure 4.11) by directly adding the output of the higher layer to the update of the threshold h in equation 2.39. This allows a targeted change of the behavior of the system in contrast to the setup described in the previous section.

Like in the previous experiment the synaptic strengths w_{ij} of the Hebbian layer are initialized with zeros. The parameters of the homeokinetic layer are initialized with small random values (the eigenvalues of the Jacobian L of the loop function ψ are checked to be positive, see section 3.2.3).

4.2.2.2 Results

Applying the combined controller to the simulated hand leads to the generation of vivid motions of all fingers, as long as there is no object in the hand. After adding an object for manipulation and giving the Hebbian layer some time to adapt, the fingers will flinch when arriving close to the surface of the object but remain active otherwise like in the free case.

By changing the sign of the contribution of the higher layer to the bias update we can shape the behavior of the system in order to show a gripping reflex. Hence as soon as the Hebbian layer is adapted and the context sensors indicate the presence of an object in the hand, the fingers will grip the object. Figure 4.12 shows a picture of the simulated hand holding a bar of simulated wood.

Details of the adaptation process for one synaptic strength of the Hebbian layer are shown in figure 4.13. Only when the modeling error and context sensor have values different from

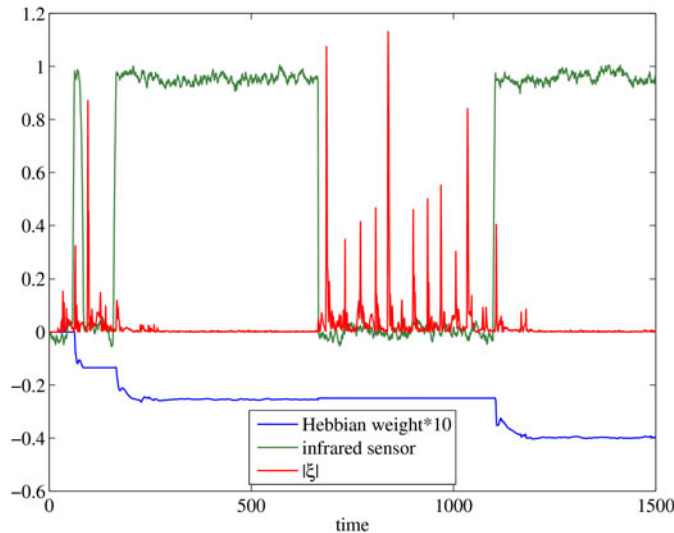


Figure 4.13: Adaptation of the synaptic strength of the Hebbian layer accounting for the middle finger and the attached infrared sensor during the experiment. According to equation 4.4 the change of a weight is defined by the corresponding modeling error ξ and context sensor x^c (in this case the infrared sensor). If both values are considerable a change of the weight is triggered.

zero an adaptation of the corresponding Hebbian weight will take place. Remember that the infrared sensor activity is considered zero as long as it is smaller than 0.15 to suppress sensory noise. Hence the Hebbian weight in figure 4.13 does not change when the infrared sensor value is around zero.

The results of the adaptation of the Hebbian layer are shown in figure 4.14. At the beginning the motor commands (lower diagram) indicate vivid movements of the fingers while an object is present. Hence the infrared sensors change between full and no activity. As the prediction ability of the Hebbian layer increases over time, it starts to dominate. Thus the gripping behavior is preferred, as indicated by the positive motor commands (closing the hand) and infrared sensor activities around 1 (fingertips touch the object). When the object is removed the infrared sensors are inactive and the motor commands show the vivid movements of the fingers again. If the object reappears the fingers will immediately grip it, since the Hebbian layer is already learned (right hand side of figure 4.14).

4.2.2.3 Summary

We could show that even in this more complex hardware setup the combination of homeokinetic control and second-order learning, realized by a Hebbian layer, leads in a first instance to a behavior where areas with a large modeling error are avoided. This was achieved by interpreting only information of the low-level control, namely the modeling error already calculated in the homeokinetic layer.

When providing a further internal mechanism or giving a goal from outside the behavior can also be shaped differently. This was shown in the presented experiment by the generation of a gripping reflex in the simulated hand model (see videos [Der et al. \(2008\)](#)).

4.3 Discussion

In the first experiment (section 4.1) the question of sensor integration was investigated. The experiments showed that the homeokinetic control is able to integrate online newly attached sensors as well as to no longer consider removed or damaged ones. Thus the system exhibits a lifelong plasticity and is able to adapt to the actual situation. The additional long-term memory provided with context information enabled the system to adapt to different situations without relearning of the parameters. Thus also sensor showing fast switching activity and characteristics can be integrated in the sensorimotor loop. The basic behaviors shown by realistically simulated hardware agents in section 4.2 are obtained from the interplay of the mildly destabilizing homeokinetic controller with the environment, which is constrained by the prediction quality achieved by an internal model. In unforeseen situations, i.e. “obstacles”, parameter changes are triggered which are time-consuming and may even cause unlearning of previously acquired behaviors. The proposed second-order learning schemes are coping with such situations in different ways: Either the robot is controlled so as to avoid these situation, which generates an interpretation of additional sensory inputs in terms of the low-level affordances, or the robot is guided towards these situations in order to further improve its prediction quality. The decision which mode of operation of the second-order learning is to be activated is to be taken depending on the quality of the internal model, such

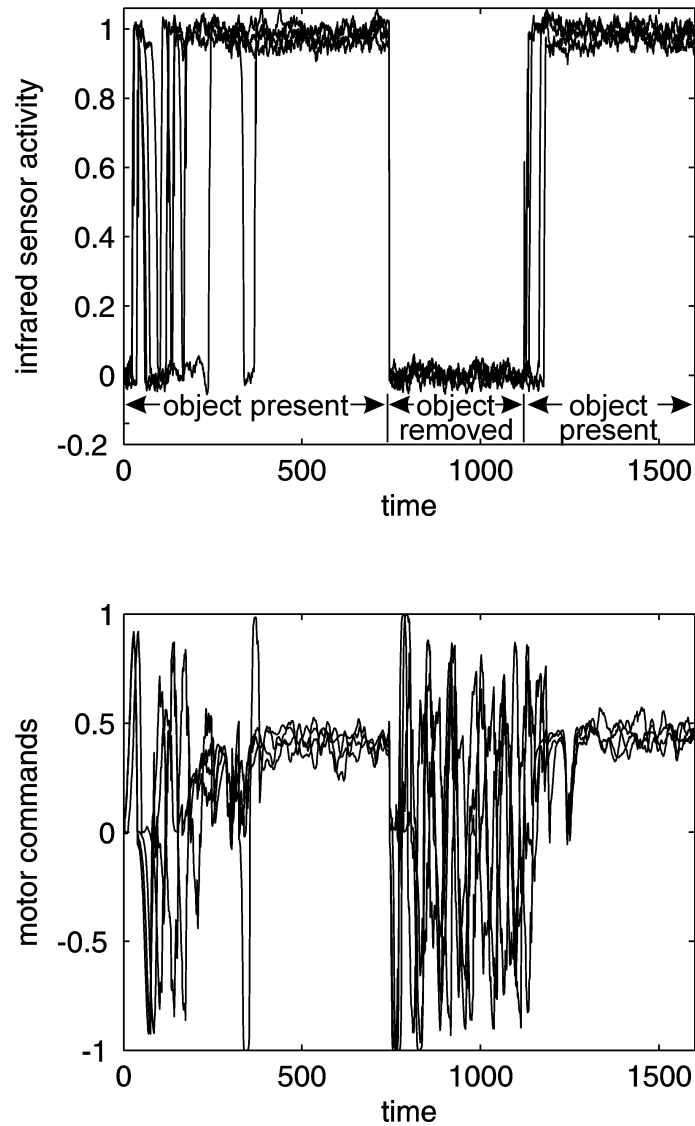


Figure 4.14: The finger movements that are initiated by the self-organizing controller soon converge to a grip of the object (high infrared sensor activity) with only small deviation of single fingers from the surface. When the object is removed, the exploratory movements restart. If the object is present again the fingers will immediately grip it since the Hebbian layer already learned this reflex.

that increasing prediction quality should favor the explorative mode, while unsurmountable errors should lead to a preference of the avoidance behavior. The explorative character of the low-level self-organizing controller is retained in both cases and the robot still occasionally explores risky regions and is hence able to adapt to slow changes in the environment. The work also shows parallels to the early motor development in biology, confer e.g. the work of [Kuniyoshi and Sangawa \(2006\)](#), and provides a scheme for the formation of reflexes based on an approach to the self-organization of autonomous behavior.

Chapter 5

Self-Organization for the Control of Myoelectric Prostheses

5.1 Introduction

Since a long time prosthetic devices have brought back a new quality of life to people which for some reason lost a limb. Since the seventies of the last century myoelectric control especially for the upper limb has enormously increased the capabilities of prosthetic devices. In myoelectric control signals from the muscles of the residual limb, which have no use anymore due to the loss of the limb, are recorded by electrodes placed on the skin above the muscles. By contracting these muscles the amputee can thus send a control signal to the prosthetic device. In this chapter we will focus on forearm prosthesis with myoelectric control. The decoding of the muscle signals is a challenging problem, since in the forearm there are a lot of muscles next to and on top of each other (see figure 5.1(a)). Additionally they can move against each other. This anatomic fact complicates the mapping from measured signals to muscle activity (Farina et al., 2004). That is why so far amputees have to train special movements with their (imaginary) limb, which are known to give reliable results in

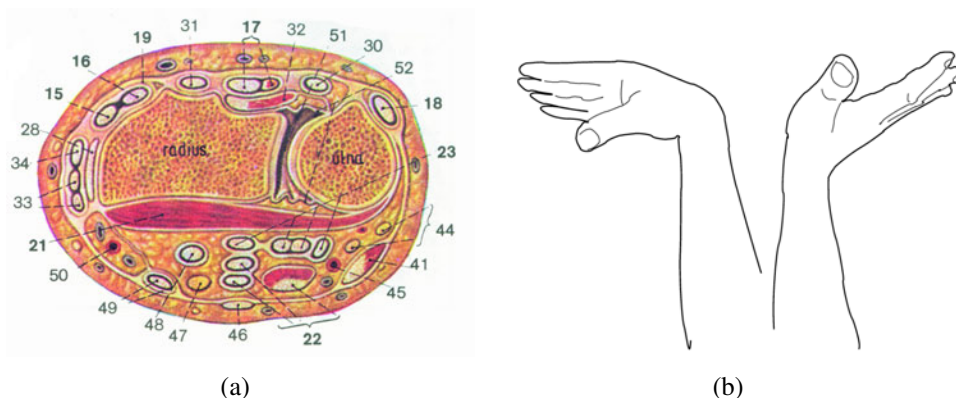


Figure 5.1: (a) Cross section of a human forearm showing the complexity of the anatomical structure (Lippert, 1995). (b) Wrist flexion and wrist extension of a healthy hand (Willburger, 2007). These motions are commonly used as control commands for prosthetic devices.

most cases. Wrist flexion and extension, as shown in figure 5.1(b), are commonly used to control the two directions of a one degree of freedom device proportional to the extent of the muscle contraction. Usually the hand aperture is controlled in such prostheses.

