Modelling closed-loop receptive fields: On the formation and utility of receptive fields in closed-loop behavioural systems

Dissertation

ZUR ERLANGUNG DES MATHEMATISCH-NATURWISSENSCHAFTLICHEN DOKTORGRADES
"DOKTOR RERUM NATURALIUM" DER GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

vorgelegt von

Tomas Kulvicius

aus Kaunas, Litauen

Göttingen 2010

Referentin/Referent: Prof. Dr. Florentin Wörgötter
Koreferentin/Koreferent: Prof. Dr. Jens Grabowski
Tag der mündlichen Prüfung: 20/04/2010

Zusammenfassung

Bei höher entwickelten Tieren nimmt die Komplexität der visuellen rezeptiven Felder mit dem hierarchischen Aufbau von den visuellen Eingangsarealen zu den höheren Hirnarealen in dem Maße zu, dass visuelle Reize in den höheren Hirnarealen einen indirekteren Einfluss als in den Eingangsarealen ausüben. Von diesen Arealen aus gibt das System diese Aktivität dann wieder über weitere Stufen an die Endeffektoren (Muskeln) weiter. Neuere Erkenntnisse zeigen jedoch, dass bei einfacheren Tieren, beispielsweise Fliegen, ein Motorneuron über ein visuelles rezeptives Feld verfügen kann (Krapp und Huston, 2005) und das Motorneuron dadurch entsprechende sensorische Merkmale aufweisen kann. Solche rezeptiven Felder beeinflussen das Verhalten in direkter Weise, indem diese Neuronen ohne Zwischenschritte direkt die Wahrnehmungs-Handlungs-Schleife schließen und Feedback über die Umwelt wieder an die Sensoren geben.

Im ersten Teil dieser Doktorarbeit werden wir aufzeigen, dass es möglich ist, solche direkt gekoppelten Sensor-Motor-Felder in einfachen Verhaltenssystemen mit Hilfe eines auf Korrelationen basiertem Temporal-Sequence-Learning-Algorithmus zu entwickeln. Das Hauptziel besteht darin aufzuzeigen, dass Lernen stabiles Verhalten generiert und dass die erzeugten rezeptiven Felder sich ebenfalls stabilisieren, wenn das neuerlernte Verhalten erfolgreich ist. Die Entwicklung von stabilen neuronalen Eigenschaften als auch von stabilem Verhalten ist schwierig, da die Konvergenz von funktionalen Neuroneneigenschaften und vom Verhalten gleichzeitig sichergestellt werden muss. Diese Arbeit stellt einen ersten Versuch dar, dieses Problem mit Hilfe eines einfachen Robotorsystems zu lösen. Dieser Teil der Arbeit wird mit der Frage geschlossen, wie eine indirekte Sensor-Motor-Kopplung, wie sie bei höher entwickelten Tieren vorkommt, aufgebaut werden kann. Durch die Nutzung von einfachen hintereinandergeschalteten Lernstrukturen werden wir aufzeigen, dass damit ähnliche Ergebnisse erzielt werden können; selbst für die sekundären rezeptiven Felder, die indirekte visuelle Reize erhalten.

Im zweiten Teil dieser Doktorarbeit werden wir verschiedene Agenten quantitativ analysieren, die sich mit dem im ersten Teil vorgestelltem Temporal-Sequence-Learning-Algorithmus an die Umwelt anpassen. Hierbei werden wir versuchen folgende Frage zu beantworten: Wie können wir vorhersagen, welcher der gegebenen Agenten sich am besten für ein bestimmtes Szenario (Umwelt) eignet? Direkt gekoppelte Umwelt-Agent-Systeme in ihrem Verhalten zu verstehen, stellt kein triviales Problem dar, vor allem wenn sich diese Systeme im Lernprozess verändern. Geschlossene Regelschleifen, wie das Umwelt-Agent-System, wurden in der Informationstheorie bereits in den 50er Jahren beschrieben, dennoch gab es nur wenige Versuche Lernen mitzuberücksichtigen, wobei meist der Informationsgehalt der Eingangsgrößen gemessen wurde. Zur Beantwortung der oben genannten Frage werden