According to [Sears and Shaperman \(1998\)](#) the majority of patients prefers the use of an additional wrist rotator over a single motor setup. Controller of prosthesis with more than one degree of freedom nowadays available to amputees are realized as state machines. This means the device can be in different states where one and the same muscle contraction leads to different movements depending on the current state. Hence the patient has to keep the actual state of the device always in mind in order to be able to correctly control it. The transition between different states is realized by a co-contraction of flexor and extensor muscle. This works in practice but the patients have to accustom to this movement. Furthermore it is time consuming, so complex movements of different degrees of freedom can only be executed in a serial order. Figure 5.2 depicts a state machine for a prosthetic device with two degrees of freedom: the abilities to rotate the wrist and to control the aperture of the hand (open/close the hand).

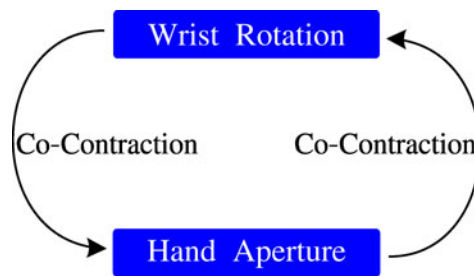


Figure 5.2: State machine for a two degree of freedom prosthetic device. Each of the two rectangles represents the control of one degree of freedom of the device. State transitions are triggered by co-contractions as indicated by the arrows.

For the use of prosthetic devices it would be beneficial if more than one degree of freedom could be controlled by extracting different features from the surface electromyographical signal instead of using state machines. This becomes more and more important when robotic and also prosthetic devices with many degrees of freedom will be available (figure 5.3). The control of such actuators should be realized in a proportional manner, where the velocity and grip force of the prosthesis are controlled proportional to the strength of the muscle activation, as preferred by amputees according to a study by [Sears and Shaperman \(1991\)](#).

In recent years approaches using multi layer perceptrons and/or support vector machines to differentiate between more control signals were tested, leading to an increased number of motion commands and the abolition of the co-contraction and different states of the device. Hence the usability of prostheses was considerably increased. One necessity of this methods is the execution of prespecified motions by the amputee. The muscle activity of these motions is recorded in order to train a classifier. When later using the prosthetic device the patient has to try to generate the same motions, since these are required to control the prosthesis (see for example [Eriksson et al. \(1998\)](#); [Karlik et al. \(2003\)](#); [Willburger \(2007\)](#)).

Inspired by this topic a mechanism allowing the adaptation of the motion commands according to the abilities of the patient in an automated manner, based on the homeokinetic

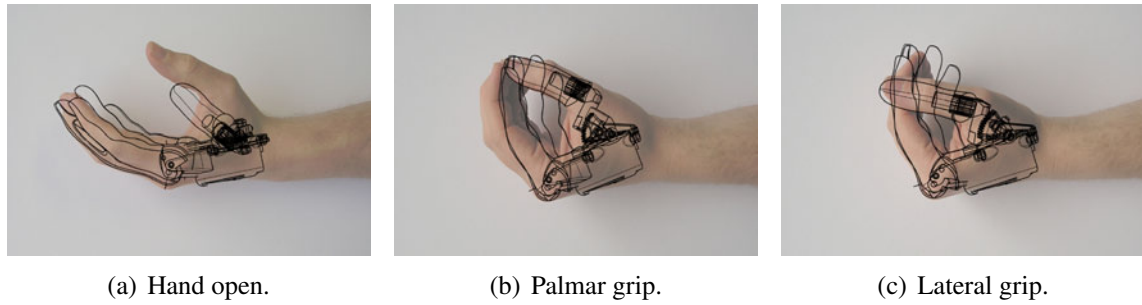


Figure 5.3: Different types of grip: Future prosthetic hands will be able to perform different types of grip similar to those sketched here. Conventional control schemes are not suitable. Advanced control schemes are required. (Willburger, 2007).

approach (as presented in chapter 2) was investigated, as will be explained in the following section.

5.2 Self-Organization in Prostheses Control

The goal of applying the self-organization approach (section 2) to prosthesis control is twofold. On the one hand we want to come up with an individual set of surface electromyographical signal features which allows patients to control the prosthetic device. On the other hand we want to increase the capabilities of the patient to produce discriminable signals. The field of application is seen in a training period, previous to the use in everyday life. In this period the patient-specific set of features is to be found in an interaction process between patient and prosthesis. Since the homeokinetic controller generates already a little activity of the prosthesis, patients, especially patients with weak signals (for example due to degenerated forearm muscles), get the impression to be able to bring the device into motion and could thus be encouraged to work with the system for longer periods. Thereby the required muscles will be strengthened, and the signal quality and the controllability are expected to be improved.

The control loop in this case will include two learning systems, namely the controller and the subject (or patient), compare figure 5.4. Motor commands of the controller are passed to the prosthesis. The movements of the latter can be visually observed by the subject. The reactions of the subject, measured via the surface electromyographic signal, are used as input to the self-organizing controller and build the basis for the generation of the next motor command.

In the beginning of the training period the prosthesis will start movements even if only noise is applied to the sensors, based on the driving term in the update rule (as described in section 2.4.1). This marks the beginning of the interactive process. During the interaction the patient tries to support the movements shown by the prosthesis with motions he prefers and is able to generate. If after some minutes the prosthesis does not react to this movements, other movements have to be tried. The controller on the other hand will detect motion signals which are useful to control the prosthetic device (as described in section 2.5 and further

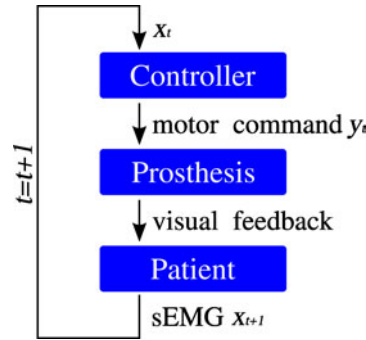


Figure 5.4: Sketch of the control loop for a prosthetic device with self-organizing controller. Motor commands generated by the controller actuate the prosthetic device. This can be observed by the subject via visual feedback. The surface electromyographic signal (sEMG) measuring the muscle activity of the subject is used as input signal of the controller, on which new motor commands are based.

investigated in section 4.1). These control commands are not prespecified and therefore allow an adaptation to the individual patient.

The control realized by the self-organization approach is proportional if the features provided as input to the controller are proportional. Since the motor commands are generated as a linear combination of the sensory inputs (the nonlinearity only confines the absolute value of the motor commands) the controller keeps the proportionality, provided the latter existed in the inputs. In a possible further processing step the motor commands could also be discretized to use them in binary mode, or to weight them with the overall muscle activity obtained, for example, from different features.

5.3 Equipment

The experimental setup consisted of special electrodes, an analog-to-digital converter, a personal computer and software, as described in the following.

5.3.1 Electrodes

Horst Willburger (Otto Bock Healthcare Austria) developed and provided the electrodes used in this work. Such an electrode consists of four contact surfaces attached to a carrier, as depicted in figure 5.5.

Since these electrodes do not carry ground contacts, ground connection was established with a separate electrode. Three voltage signals U_i were calculated by building the difference between the potentials P_j of neighboring contacts:

$$U_i = P_i - P_{i+1} \text{ with } i = 1..3 \quad (5.1)$$

With this setup a surface electromyographical signal with 3 channels could be provided from each electrode. On the electrode the signals were amplified by a factor of 10 and before

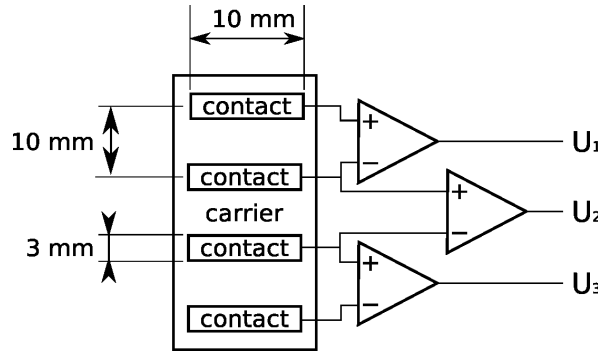


Figure 5.5: Sketch of a used surface electrode with four contacts. A differential surface electromyographical signal pick-up was used, providing three channels $U_1 \dots U_3$.

entering the analog-to-digital converter they were again amplified by a factor of 100 on a special circuit board.

One electrode was placed at the flexor digitorum superficialis and a second one at the extensor digitorum (forearm). Hence the recorded surface electromyographical signal consists of six parallel channels, which are then processed by the analog-to-digital converter.

5.3.2 Analog-to-Digital Converter

In the experiments a DAQCard-6036E analog-to-digital converter from [National Instruments \(2008a\)](#) was used. The sampling rate was chosen to be 3000 samples per second. From the analog-to-digital converter the recorded digital data were delivered to the LabVIEW software for further processing.

5.3.3 LabVIEW

In the presented experiments LabVIEW ([National Instruments, 2008b](#)), a graphical programming framework for signal processing, control applications and the interaction with measurement hardware, contains the main functionality consisting of the preprocessing of the acquired data, the feature extraction and the controller. As first processing step a high pass filter with 10 Hz cut of frequency and a band-stop filter (to filter out any noise from the 50 Hz power line) were applied to the signals. Further details of the feature extraction can be found in the setup section of the respective experiment.

When using a real prosthetic device to be controlled by the presented approach this could be done by sending the control signals via a digital-to-analog converter to this device. In the current setup the motor commands were send via the pipe mechanism¹ to a simulation software, which will be described in the next paragraph.

¹ In software engineering, a pipe connects different data processing entities, so that the output of one element is the input of the other. Usually some amount of buffer is provided between the elements.

5.3.4 Hand Prosthesis Simulation

Due to the need for a visual feedback a three-dimensional physics simulation of a hand prosthesis was created using the Open Dynamics Engine ([Smith, 2008](#)). Forces and torques can be applied to any body and motors are available. Figure 5.6 shows a screenshot of the hand model with sixteen degrees of freedom. In the simulation all degrees of freedom can be

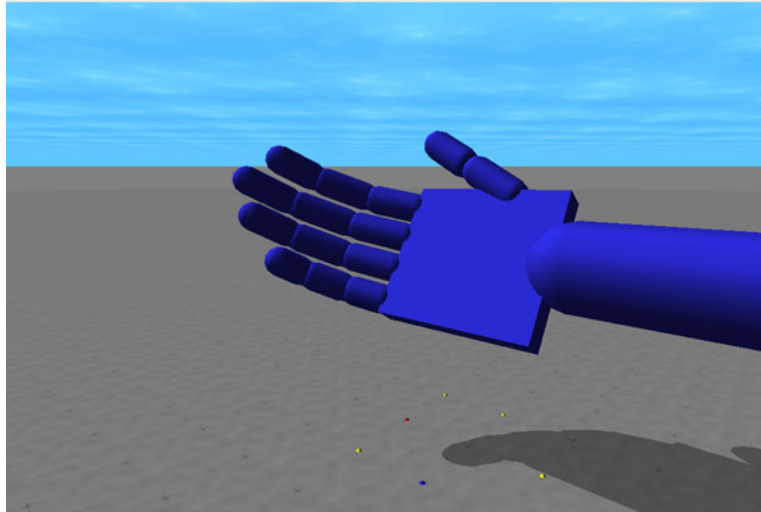


Figure 5.6: Screenshot of the three-dimensional hand simulation based on the Open Dynamics Engine. Sixteen degrees of freedom are independent and simultaneously controllable in this hand model. To match current prostheses the degrees of freedom are coupled to result in a device with two degrees of freedom. One for wrist rotation and one for the hand aperture.

independent and simultaneously controlled. To fit the properties of current hand prosthesis the degrees of freedom were coupled to result in two independent and simultaneously controllable degrees of freedom. One degree of freedom is used for open and closing the hand while the other is used for rotating the wrist clock- or counterclockwise.

5.4 Illustrative Example

5.4.1 Goal

In this experimental setup we will investigate the possibility of a self-organized control of a myoelectric hand prostheses with one degree of freedom. We are not focusing on the optimization of the myoelectric signal generation nor the extraction of features from these signals. Our interest is to find out if it is possible to equip a myoelectric prosthesis with a self-organizing controller, in order to select those features of the myoelectric signal which allow the subject to control the prosthetic device. This selection process is done in interaction of subject and prosthesis in a training period, previous to the use of the prosthetic device.

5.4.2 Setup

In the following experiment six signals were used as input to the self-organizing controller. Four signals were extracted from the six available surface electromyographical signals. Signal s_1 is the difference between the root mean square values of two voltage signals U_1 and U_4 (see section 5.3) obtained above the antagonistic muscles:

$$s_1 = \text{RMS}(U_1) - \text{RMS}(U_4),$$

where the root mean square value for a collection U of N values $\{u_1, u_2, \dots, u_N\}$ is

$$\text{RMS}(U) = \sqrt{\frac{1}{N} \sum_{i=1}^N u_i^2}. \quad (5.2)$$

N was chosen to be 512. With a sampling rate of the analog-to-digital converter of 3000 samples per second, roughly six values are calculated per second. Signals s_2 to s_4 are the differences between the root mean square values of two voltage signals within the same electrode:

$$\begin{aligned} s_2 &= \text{RMS}(U_1) - \text{RMS}(U_2) \\ s_3 &= \text{RMS}(U_2) - \text{RMS}(U_3) \\ s_4 &= \text{RMS}(U_5) - \text{RMS}(U_6). \end{aligned}$$

In order to simulate unusable channels the two remaining signals s_5 and s_6 contain white noise and are therefore hardly predictable by the internal model. Hence these signals should not be used for the control of the prosthetic device (according to section 2.5). The sensor values of the homeokinetic control are $x = s = (s_1, s_2, s_3, s_4, s_5, s_6)^T$. The motor command y generated by the homeokinetic controller is directly used for the control of the prosthetic device.

5.4.3 Results

The presented experimental run, conducted with a healthy subject, lasted seven minutes. The learning of the controller parameters was stopped when the patient reported the feeling of being able to control the prosthetic device. This happens when features are selected by the controller which allow the subject to control the prosthesis. Then a test was conducted, where the task of the subject is to move the prosthesis according to a given reference signal.

Figure 5.7 shows the input signals x and the motor command y generated by the controller during the experimental run. It can be obtained by visual inspection x_1 gives the strongest response to the movements initiated by the subject. The generated motor command is very noisy in the beginning, but subsequently it gets more similar to x_1 due to the adaptation of the controller parameters, as explained below.

The model parameters during the experimental run are shown in figure 5.8(a). Since the sensor value x_1 (see figure 5.7) gives the strongest response to the subjects movements the model parameter a_1 shows the largest absolute values of the model parameters. The other sensor values do not depend that strongly on the motor command y .

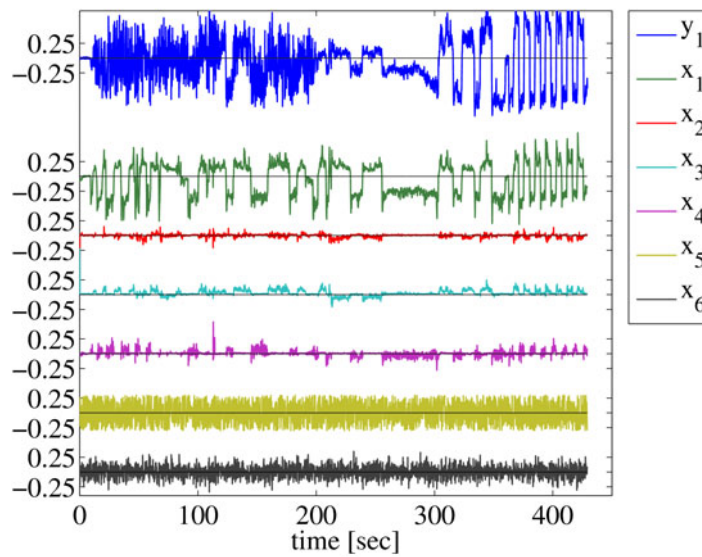


Figure 5.7: Inputs x to and motor command y generated by the self-organizing controller during the experiment. The inputs x_1 to x_4 were obtained by subtracting the root mean square values of the electromyographical signals. The inputs x_5 and x_6 are noise signals, which do not depend on the muscle activity of the subject. The generated motor command y is in the initial part very noisy, since the inputs x_5 and x_6 are relatively strong weighted by the synaptic strength c (compare figure 5.8(b)). Later on the inputs obtained from the electromyographical signals (x_1 to x_4) are weighted stronger and the motor command becomes more similar to x_1 . After around 360 seconds the subject reported the ability of being able to control the prosthesis. At this point learning was disabled for the test (see figure 5.9(b)).

After fluctuations and different constellations of the controller parameters the synaptic strength c_1 is dominating (figure 5.8(b)), according to the parameter of the internal model and the proportionality between c_i and a_i (see equation 2.43). The contribution of the other sensor values to the generation of the motor command are negelectable. With this parameter setting the subject gets the impression of being able to control the prosthetic device. Hence the adaptation of the controller parameters was stopped, in order to run the test with this set of controller parameters and hence selected features.

The feedback strength in the loop, as obtained by model and controller parameters is seen to increase in the course of time and reach slightly supercritical values (figure 5.9(a)), as expected from the homeokinetic controller (see section 2.4).

The reference signal and motor command during the test of the adapted controller and hence the selected input signals is shown in figure 5.9(b). A combined visual and acoustic stimuli was used to indicate the desired movements of the prosthetic device the subject should realize. This stimuli is plotted in the diagram as reference signal r . The plot of the motor command shows that the patient was able to generate the desired sequence of input signals and hence the desired motions.

The solution found in this setup is not surprising. The input signal s_1 was the most promising signal provided and is also used in nowadays prosthetic devices. The interesting point, however, is that this solution was found. The self-organizing controller, in interaction

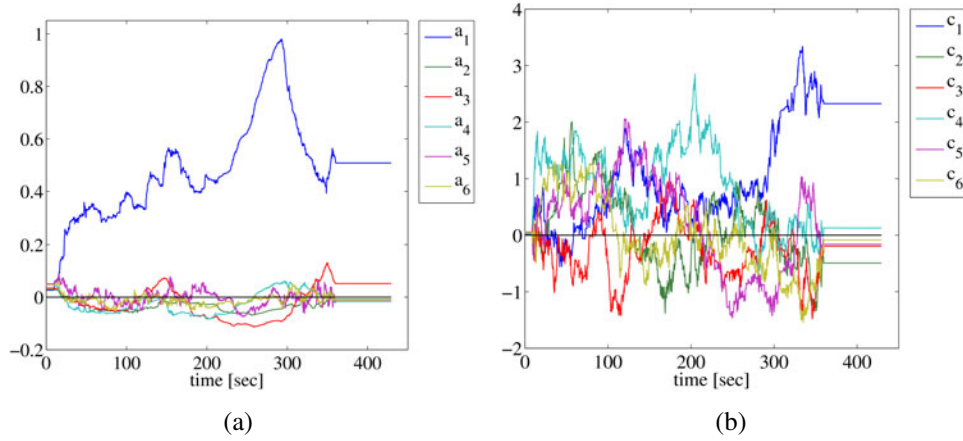


Figure 5.8: Parameters of (a) the internal model and (b) the controller. It can easily be seen that the model parameter a_1 has the largest absolute value. This denotes that the motor command y provides useful information about the input signal x_1 . The latter can also be seen in figure 5.7, where x_1 shows the strongest response to the motions executed by the subject. In the initial part of the experiment the controller parameters change continuously, showing many different constellations. From about 300 seconds on the synaptic strength c_1 is dominating, reflecting the values of the model parameters. After approximately 360 seconds the subject reported the ability to control the prosthetic device, hence the adaptation of the parameters of model and controller was disabled.