wir mit Hilfe von Energie- und Entropiemessungen simulierte Agenten untersuchen und deren Entwicklung im Lernprozess beobachten. So kann nachgewiesen werden, dass es in genau definierten Szenarien lernende Agenten gibt, die in Bezug auf ihren Aufbau und ihr Anpassungsvermögen optimale Eigenschaften aufweisen. Darüber hinaus werden wir aufzeigen, dass es im Rahmen von vergleichsweise einfachen Fällen analytische Lösungsmöglichkeiten für die zeitliche Entwicklung solcher Agenten gibt.

In den ersten beiden Teilen der Arbeit werden Agenten mit unimodalem sensorischen Eingang analysiert (visuell oder somatosensorisch). Im dritten und letzten Teil dieser Arbeit wird untersucht, inwieweit der Einsatz von multimodalen Sensoren die Entwicklung der rezeptiven Felder und des Verhaltens beeinflusst. Dieser Ansatz geht auf Experimente mit Nagetieren zurück, in denen nachgewiesen werden konnte, dass, obwohl visuelle Reize für die Entstehung von hippocampischen Ortszellen ("place cells") und der räumlichen Orientierung eine wichtige Rolle spielen, Ratten sich auch auf olfaktorische, auditive und somatosensorische Reize sowie solche aus ihrer Eigenbewegung stützen können. In dieser Doktorarbeit wird erstmalig ein Modell einer Ortszelle vorgestellt, in dem visuelle und olfaktorische Reize zur Herausbildung eines Ortsfeldes kombiniert werden. Dies wird durch ein einfaches Feed-Forward-Netzwerk und einem Winner-Takes-All-Lernmechanismus realisiert. Eine Orientierungsaufgabe wird mit Hilfe der vorgestellten Orientierungsmechanismen, basierend auf geruchliche Eigenmarkierungen, die mit einem Q-Lernalgorithmus kombiniert werden, gelöst. Wir zeigen, dass olfaktorische Reize eine wichtige Rolle bei der Bildung von Ortsfeldern darstellen und weisen nach, dass eine Kombination von visuellen und olfaktorischen Reizen, die mit einer gemischten Orientierungsstrategie einhergeht, zu einer Verbesserung der zielgerichteten Orientierung führt.

Contents

Citations to Related Publications

Large portion of Chapter 2 is based on the following three papers:

> Kulvicius, T., Porr, B. and Wörgötter, F. (2007). Chained learning architectures in a simple closed-loop behavioural context. *Biological Cybernetics*, 97(5), 363-378;
>
> Kulvicius, T., Kolodziejski, C., Tamosiunaite, M., Porr, B. and Wörgötter, F. Behavioral analysis of differential hebbian learning in closed-loop systems. *Biological Cybernetics*, accepted for publication.
>
> Kulvicius, T., Porr, B. and Wörgötter, F. (2007). Development of receptive fields in a closed-loop behavioural system. *Neurocomputing*, 70(10-12), 2046-2049.

Finally, Chapter 4 appears in its entirety as

> Kulvicius, T., Tamosiunaite, M., Ainge, J., Dudchenko, P. and Wörgötter, F. (2008). Odor supported place cell model and goal navigation in rodents. *Journal of Computational Neuroscience*, 25(3), 481-500.

Acknowledgments

First of all I would like to thank my supervisors Prof. Dr. Florentin Wörgötter and Dr. Minija Tamosiunaite for guiding me through my work by sharing their experiences with me and for countless hours of fruitful discussions without which this work would not have been successful. This work has been done in collaboration with Dr. Bernd Porr, Dr. Christoph Kolodziejski, Dr. Paul Dudchenko and Dr. James Ainge, so I am very thankful for their efforts, too.

Secondly, I would like to thank all my colleagues and friends for their direct and/or indirect input to my work and a having great time together. Many thanks go to Ausra Saudargiene, Sinan Kalkan, Nicolas Pugeault, Tao Geng, Matthias Hennig, Marina Wimmer, Ailsa Millen, Ursula Hahn-Wörgötter, Steffen Wischmann, Alexander Wolf, Poramate Manoonpong, Irene Markelic, Babette Dellen, Markus Butz, Natalia Shyllo, Christian Tetzlaff, Kristin Stamm, Silke Steingrube, Daniel Steingrube, Johannes Schröder-Schetelig, Harm-Friedrich Steinmetz, Alexey Abramov, Eren Erdal Aksoy, Liu Guo Liang, KeJun Ning, Rokas Sabaliauskas, Jan-Matthias Braun, Thomas Wanschik, Johannes Dörr, Waldemar Kornewald, Visvaldas Seskus, Ricardas Maciulis, Andrius Kasuba, Ausra Mackute-Varoneckiene, Audrius Varoneckas and Andrius Balciunas.

A special thanks goes to my father Kestutis and my mother Marija without whom I would not have achieved all that in my life what I have now. And finally, I would like to thank my wife Ingrida for the patience, understanding, support and being always by my side no matter what. Thank you very much indeed!

Dedicated to my father Kestutis,
my mother Marija,
and my wife Ingrida.

1 Introduction

In control theory systems are often classified into two groups: 1) Open-loop systems and 2) Closed-loop systems. Open-loop systems are systems in which the output is not used as a control variable. Since there is no feedback used to control the system, open-loop systems can not cope with unexpected situations. For example, imagine we are driving a car on very well known road and we close our eyes for a short moment of time and then some creature suddenly enters the road. Evidently, the disrupted visual feedback prevents us from reacting. While this is clearly dangerous, many examples exist in biology for such feed-forward open-loop behaviour, too, such as ballistic movements, i.e. a forced movement initiated by muscle actions (such as a tennis serve or boxing punch), or ballistic stretching, i.e. a quick, bouncing movement that often take a joint beyond its normal range (usually it is painful). The advantage of such movements is that they are very fast. The lack of control therein, however, normally leads to the situation that behaving systems form a closed-loop with their environment where sensory inputs influence motor output, which in turn will create different sensations. Let's get back to our example of driving a car on a curvy road. In this example the view of the curve segment generates visual input to the system and steering is one possible output. Clearly, our perception of the road (steepness of the curve) influences how much we have to steer, whereas turning the steering wheel will cause changes in our perception for the next time moment. Visually guided reaching and grasping, navigation in the environment, servoing in robots are also examples of such closed-loop systems. Different from open-loop systems, closed-loop systems can react to unexpected situations and/or adapt to environmental changes by ways of learning.

In this thesis we will investigate closed-loop learning systems where the emphasis is on the development and utility of receptive fields in a closed-loop behavioural context. A receptive field (RF) of a given neuron is that particular surface area of a sensory organ from which neuronal responses can be elicited. Or in other words, the collection of sensors which form synapses to a single neuron form the neuron's receptive field. For example, the RF of a ganglion cell in the retina of the eye is composed of inputs from photoreceptors which provide its input, whereas a group of ganglion cells in turn forms the RF for a cell in the brain (Kandel et al., 2000). Receptive fields are found

in different brain regions such as visual, somatosensory and auditory cortex.

Another type of receptive fields are place fields (PFs) found in rat hippocampus (O'Keefe and Dostrovsky, 1971). Place fields of pyramidal cells code for a specific location of the animal in its environment. Like other receptive fields, PFs are formed from sensory inputs but differ from conventional RFs in that PFs are formed from multiple sensory cues such as visual, olfactory, somatosensory, auditory and self-motion cues (Knierim et al., 1995; Save et al., 1998, 2000; Hill and Best, 1981; Etienne and Jeffery, 2004).