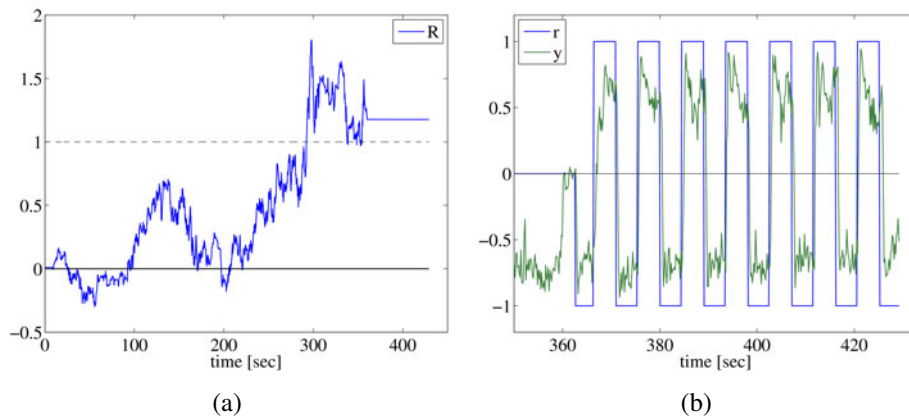


Figure 5.9: The course of (a) the feedback strength R in the loop during the experiment and (b) the reference signal r and control command y during the test after successful training. The feedback strength increases from the small initial value and reaches supercritical values after circa 300 seconds. After approximately 360 seconds the adaptation of the parameters of the self-organizing controller was disabled for the test depicted in (b), resulting in a constant feedback strength. In the test the reference signal r represents the stimuli presented to the subject, indicating the desired movements of the prosthetic device he should initiate. The motor command is the signal generated by the controller based on the muscle activity of the subject. The subject succeeded in generating the desired sequence of motions of the prosthetic device.

with the subject, was able to select this signal for the control of the prosthetic device from a set of given signals, without further information provided.

5.5 Controlling a Prosthesis

5.5.1 Goal

In this second experiment the self-organizing controller, provided with features of electromyographic signals as inputs, is used to control a two-dimensional prosthetic device. Assumed as above that the selection of prevalent features of the surface electromyographic signal we used is suitable for this task, we focused on the question if it is possible to equip a myoelectric prosthesis with a self-organizing controller, in order to select those features of the myoelectric signal which allow the subject to control a two-dimensional prosthetic device. The adaptation of the controller is done in interaction of subject and prosthesis in a training period, previous to the use of the prosthetic device. The idea is that later on a controller for the everyday use of the prosthesis can utilize the selected features to control the prosthetic device.

5.5.2 Setup

In the following experimental setup five signals extracted from the recorded electromyographic signal were used as input to the self-organizing controller. The first four signals are obtained by simply superposing adjacent voltage signals U . This superposition scheme proposed by Willburger (2007) requires little computational effort and exploits information about the motor unit action potential propagation, which varies with different motions. Motor unit action potentials are generated when a motor neuron stimulates the connected muscle fibers at the neuromuscular junction (see figure 5.10), from where the action potential travels to the tendons (where the muscle is fixed to the bone). According to Willburger (2007) the superposition scheme performs similar or even superior to time domain and frequency domain features in the proposed system, which was supported by the tests we conducted previous to choosing the setup for this experiment.

The processing steps to calculate the superposition feature are depicted in figure 5.11, and read

$$f_i = \frac{\text{RMS}(U_i + U_j) - \text{RMS}(U_i - U_j)}{\text{RMS}(U_i + U_j) + \text{RMS}(U_i - U_j)},$$

where $j = i+1$, except for $i = 3$ and $i = 6$ where the first voltage signal of the corresponding electrode is used ($j = 1$ or 4 respectively).

The first two signals used as inputs to the self-organizing controller are one of the superposition features of each electrode:

$$\begin{aligned} s_1 &= f_5 \\ s_2 &= f_1 \end{aligned}$$

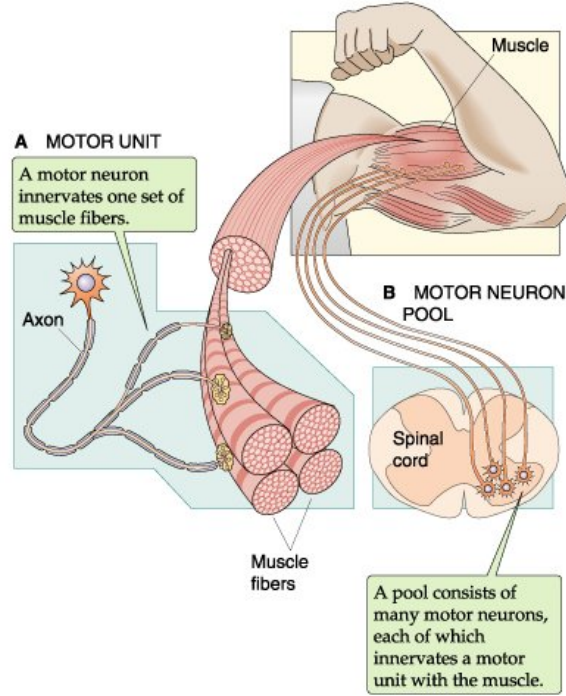


Figure 5.10: Muscle, motor unit and the neuromuscular junction at the transition from the axon to the muscle fibers, exemplified at the upper arm. (Boron and Boulpaep, 2003).

The input signals two and three contain differences of the signals from the two electrodes:

$$\begin{aligned}s_3 &= f_2 - f_5 \\ s_4 &= f_3 - f_6\end{aligned}$$

The last signal is similar to the one used in the previous section

$$s_5 = \text{RMS}(U_2) - \text{RMS}(U_5).$$

Hence, the sensor values are $x = s = (s_1, s_2, s_3, s_4, s_5)^T$. The motor command $y = (y_1, y_2)^T$ generated by the homeokinetic controller is directly used for the control of the two-dimensional prosthetic device.

5.5.3 Results

The following experiment was conducted with a healthy subject and lasted about one hour. The subject had seven trials from about the same length before, where the two degrees of freedom could not be independently controlled. Hence, in the current setup several hours of interaction are required to eventually be able to control the prosthesis. Another point to mention is, that subjects have problems to remember the specific hand position leading to one of the movements of the prosthesis, since small deviations of the finger position can

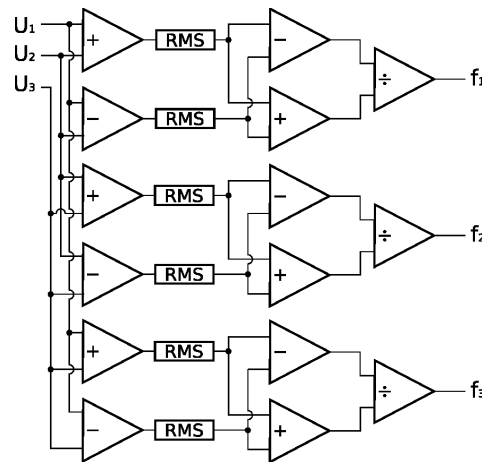


Figure 5.11: Superposition scheme outlining the processing steps to calculate the superposition feature for the voltage signals U_1 to U_3 of electrode one.

lead to different inputs for the self-organizing controller. This is caused by the fact that the superposition features are very sensitive. On the one hand this allows to provide distinct information to control the prosthetic device, on the other hand it can be too sensitive for remembering the hand position for a specific motor command.

The adaptation of controller and model parameters during the experiment is shown in figure 5.12. The internal model, which tries to adapt to the mapping from motor commands to the features, does not show a dominating channel like in the previous experiment. Rather all model parameters show considerable absolute values. This means that all sensors depend on the motor command. Reflecting the model structure all sensors will be included in the generation of the motor command, as depicted in figure 5.12(b) (compare section 2.5). The adaptation of controller and internal world model was disabled, when the subject reported the feeling of being able to control the prosthetic device, which happened after about half an hour.

The feedback strength in the loop developed to positive values, even so R_2 was subcritical when the learning was disabled (figure 5.13(b)). The input features during the interaction period of subject and prosthetic device are shown in figure 5.13(a). By visual inspection no favorable feature can be identified.

The test of the ability to control the prosthetic device with the selected input features was conducted after the interaction period, when learning was disabled. In this test, the subject is instructed to generate movements of the prosthetic device as indicated by visual and acoustic stimuli. Figure 5.14(a) shows the five features of the surface electromyographic signal during the corresponding motions. The motor commands generated by the homeokinetic controller based on these inputs and the reference signal r indicating the movements the subject should generate are depicted in figure 5.14(b). Three different combinations of the two motor commands x_1 and x_2 were generated by the subject, plotted as three distinct parts in the diagram. The first part shows movements where the motor command y_1 oscillates between positive and negative values while the motor command y_2 stays at positive values. This means the wrist of the prosthesis rotates inwards (pronation) and outwards (supination) while it is open. The

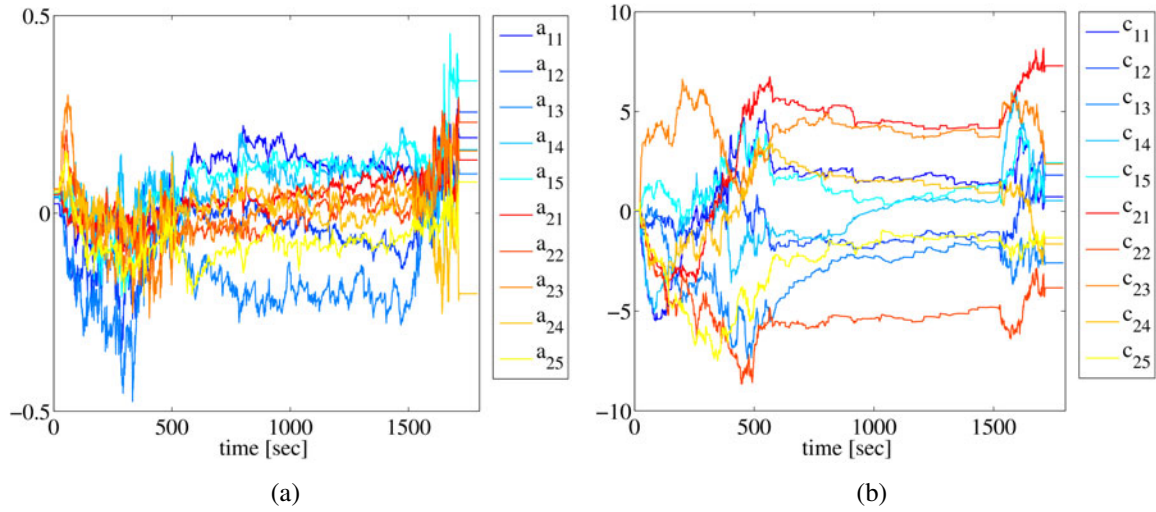


Figure 5.12: Parameters of (a) the internal model and (b) controller. The adaptation of the model parameters shows no favorable input feature, since all absolute values of the parameters are in about the same range. This is also the case for the controller parameters, where a combination of input signal is used to generate the control command for the two degrees of freedom rather than one channel is exclusively selected. After 1700 seconds (≈ 28 minutes) the adaptation of the homeokinetic controller was disabled, since the subject reported the feeling of being able to control the prosthesis.

second part shows an oscillation of both motor commands, with opposing sign. This corresponds to a supination with simultaneously opening of the hand alternating with a pronation with simultaneously closing of the hand. The third part shows motor command y_2 oscillating between positive and negative values while y_1 stays positive. In this case the hand opens and closes while the wrist rotates inwards.