There have been different methods proposed for the development of visual receptive fields in the visual cortex (Olshausen and Field, 1996; Bell and Sejnowski, 1997; Blais et al., 1998; Weber and Obermayer, 1999; Hurri and Hyvärinen, 2003; Körding et al., 2004; Wyss et al., 2006). However, in these studies the output of the receptive fields is not used to control behaviour (open-loop system). On the other hand, there exist studies which use receptive fields (place fields) for spatial navigation. However, in these studies place fields are first developed in an exploration phase and only afterwards used for goal directed learning (Arleo and Gerstner, 2000; Arleo et al., 2004; Strösslin et al., 2005; Sheynikhovich et al., 2005). The novelty of our approach is that we *simultaneously* develop and use receptive fields in behavioural tasks as shown in Fig. 1.1 creating a closed-loop scenario. We form receptive fields from sensory inputs where at the same time RFs are used to drive the behaviour of the agent. When acting in the environment, sensory inputs change, which in turn influence the formation of the receptive fields closing the loop. In one approach (presented in Chapter 2) we will directly use receptive fields for the driving behaviour of a robot, whereas in the other system (presented in Chapter 4) receptive fields will be used as an input to the upper layer (motor neurons) in the network for path learning. Note that here development of RFs and path learning will be performed simultaneously.

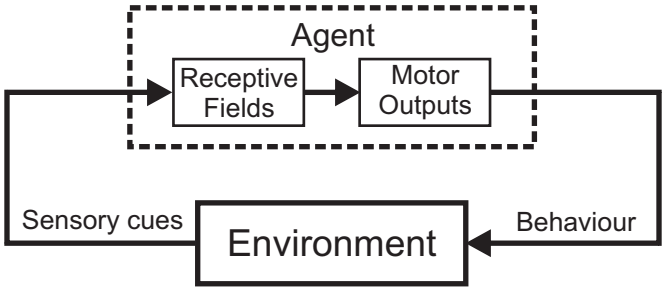

Figure 1.1: Schematic diagram of the development and utility of receptive fields in a closed-loop behavioural context.

This thesis divides into three parts. The first part is devoted to the learning in a sensory-motor loop and the development of primary and secondary "indirect"

receptive fields. In the second part we will be concerned with the quantitative analysis of closed-loop learning systems, whereas in the last part we will investigate multi-sensor integration for receptive field development and its influence onto behavioural performance.

In Chapter 2 we will present simple as well as chained learning architectures and show the development of visual receptive fields by using temporal sequence learning. By implementing simple chained learning architectures we will for the first time generate and stabilise secondary receptive fields in a closed-loop behavioural context. Here by secondary RFs we mean the development of receptive fields in higher layers of chained architectures which receive "indirect" inputs from lower layers.

Further on, in Chapter 3, we will investigate closed-loop learning systems which perform temporal sequence learning as presented in the first part in a more generic way by analysing aspects of system dynamics during learning. To our knowledge this is one of the first attempts to address such issues in closed-loop systems during learning.

And finally, in Chapter 4 we will present a navigation system based on place fields, where for the first time we will show the importance of the integration of multi-modal cues for place field formation and goal directed navigation. We will also present a novel navigation mechanism based on self-marking which makes the learning process even faster.

Each chapter starts with its own Introduction section, where we discuss the state of the art and our goals related to that topic, and ends with Discussion section where we compare our approach to other methods and relate it to biological data. We will conclude this thesis with Chapter 5 where we will summarize all main findings and provide an outlook for future investigations.

2

Behaviourally Guided Development of the Receptive Fields

2.1 Introduction

Normally many sensor events, which follow each other in time, are associated to a real life situation. However, only a few can be used to improve the behaviour. This can be achieved by temporal sequence learning. It rests on the assumption that it is in most cases advantageous to react to the earliest of such sensor events not having to wait for later following ones. For example, it is useful to react to a heat radiation signal and not to the later following pain on having finally touched a hot surface. Many similar sequences of sensor events are encountered during the life time of a creature as the consequence of the existing far-senses (e.g.: vision, hearing, smell) and near-senses (touch, taste, etc.). Generically one observes that the trigger of a near-sense is preceded by that of a far sense (smell precedes taste, vision precedes touch, etc.). Far-senses act predictive with respect to the corresponding near-senses (Verschure and Coolen, 1991). Conceptionally this type of learning is related to classical and/or operant conditioning (Sutton and Barto, 1981, 1990; Wörgötter and Porr, 2005). Algorithmically all these approaches (Sutton and Barto, 1981; Kosco, 1986; Klopf, 1988; Porr and Wörgötter, 2003a) share the property that they are built in a very simple way, in general only consisting of a single learning unit.