This experiment shows that during an interaction period of a subject and a prosthesis equipped with a self-organizing controller input signals are automatically selected, which allow the subject to control the two degrees of freedom of the prosthetic device.

5.6 Summary

For the one and the two-dimensional case it was possible for the subject to control the prosthetic device with the features selected by the self-organizing control in the interaction period.

In the one-dimensional case the homeokinetic controller selected the root mean square feature for control of the prosthetic device. With this feature the subject was able to control the prosthesis according to a given reference signal. This feature is also used in nowadays prosthetic devices. Hence the solution found is not surprising. But that the self-organizing controller, in interaction with the subject, was able to select this signal from a set of given signals, without further information provided, is remarkable.

In the two-dimensional case a combination of the provided features was selected and

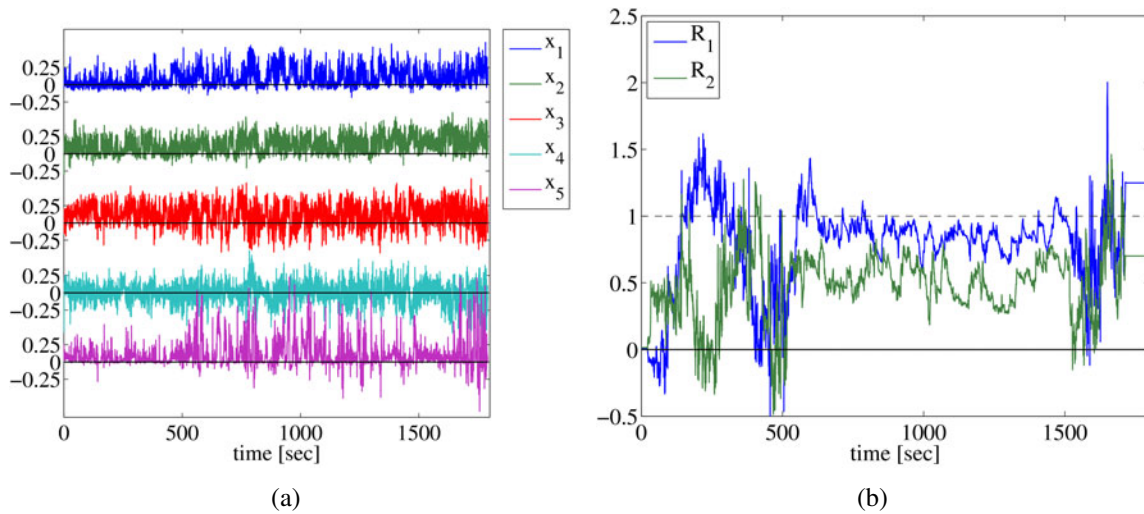


Figure 5.13: (a) Features of the surface electromyographic signal used as inputs to the controller and (b) the feedback strength in the loop during the interaction of patient and prosthesis. All input signals show activity during the interaction period. A single, promising feature can not be identified. The feedback strengths go to positive values, R_1 reaches supercritical values.

allowed the subject to control wrist rotation and hand aperture of the prosthetic device according to the reference signal. A difficulty was observed based on the sensitivity of the used feature. Small changes of finger positions resulted in different inputs to the controller and hence different motor commands. This made it sometimes problematic for subjects to remember the correct hand positions for the motor commands.

5.7 Discussion

In the presented experiments we have shown that it is possible to select input signals for the control of a prosthetic device in an interaction period of a subject and a prosthesis equipped with a self-organizing controller.

An interesting point is, that the input features are not selected according to the signals the subject provides in the beginning, when in the usual scheme the signals are recorded and a classifier is learned with these signals. Here the subject is encouraged to interact with the device, thereby generating different movements and input features. Due to this training period the subject may provide better input signals to be used later on.

During the experiments it was observed that the used superposition features are very sensitive to small changes of the finger positions, which made it sometimes problematic for subjects to remember the correct hand positions for the motor commands. Nevertheless the self-organizing controller found the sensitive patterns indicating the possibility to control even more degrees of freedom. For future work the use of larger electrode arrays or ring electrodes, perhaps incorporating different features, is expected to ease and improve the possibilities of the feature selection, since more input channels and hence more information

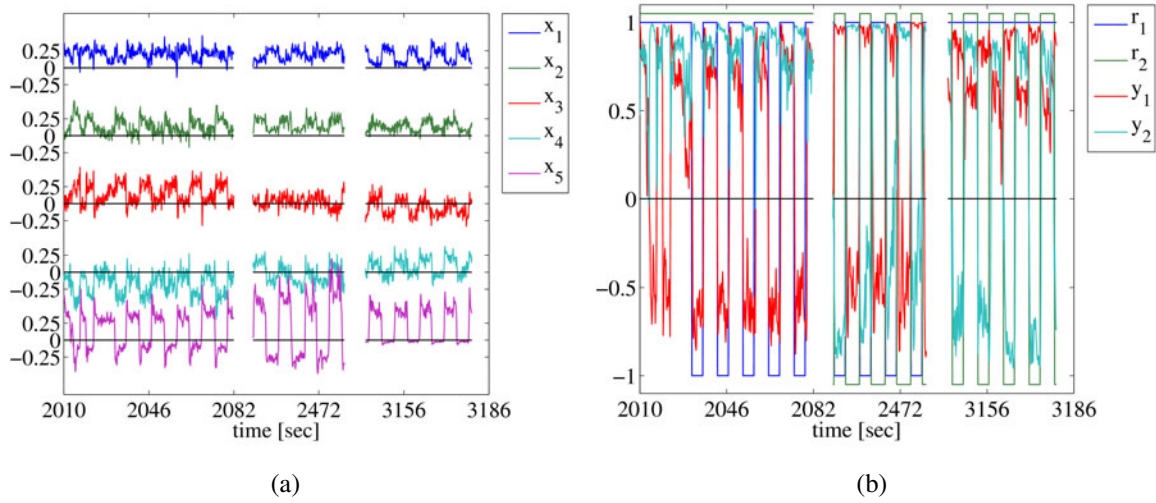


Figure 5.14: (a) Sensors values and (b) motor commands of the homeokinetic controller during the controllability test. In the test the subject was able to control the two degrees of freedom independently. In the first part of the diagram y_1 oscillates between positive and negative values as indicated by the reference signals, while y_2 stays at positive values. In the second part both motor commands change sign, but keep opposite sign. The third part shows an oscillation of y_2 while y_1 stays at positive values.

about the muscle activity can be exploited.

If the self-organizing controller is provided with more input signals the adaptation of the parameters will require more computational effort. Serializing the parameter updates, for example update only the parameters corresponding to one sensor in a time step, can reduce the computational effort. However, since the calculations during the interaction period are executed on a personal computer and not on the prosthesis itself, no computational problems are expected.

In general, if several degrees of freedom are available it is possible, that some actuators focus on the same input provided to the controller, since there is no direct feedback from actuators to sensors, as for example in robotic applications. Hence some actuators can select the same input features, resulting in coupled degrees of freedom. For future work, especially when more than two degrees of freedom should be controlled, an additional competitive learning scheme (see for example the work of [Rumelhart and Zipser \(1985\)](#)) applied to the internal model, not allowing similar couplings for different actuators, is expected to solve this.

The time required for generating a motor command is determined by the effort required to calculate the feature values used as input to the homeokinetic controller. The controller requires basically only one matrix multiplication of the controller parameters with the input values to generate the motor command. Hence also faster processing setups than the one used here are possible, depending on the effort required for the feature extraction.

With the results achieved so far we could show that the selection of appropriate input channels by interaction of a self-organizing controller and a human is a feasible approach.

The idea is not limited to prosthesis control. Other areas of application where input features for the control of actuated devices or for interaction with electronic media have to be selected are imaginable, for example surface electromyographic signal or electroencephalogram based control of robotic actuators or game consoles.

Chapter 6

Conclusion and Outlook

In this thesis a realization of the homeokinetic approach to self-organized acquisition of behavior in autonomous robotic systems, proposed by Ralf Der ([Der and Liebscher, 2002](#); [Der, 2003](#)), was introduced and analyzed with respect to different questions in various experiments. Extensions to the homeokinetic controller, allowing to adapt to different situations and to shape the generated behaviors, as well as an application to the control of myoelectric prostheses were investigated.

Homeokinesis follows the dynamical systems approach. Based on the mathematical description of the sensorimotor dynamics, the controller parameters can be adapted with reference to an objective function, which in case of the homeokinetic controller is completely internal to the robot. Since this objective function is the error arising in a time loop it is called time-loop error.

In chapter 2 we showed that homeokinesis can be used to derive concrete learning rules even though it is a general domain-invariant principle, hence not depending on the concrete realization of the robot nor the environment. If the controller of an autonomous robot is equipped with these rules the robot develops domain-related behaviors. Without relying on prior knowledge about the robotic body or the environment, body- and environment-related behaviors will be generated in the sensorimotor loop in a self-organized way. As known from physics, self-organization results from the compromise between a driving force which amplifies fluctuations and a regulating force which tries to constrain the system. In the homeokinetic paradigm the interplay between regulating and destabilizing force has its counterpart in the two terms of the objective function. On the one hand the system tries to increase the predictive ability of an internal model, which aims for smooth behaviors. On the other hand it tries to increase the Jacobian of the system which will destabilize the sensorimotor dynamics. Hence small changes in the sensor values will be amplified and lead to activity of the system and to the generation of different behaviors over time, while the requirement for predictability keeps the system back from chaotic actions.

The body- and environment-related behaviors shown in this work arise in a self-organized fashion, as a result of the interplay of state and parameter dynamics in the proposed system. Since the concomitant learning of controller and the internal model starts from scratch homeokinesis shows a way to face the cognitive bootstrapping problem (see section 1.6). This problem arises because on the one hand the controller needs a feasible model to exploit the properties of the device under control. On the other hand the model has to be provided

with necessary information about the behavior of the body which requires controller actions making this information available.