The development of visual receptive fields, for example in the primary visual cortex, has been an intriguing problem addressed in numerous studies (Olshausen and Field, 1996; Bell and Sejnowski, 1997; Blais et al., 1998; Weber and Obermayer, 1999; Hurri and Hyvärinen, 2003; Körding et al., 2004; Wyss et al., 2006). However, in these studies the receptive field output does not change the actual behaviour. This means that these learning algorithms operate in open-loop or as so called input/output systems. Evidence, however, exists that visual receptive fields can indeed be influenced by the behavioural context on quite different time-scales (Sugita, 1996; Dragoi et al., 2003). Indeed there is one recent study that is able to generate receptive fields in a behaviourally closed loop context (McKinstry et al., 2006) but it remains unclear if these fields are stable over time (see section 2.13).

Furthermore, one can ask the question, how higher order receptive fields, like those in visual areas beyond V1 are generated where the input becomes more and more indirect and neurons receive their vision information conveyed by several intermediate stages? In our context the question can be rephrased asking: Can we chain our learning architectures and still arrive at a stable behaving system, which also generates stable receptive fields?

Here we will apply temporal sequence learning to a driving robot that is supposed to learn to better follow a line painted on the ground. We will demonstrate: 1) That it is possible with such architectures to generate "receptive fields" from sensory inputs. 2) That the output of these RFs can drive the motors of the robot in order to create better and more stable behaviour, (which in turn influences its sensor inputs) and 3) that RF development will stop as soon as the system has obtained behavioural stability after learning. Furthermore we will show (4) that it is possible to design simple chains of such learning units while at the same time still guaranteeing behavioural stability, and that such architectures outperform simple architecture in cases where we have only weakly correlated (in time) inputs. We will also demonstrate (5) that secondary receptive fields can be developed by using simple chained architectures. The central goal of this approach is to demonstrate that direct sensor-motor coupling in a very simple architecture can lead to the generation of stable structural elements and simultaneously to stable behaviour without additional assumptions, while it is possible to gradually extend such architectures towards lattices without the need for additional free parameters.

The chapter is organised in the following way. After presenting the sequence learning rule called "ICO" (Input Correlation learning, Porr and Wörgötter, 2006) and its embedding into a closed loop scenario we will first discuss some setups without receptive fields. By this we would like to demonstrate the efficiency and stability of the ICO-rule in the line-following task using high learning rates. Next we start to look at receptive field development and analysis, which requires lower learning rates without which fine structure would not develop. Later on we will present two simple chained learning architectures and show results of receptive field development by using such chained architectures. Finally, we will conclude this chapter with a discussion section.

2.2 Experimental setup

2.2.1 Robot setup

A small two-wheeled Rug Warrior Pro mobile robot (diameter of 18 cm) was used for investigation which was tested on a line following task as shown in Fig. 2.1 A. The robot has built in camera which produces images of the track and is driven by two DC motors. The robot was connected to the desktop PC via cables. DA/AD converter

board USB-DUX[1] was used for receiving visual input signals from the robot and for sending motor output signals to control the robot. The sampling rate of the system was 25 Hz.