Considering the update rule of the homeokinetic controller we find the following three effects: (i) The generation of activity (section 2.4.1) in the sensorimotor loop by increasing the synaptic strength of the controller neurons and hence the feedback strength in the loop. Eventual sensory noise is enough to bring the system into motion due to the noise amplification effect. (ii) The requirement for predictability (section 2.4.2) prevents an unlimited increase of the synaptic strength of the controller neurons and realizes a slightly supercritical feed-back strength in the sensorimotor loop as soon as a system is active. Thus the systems generate behaviors at the borderline between predictable but inactive and active but unpredictable/chaotic behaviors. (iii) The frequency effect (section 2.4.3) as the third component in the update rule is responsible for the change between different behaviors by switching between the fixed points of the system. Thereby the time span of a specific behavior depends on the strength of the modeling error.

In hardly predictable situations (e.g. the two-wheeled robot near the wall, section 3.3) the frequency effect causes frequent changes of behavior. Hence the system is provided with a lot of information resulting from different behaviors in order to improve the predictive abilities in this situation. On the other hand behaviors which are well predictable (e.g. in the obstacle free part) are kept for longer time. Hence, these regions are swiftly crossed. Thus the homeokinetic control realizes active perception (Nolfi and Marocco, 2002).

In chapter 3 it was shown that in experiments with various robotic devices the homeokinetic principle, after solving the bootstrapping problem, generates environment- and body-related, active behaviors. In section 3.2 the proposed system with its parameter dynamics outperforms a static controller (with the parameters being in the same range) in a one dimensional hit-and-return task.

In the long-run experiments in section 3.3 it was shown that the parameter dynamics did not get stuck in the saturation region of the neurons and realizes the theoretically expected values. Also the activity of the agents does not decrease for a longer time, even in an environment crowded with other active agents.

Furthermore, the generation of coordinated behaviors is interesting in complex robotic devices. The experiments in section 3.4 showed the generation of crawling and jumping behaviors of a snake-like artifact. A differently realized snake-like robot was considered in section 3.5. In these experiments a whole-body motion of all, active as well as passive, elements of the under-actuated robot was generated by the homeokinetic control. Such modes of behavior are only possible by exploiting the physical properties of the robotic device, which are not known to the controller. Hence these experiments show the emergence of sensorimotor coordination under homeokinetic control.

In chapter 4 we presented extensions to the homeokinetic controller by a long-term memory and second-order learning, which allow to adapt to different situations or to shape the generated behaviors with respect to internally or externally available signals. The long-term memory (section 4.1) enabled the system to adapt to different situations without relearning the internal parameters. Instead parameter configurations for these situations were stored and recalled in similar situations, as qualified by context information. Furthermore the ability of the homeokinetic system to deal with the integration and disintegration of newly attached/removed sensors was shown for the case of a Khepera robot and it was demonstrated

that, if multiple sensors are available, those which show a reasonable response to the motor actions are predominantly integrated into the control loop. For the second order learning a higher control layer equipped with Hebbian learning was introduced and provided with context information (section 4.2). This way the behavior of the agent under control was shaped following internal available signals or an external specified goal, while the properties of the homeokinetic system could be retained. These results are especially interesting in the framework of developmental/epigenetic robotics (see for example the work of [Berthouze and Metta \(2005\)](#), [Zlatev and Balkenius \(2001\)](#)) to allow the robot to further “develop” to more complex behaviors or achieving of tasks by building on the primitive behaviors generated by the homeokinetic principle.

In chapter 5 the proposed approach to self-organized control of autonomous robots was applied to the control of myoelectric hand prostheses, where in a training period an amputee interacts with the prosthesis. In doing so he provides input signals to the homeokinetic controller via features calculated from the activities of the remaining muscles, measured with skin electrodes. The homeokinetic controller on the other hand controls the (simulated) prosthesis on the basis of this input signals and thus generates visual feedback for the amputee. Due to the property of the self-organizing controller to predominantly integrate sensors into the control loop which show a reasonable response to the motor actions, user-specific input features were selected in the interaction between subject and prosthesis. The self-organizing controller thus allows the selection of user-specific input features from a set of given features, which can later on be used in the everyday control of the prosthesis. The interaction period also encourages amputees to train the remaining muscles and hence improve the provided signals.

It is interesting to compare the presented results with related efforts from artificial evolution. From the point of view of dynamical complexity the robots presented in chapter 3 are close to creatures of the Framstick world ([Komosinski, 2000, 2005](#)) or also with some of Karl Sims’ creatures ([Sims, 1994](#)). The main difference is seen in the fact that the latter have been evolved for a specific behavior and the creatures can perform only correctly if situated in an appropriate environment. To the contrary the behaviors displayed by robots controlled by the homeokinetic principle adapt to the environment they are situated in.

Another example close to the work at hand is [Kuniyoshi et al. \(2003\)](#) which aims at the emergence of higher cognitive abilities from the physical dynamics of the robotic system and its sensorimotor interaction with the world. The parallel is at the level of motor learning where the authors demonstrate the need to acquire a set of explorative behaviors from scratch. This is exactly the task solved by the homeokinetic system.

In the work presented here only a linear internal model was used, which clearly restricts the predictive ability. This way only a linear response of the sensors is taken into account which favors the use of proprioceptive sensors inside the sensorimotor loop. For future work it would be a main point to extend the abilities of the model and hence allow an adaptation to complex situations and a better inclusion of different sensor modalities.

It would also be interesting to build more complex internal models in a higher control layer which are additionally provided with context information. Such a model, which is assumed to generate a good predictive ability of the whole system in many situations, may later on be used for reflections or to solve tasks or achieve goals provided from outside. The slow feature analysis especially ([Berkes and Wiskott, 2005](#); [Franzius et al., 2007a,b](#)) could

be used to develop navigational support, similar to complex cells in the primary visual cortex and place cells in the hippocampal areas respectively ([Thompson and Best, 1989](#); [Brun et al., 2002](#)), in a self-organized way. This would yield another advance in self-organized robot development.

Appendix A

Derivation of the Objective Function for Several Sensors

In the case of several channels the proposed principle needs some customizing since the input shift v , required for the objective function, equation 2.32, is not uniquely defined. We may remove the ambiguity by making an assumption on the direction of v , so we stipulate that v is in the direction of the response strength a given by the internal model with $v_t = u_t \hat{a}$ ($\hat{a} = \frac{a}{\|a\|}$; $\hat{a}^2 = \hat{a}^T \hat{a} = 1$). The reason behind this is that the inputs x should be produced by the deterministic part of the sensorimotor dynamics. Using the approximation $\xi_t = Lv_t$ we get

$$\xi_t = u_t L \hat{a}$$

and since the Jacobian $L = g'(z_t) a c^T$ ($L_{ij} = \frac{\partial}{\partial x_{j,t}} \psi_i(x_t) = a_i c_j g'(z_t)$) we get

$$\begin{aligned} \xi_t &= u_t g'(z_t) a c^T \hat{a} \\ &= u_t g'(z_t) a c^T a \frac{1}{\|a\|} \end{aligned}$$

Since the response strength $R = c^T a (= \sum_i c_i a_i)$ we can write

$$\xi_t = u_t g'(z_t) a R \frac{1}{\|a\|}$$

With $\hat{a} = \frac{a}{\|a\|}$ the right hand side changes to

$$\xi_t = \hat{a} u_t g'(z_t) R$$

To consider only the projection of ξ on a both sides are multiplied with \hat{a}^T

$$\hat{a}^T \xi_t = \hat{a}^T \hat{a} u_t g'(z_t) R$$

With $\hat{a}^T \xi_t = \|\xi_t\| \cos \phi$ and $\hat{a}^T \hat{a} = 1$ we get

$$\|\xi_t\| \cos \phi = u_t g'(z_t) R$$

The objective function for the case of several sensors is now

$$E = \|u_t\|^2$$

where

$$u_t = \frac{\|\xi_t\| \cos \phi}{g'(z_t)R}$$

Bibliography

- Ahissar, E. and Kleinfeld, D. (2003). Closed-loop neuronal computations: Focus on vibrissa somatosensation in rat. *Cerebral Cortex*, 13(1):53–62.
- Asada, M., MacDorman, K. F., Ishiguro, H., and Kuniyoshi, Y. (2001). Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems*, 37(2-3):185–193.
- Asano, F., Hashimoto, M., Kamamichi, N., and Yamakita, M. (2001). Extended virtual passive dynamic walking and virtual passivity-mimicking control laws. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 3139–3144, Seoul, Korea.
- Asano, F., Yamakita, M., and Furuta, K. (2000). Virtual passive dynamic walking and energy-based control laws. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 2, pages 1149–1154, Takamatsu, Japan.
- Augustsson, P., Wolff, K., and Nordin, P. (2002). Creation of a learning, flying robot by means of evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, pages 1279–1285, New York. Morgan Kaufmann.
- Bagnell, J. A. and Schneider, J. G. (2001). Autonomous helicopter control using reinforcement learning policy search methods. In *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, volume 2, pages 1615–1620. IEEE.
- Bekey, G. A. and Goldberg, K. Y., editors (1993). *Neural Networks in Robotics*. Kluwer, Boston, MA.
- Bellin, C. U. (2008). Fotos vom Hamburger Hafen. <http://www.fotograf-hamburg.de>.
- Bereiter, C. (1985). Towards a solution to the learning paradox. *Review of Educational Research*, 55(2):201–226.
- Berkes, P. and Wiskott, L. (2005). Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):579–602.
- Berthouze, L. and Metta, G. (2005). Epigenetic robotics: modelling cognitive development in robotic systems. *Cognitive Systems Research*, 6(3):189–192.

- Bessiere, P., Dedieu, E., and Mazer, E. (1994). Representing robot/environment interactions using probabilities: the "beam in the bin" experiment. *From Perception to Action Conference, 1994., Proceedings*, pages 158–169.
- Bi, G.-q. and Poo, M.-m. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24(1):139–166.
- Böhme, H.-J., Scheidig, A., Wilhelm, T., Schröter, C., Martin, C., König, A., Müller, S., and Gross, H.-M. (2006). Progress in the development of an interactive mobile shopping assistant. In *Proc. of the Joint Conf. on Robotics: 37th International Symposium on Robotics (ISR 2006) and 4th German Conference on Robotics (Robotik 2006)*, number 83, Munich, Germany. VDI Verlag.
- Boron, W. F. and Boulpaep, E. L. (2003). *Medical Physiology*. Elsevier.
- Bredenfeld, A., Jaeger, H., and Christaller, T. (2001). Mobile robots with dual dynamics. *ERCIM News*, 42.
- Brooks, R. (1986a). Achieving artificial intelligence through building robots. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts.
- Brooks, R. A. (1986b). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2, No1(1):14–23.
- Brooks, R. A. (1991). Intelligence without reason. pages 569–595. IJCAI.
- Brown, T. H., Kairiss, E. W., and Keenan, C. L. (1990). Hebbian synapses: Biophysical mechanisms and algorithms. *Annual Review of Neuroscience*, 13(1):475–511.
- Brun, V. H., Otnass, M. K., Molden, S., Steffenach, H.-A., Witter, M. P., Moser, M.-B., and Moser, E. I. (2002). Place Cells and Place Recognition Maintained by Direct Entorhinal-Hippocampal Circuitry. *Science*, 296(5576):2243–2246.
- Cannon, W. B. (1939). *The Wisdom of the Body*. Norton, New York.
- CC License (2008). Creative commons attribution-noncommercial-sharealike 2.5 license. <http://creativecommons.org/licenses/by-nc-sa/2.5/>.
- Citri, A. and Malenka, R. C. (2008). Synaptic plasticity: Multiple forms, functions, and mechanisms. *Neuropsychopharmacology Reviews*, 33:1841.
- Cliff, D., Husbands, P., and Harvey, I. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73–110.
- Der, R. (2000). Self-organized robot behavior from the principle of homeokinesis. In Groß, H.-M., Debes, K., and Böhme, H.-J., editors, *Workshop SOAVE-2000: Selbstorganisation von adaptivem Verhalten*, volume 10/643 of *Fortschrittsberichte VDI: Informatik/Kommunikationstechnik*. VDI-Verlag.

- Der, R. (2001). Self-organized acquisition of situated behaviors. *Theory in Biosciences*, 120:179–187.
- Der, R. (2003). Homeokinesis and the moderation of complexity in neural systems. <http://www.informatik.uni-leipzig.de/~der/Veroeff/leakint.pdf>. working paper.
- Der, R., Hesse, F., and Liebscher, R. (2004). Self-organized exploration and automatic sensor integration from the homeokinetic principle. In Groß, H.-M., Debes, K., and Böhme, H.-J., editors, *Proc. 3rd Workshop on Self-Organization of Adaptive Behavior (SOAVE'04)*, volume 10/743 of *Fortschrittsberichte VDI: Informatik/Kommunikationstechnik*, pages 220–230. VDI-Verlag.
- Der, R., Hesse, F., and Martius, G. (2005). Learning to feel the physics of a body. In *CIMCA '05: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06)*, pages 252–257, Washington, DC, USA. IEEE Computer Society.
- Der, R., Hesse, F., and Martius, G. (2006). Rocking stamper and jumping snake from a dynamical system approach to artificial life. *Adaptive Behavior*, 14(2):105–115.
- Der, R., Hesse, F., and Martius, G. (2008). Videos of self-organised robot behavior. <http://robot.informatik.uni-leipzig.de/Videos>.
- Der, R. and Liebscher, R. (2002). True autonomy from self-organized adaptivity. In *Proc. Workshop Biologically Inspired Robotics*, Bristol. Internet: <http://www.informatik.uni-leipzig.de/~der/Veroeff/bristol.pdf>.
- Der, R. and Martius, G. (2006). From motor babbling to purposive actions: Emerging self-exploration in a dynamical systems approach to early robot development. In Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J. C. T., Marocco, D., Meyer, J.-A., Miglino, O., and Parisi, D., editors, *From Animals to Animats 9, 9th International Conference on Simulation of Adaptive Behavior, SAB 2006, Rome, Italy, September 25-29, 2006, Proceedings*, volume 4095 of *Lecture Notes in Computer Science*, pages 406–421. Springer.
- Der, R. and Pantzer, T. (1999). Emergent robot behavior from the principle of homeokinesis. In Löffler, A., Mondada, F., and Rückert, U., editors, *Experiments with the Mini-Robot Khepera. Proceedings of the 1st International Khepera Workshop'99*, number 64 in Heinz-Nixdorf-Institut Verlagsschriftenreihe.
- Der, R., Steinmetz, U., and Pasemann, F. (1999). Homeokinesis - a new principle to back up evolution with learning. In *Computational Intelligence for Modelling, Control, and Automation*, volume 55 of *Concurrent Systems Engineering Series*, pages 43–47, Amsterdam. IOS Press.
- Dill, M., Wolf, R., and Heisenberg, M. (1993). Visual pattern recognition in drosophila involves retinotopic matching. *Nature*, 365:751–753.

- Eriksson, L., Sebelius, F., and Balkenius, C. (1998). Neural control of a virtual prosthesis. In Niklasson, L., Bodn, M., and Ziemke, T., editors, *Perspectives in Neural Computing: Proceedings of ICANN '98*. Berlin: Springer-Verlag.
- Farina, D., Merletti, R., and Enoka, R. M. (2004). The extraction of neural strategies from the surface EMG. *Journal of Applied Physiology*, 96:1486–1495.
- Ferezou, I., Haiss, F., Gentet, L. J., Aronoff, R., Weber, B., and Petersen, C. C. (2007). Spatiotemporal dynamics of cortical sensorimotor integration in behaving mice. *Neuron*, 56:907–923.
- Floreano, D. and Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(3):396–407.
- Fodor, J. A. (1980). Fixation of belief and concept acquisition. In Piatelli-Palmerini, M., editor, *Language and learning: The debate between Jean Piaget and Noam Chomsky*, pages 142–149, Cambridge, MA. Harvard University Press.
- Franceschini, N. (1997). Combined optical, neuroanatomical, electrophysiological and behavioural studies on signal processing in the fly compound eye. In Taddei-Ferretti, C., editor, *Biocybernetics of Vision: Integrative mechanisms and Cognitive Processes*, London. World Scientific.
- Franzius, M., Sprekeler, H., and Wiskott, L. (2007a). Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Comput Biol*, 3(8):e166.
- Franzius, M., Vollgraf, R., and Wiskott, L. (2007b). From grids to places. *Journal of Computational Neuroscience*.
- G. Schöner, M. D. and Engels, C. (1995). Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16:213–245.
- GNU GPL (2008). Gnu general public license, version 2. <http://www.gnu.org/licenses/gpl-2.0.html>.
- Goschin, S., Franti, E., Dascalu, M., and Osiceanu, S. (2007). Combine and compare evolutionary robotics and reinforcement learning as methods of designing autonomous robots. *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1511–1516.
- Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11(1):23–63.
- Günther, M. (1997). *Kontinuierliche und zeitdiskrete Regelungen*. Teubner, Stuttgart.
- Hamed, N. (2007). *Self-referential dynamical systems for the self-organization of behavior in robotic systems*. PhD thesis, Universität Leipzig.
- Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.
- Haykin, S. (1998). *Neural Networks A Comprehensive Foundation*. Prentice Hall.

-
- Hebb, D. O. (1949). *The organization of behavior; a neuropsychological theory*. Wiley, New York.
- Hesse, F., Der, R., and Herrmann, J. M. (2007). Reflexes from self-organizing control in autonomous robots. In Berthouze, L., Prince, C. G., Littman, M., Kozima, H., and Balke-nius, C., editors, *7th International Conference on Epigenetic Robotics: Modelling Cognitive Development in Robotic Systems, Rutgers University, Piscataway, NJ, USA*, volume 134 of *Cognitive Studies*, pages 37–44. Lund University.
- Hock, H. S., Schöner, G., and Giese, M. A. (2003). The dynamical foundations of motion pattern formation: Stability, selective adaptation, and perceptual continuity. *Perception & Psychophysics*, 65:429–457.
- Hopfield, J. J. (1984). Neurons with graded response have collective computational prop-erties like those of two-state neurons. *Proceedings of the National Academy of Sci-ences*, 81(10):3088–3092. <http://www.pnas.org/cgi/content/abstract/81/10/3088>.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH*, 4:237–285.
- Karlik, B., Osman Tokhi, M., and Alci, M. (2003). A fuzzy clustering neural network archi-tecture for multifunction upper-limb prosthesis. *Biomedical Engineering, IEEE Transac-tions on*, 50:1255–1261.
- Komosinski, M. (2000). The world of framsticks: simulation, evolution, interaction. In *Proceedings of the 2nd International Conference on Virtual Worlds (VW2000)*, Lecture Notes in Artificial Intelligence 1834, pages 214–224, Paris, France. Springer-Verlag.
- Komosinski, M. (2005). Computer science in evolutionary biology and robotics: Framsticks artificial life.
- Kuniyoshi, Y. and Sangawa, S. (2006). Early motor development from partially ordered neural-body dynamics: experiments with a cortico-spinal-musculo-skeletal model. *Bio-logical Cybernetics*, 95(6):589–605. DOI: 10.1007/s00422-006-0127-z.
- Kuniyoshi, Y., Yorozu, Y., Ohmura, Y., Terada, K., Otani, T., Nagakubo, A., and Yamamoto, T. (2003). From humanoid embodiment to theory of mind. In *Embodied Artificial Intelli-gence*, pages 202–218.
- Kuppuswamy, N. S., Cho, S.-H., Stonier, D., Choi, S.-L., and Kim, J.-H. (2006). Design of an omnidirectional robot for fira robosot. In *2006 FIRA Robot World Congress*, Dortmund, Germany.
- Leigh, J. R. (2004). *Control Theory*. IET.
- Lenser, S. (2005). *On-line Robot Adaptation to Environmental Change*. PhD thesis, School of Computer Science, Computer Science Department, Carnegie Mellon University, Pitts-burgh, PA 15213.

- Liao, S.-H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications*, 28(1):93–103.
- Lippert, H. (1995). *Anatomie: Text und Atlas*. Urban und Schwarzenberg, München, Wien, Baltimore, 6 edition.
- Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978.
- Martius, G., Hesse, F., and Der, R. (2008). lpzrobots. <http://robot.informatik.uni-leipzig.de/software/>.
- McGeer, T. (1990). Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82.
- Meiss, J. (2007). Dynamical systems. *Scholarpedia*, 2(2):1629. revision # 37019.
- Merritt, E., Broeker, H.-B., Campbell, J., Cunningham, R., Denholm, D., Elber, G., Fearick, R., and et al., C. G. (2008). gnuplot. <http://www.gnuplot.info/>.
- Michel, O. (1995). Khepera simulator. <http://diwww.epfl.ch/lami/team/michel/khep-sim/>.
- Mitchinson, B., Martin, C. J., Grant, R. A., and Prescott, T. J. (2007). Feedback control in active sensing: rat exploratory whisking is modulated by environmental contact. *Proceedings of the Royal Society B: Biological Sciences*, 274(1613):1035–1041.
- MobileRobots Inc (2008). <http://www.mobilerobots.com>.
- Mondada, F., Franzi, E., and Ienne, P. (1994). Mobile robot miniaturisation: A tool for investigation in control algorithms. In Yoshikawa, T. and Miyazaki, F., editors, *Proceedings of the Third International Symposium on Experimental Robotics*, volume 200 of *Lecture Notes in Control and Information Sciences*, pages 501–513. Springer Berlin / Heidelberg.
- Morimoto, J. and Doyab, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, 36(1):37–51.
- National Instruments (2008a). DAQCard-6036E. <http://www.ni.com>.
- National Instruments (2008b). LabVIEW. <http://www.ni.com/labview/>.
- Nelson, A. L., Grant, E., Galeotti, J. M., and Rhody, S. (2004). Maze exploration behaviors using an integrated evolutionary robotics environment. *Robotics and Autonomous Systems*, 46(3):159–173.
- Neuroinformatics and Cognitive Robotics Lab (2007). Shoppingroboter kommt im Baumarkt an! <http://www.tu-ilmeneu.de/neurob/Toomas.6483.0.html?&L=1>. Ilmenau University of Technology.
- Nguyen, Q.-T. and Kleinfeld, D. (2005). Positive feedback in a brainstem tactile sensorimotor loop. *Neuron*, 45(3):447–457.

- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The biology, Intelligence and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA.
- Nolfi, S. and Floreano, D. (2002). Synthesis of autonomous robots through evolution. *Trends in Cognitive Sciences*, 6(1):31–37.
- Nolfi, S. and Marocco, D. (2002). Active perception: A sensorimotor account of object categorization. In Hallam, B., Floreano, D., Hallam, J., Hayes, G., and Meyer, J.-A., editors, *From Animals to Animats 7*, Proceedings of the VII International Conference on Simulation of Adaptive Behavior, pages 266–271, Cambridge. MIT Press.
- Osfield, R., Burns, D., Jez, M., Weiblen, M., Wing, E., Johansen, B., Michel, G., Martz, P., Lashkari, F., and et al., S. H. (2008). Open scene graph. <http://www.openscenegraph.org>.
- Ott, E. (1993). *Chaos in Dynamical Systems*. Cambridge University Press.
- Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. The MIT Press, Cambridge, Massachusetts; London, England.
- Popp, J. (2008). <http://www.sphericalrobots.org>.
- Rapaport, W. J. (1986). Searle’s experiments with thought. *Philosophy of Science*, 53(2):271–279.
- Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer.
- Rojas, R. (2005). Omnidirectional control. <http://www.inf.fu-berlin.de/lehre/SS05/Robotik/omnidriveNew.pdf>.
- Rojas, R. and Förster, A. G. (2006). Holonomic control of a robot with an omnidirectional drive. *KI - Künstliche Intelligenz*, 20(2):12–17.
- Rosheim, M. E. (1996). Leonardo’s lost robots. *Achademia Leonardi Vinci. Journal of Leonardo Studies and Bibliography of Vinciana*, IX:99110.
- Rumelhart, D. E. and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, 9(1):75 – 112.
- Salomon, R. (1996). Increasing adaptivity through evolution strategies. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S., editors, *From Animals to Animates 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 411–420, Cambridge. The MIT Press/Bradford Books.
- Schöner, G., Dose, M., and Engels, C. (1995). Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2-4):213–245.
- Searle, J. (1980). Minds, brains and programs. *Behavioral and Brain Sciences*.

- Sears, H. H. and Shaperman, J. (1991). Proportional myoelectric hand control: an evaluation. *American Journal of Physical Medicine and Rehabilitation*, 70(1):20–28.
- Sears, H. H. and Shaperman, J. (1998). Electric wrist rotation in proportional-controlled systems. *Journal of Prosthetics and Orthotics*, 10:92–97.
- Sharkey, N. (2007). The programmable robot of ancient greece. *New Scientist*.
- Sims, K. (1994). Evolving virtual creatures. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM Press.
- Smith, R. (2008). Open dynamics engine. <http://ode.org/>.
- Steffe, L. P. (1991). The learning paradox: A plausible counterexample. In Steffe, L. P., editor, *Epistemological foundations of mathematical experience*, pages 26–44, New York. Springer.
- Steinhage, A. (1997). *Dynamical Systems for the Generation of Navigation Behavior*. PhD thesis, Ruhr-Universit at Bochum, Germany.
- Strogatz, S. H. (2001). *Nonlinear Dynamics and Chaos*. Westview Pr.
- Sutton, R. S. (1992). Introduction: The challenge of reinforcement learning. *Machine Learning*, 8(3-4):225–227.
- Sutton, R. S. and Barto, A. G. (1998). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA.
- Tamosiunaite, M., Ainge, J., Kulvicius, T., Porr, B., Dudchenko, P., and Wörgötter, F. (2008). Path-finding in real and simulated rats: On the usefulness of forgetting and frustration for navigation learning. *J. Comp. Nsci.*, submitted.
- Tani, J. (2004). Symbols and dynamics in embodied cognition: Revisiting a robot experiment. In Butz, M. V., Sigaud, O., and Gerard, P., editors, *Anticipatory Behavior in Adaptive Learning Systems*, pages 167–178. Springer-Verlag.
- Tani, J. and Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 33(4):481–488.
- Tedrake, R., Zhang, T. W., Fong, M.-f., and Seung, H. S. (2004). Actuating a simple 3d passive dynamic walker. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 5, pages 4656–4661.
- Thompson, L. and Best, P. (1989). Place cells and silent cells in the hippocampus of freely-behaving rats. *J. Neurosci.*, 9(7):2382–2390.
- Thrun, S. (2004). Towards a framework for human-robot interaction. *Human-Computer Interaction*, 19:9–24.

- Torras, C. (1995). Robot adaptivity. *Robotics and Autonomous Systems*, 15:11–23.
- Torras, C. (2002). Neural computing increases robot adaptivity. *Natural Computing*, 1:391–425.
- Toyota Motor Corporation (2007). Partner robots. http://www.toyota.co.jp/en/vision/emerging_tech/p_robot, <http://www.toyota.co.jp/en/special/robot/>.
- von Glasersfeld, E. (1998). Scheme theory as a key to the learning paradox. Paper presented at 15th advanced course, archives Jean Piaget, Geneva, Switzerland.
- Warwick, K. (1996). *An Introduction to Control Systems*. World Scientific.
- Wiener, N. (1961). *Cybernetics, or Control and Communication in the Animal and the Machine*. The M.I.T. Press, Cambridge, Massachusetts, 2 edition edition. first edition from Wiley, 1948.
- Wikimedia Commons (2008). Industrial robot for welding or handling from KUKA Robotics. http://commons.wikimedia.org/wiki/Image:Industrial_Robot_.jpg.
- Willburger, H. (2007). *Adaptive Myo-Signal Processing For Multidimensional Prosthesis Control*. PhD thesis, Vienna University of Technology.
- Wilson, L., Williams, C., Yance, J., Lew, J., II, R. W., and Gallina, P. (2001). Design and modeling of a redundant omni-directional robocup goalie. In *RoboCup AI Conference*, Seattle, WA, USA.
- Woergoetter, F. and Porr, B. (2008). Reinforcement learning. *Scholarpedia*, 3:1448. revision # 38779.
- Wolpert, D. M. and Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience*, 3(11):1212–1217.
- Zell, A. (1997). *Simulation neuronaler Netze*. Oldenbourg.
- Zhou, C. and Meng, Q. (2003). Dynamic balance of a biped robot using fuzzy reinforcement learning agents. *Fuzzy Sets Syst.*, 134(1):169–187.
- Zlatev, J. and Balkenius, C. (2001). Introduction: Why epigenetic robotics? In Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., and Breazeal, C., editors, *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, number 85, pages 1–4. Lund University Cognitive Studies.
- Zucker, R. S. and Regehr, W. G. (2002). Short-term synaptic plasticity. *Annual Review of Physiology*, 64:355–405.
- Zunt, D. (2007). Karel Čapek website. <http://capek.misto.cz/english/interesting.html>.

Danksagung

Zunächst möchte ich mich bei Ralf Der bedanken, der mich für das Thema der Selbstorganisation autonomer Roboter begeistert und somit den Anlaß für diese Arbeit gegeben hat. Auch für seine Unterstützung in vielerlei Hinsicht sei ihm gedankt.

Theo Geisel danke ich für seine stete Unterstützung, ohne die diese Arbeit nicht möglich gewesen wäre. Weiterhin danke ich ihm für die Möglichkeit in seiner Gruppe eine großartige Wissenschafts- und Arbeitsatmosphäre erleben zu können.

Michael Herrmann sei gedankt für seine Betreuung dieser Arbeit, die gute Zusammenarbeit und Unterstützung, interessante Diskussionen und wichtige Hilfestellungen auf dem Weg.

Bei formalen und organisatorischen Fragen und Problemen war es eine Erleichterung auf die Hilfe von Tobias Niemann bauen zu können. Ebenso möchte ich mich bei den Sekretärinnen, insbesondere bei Regina Wunderlich, bedanken.

Auch der gesamten Arbeitsgruppe möchte ich für die gute Atmosphäre und Unterstützung danken, ohne allerdings namentlich ins Detail gehen zu wollen. Eine Ausnahme bilden allerdings Georg Martius, Joachim Haß und Anna Martius. Georg gilt ein besonderer Dank für die gute Zusammenarbeit, Unterstützung, Hilfsbereitschaft und vieles mehr. Joachim und Anna danke ich für die angenehme Art mit mir den Büroalltag zu bestreiten.

Für die Unterstützung sowie anregende Diskussionen und Impulse aus dem breiten Spektrum an wissenschaftlichen Arbeiten im Netzwerk des Bernstein Center for Computational Neuroscience möchte ich mich bedanken, so auch bei der Arbeitsgruppe von Florentin Wörgötter.

Ein weiterer Dank gilt allen Kooperationspartnern und Kollegen von ausserhalb, wie René Liebscher, Naglaa Hamed, Horst Willburger, dem Institut für Intelligente Systeme der Universität Leipzig und den Abteilungen für strategisches Technologiemanagement der Otto Bock HealthCare GmbH Wien und Duderstadt.

Julius Popp möchte ich dafür danken die Sichtweise auf die Selbstorganisation um den künstlerischen Aspekt bereichert zu haben.

Ich möchte mich auch bei meiner Familie bedanken ohne die diese Arbeit nicht möglich und ich nicht hier wäre. Ebenso geht ein riesengroßes Dankeschön an Anja, die viel an dieser Arbeit mitzutragen hatte.



Niemand ist der Herr seines Weges, und kein Mensch hat die Macht, den Gang seiner Schritte zu bestimmen. (Jer. 10,23)

Dank sei dem Herrn. Für Alles.

This work was supported by the BMBF in the framework of the Bernstein Centers for Computational Neuroscience, grant number 01GQ0432.