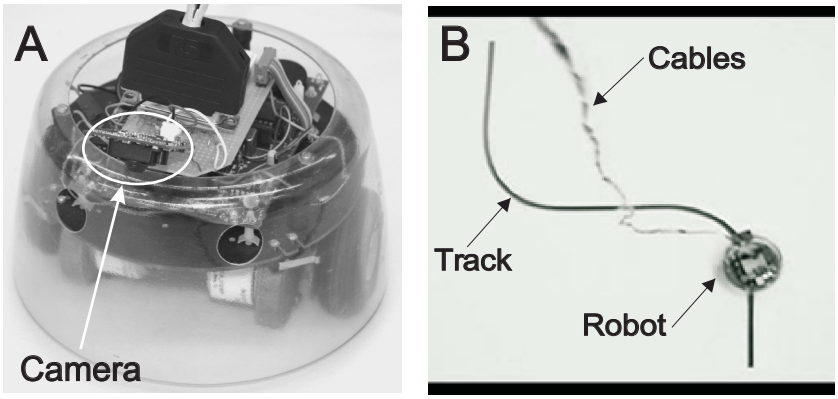

Figure 2.1: **A)** Image of the Rug Warrior Pro mobile robot. **B)** Image of the robot performing a line following task.

2.2.2 Learning task

The task for the robot was to learn following a black line painted on a white floor. Initially there is implemented only a weak, abrupt and late steering reflex which in most of cases (except of very shallow turns) will not be sufficient to steer the curve. As a consequence the robot would looses the track. The learning goal is to learn predictive and smoother steering reactions in order to stay on the track and to avoid the initial reflex.

2.3 Learning algorithm

The temporal sequence learning rule based on spike timing dependent plasticity (STDP) was used for learning (Porr and Wörgötter, 2006). The general scheme of such learning algorithm is presented in Fig. 2.2 B. The learner has inputs x_j which feed into a summation unit v. The output is calculated by

$$v = \sum_j \rho_j u_j, \tag{2.1}$$

[1]For more information please visit the web-page: *http://www.linux-usb-daq.co.uk*

where $u = h * x$ is a temporal convolution of the input x with a low-pass filter h. We define the low-pass filter by

$$h(t) = \frac{1}{b} e^{at} \sin(bt), \tag{2.2}$$

where, $a = -\pi f/Q$ and $b = \sqrt{(2\pi f)^2 - a^2}$, with f the frequency and $Q > 0.5$ the damping. This convolution correlates temporally non-overlapping signals x_1 and x_0 as shown in Fig. 2.2 C.

The learning unit receives its reflexive x_0 and predictive x_1 inputs from the sensors fields (line detectors) $x_0^{L,R}$ and $x_1^{l,R}$ respectively in the image of a forward pointing camera on the robot as shown in Fig. 2.2 A. Sensors fields $x_0^{L,R}$ are located at the bottom in the camera image whereas sensor fields $x_1^{L,R}$ are place higher up from the reflex. As a consequence the time delay T between x_1 and x_0 depends on the speed of the robot and direction angle with respect to the curvature. To accommodate some variability, x_1 is fanned out and fed into a filter-bank of different filters h as indicated by the dashed lines in Fig. 2.2 C. As shown in older studies of Porr and Wörgötter (2003a, 2006), the number of filters k is not critical and here $k = 10$ was used. The robot's base speed of 0.125 m/s together with the camera frame rate of 25 Hz used in all experiments leads to $f_{1,k} = 2.5/k\ Hz$, $k = 1, \ldots, 10$ for the filter-bank in the x_1 pathway. Frequency of the x_0 pathway was $f_0 = 1.25\ Hz$. Damping parameter of all filters was $Q = 0.6$.

Weights change according to an input-input correlation (ICO) rule (Porr and Wörgötter, 2006):

$$\dot{\rho}_j = \mu u_j \dot{u}_0, \quad j > 0, \tag{2.3}$$

which is a modification of the isotropic sequence order (ISO) learning rule (Porr and Wörgötter, 2003a). The behaviour of this rule and its convergence properties are discussed in (Porr and Wörgötter, 2006).

The weight ρ_0 is set to a fixed value ($\rho_0 = 1$), all other weights are initially zero. As discussed above this learning rule is specifically designed for a closed loop system where the output of the learner v feeds back to its inputs x_j after being modified by the environment (see Fig. 2.3).

The goal is to learn predictive steering reactions in a way that the initial reflex is avoided. This is achieved by changing the connection weights ρ_1, such that the learner can use the earlier signal at x_1 to generate an anticipatory reaction. It is known that weights stabilise and learning stops at the condition $x_0 = 0$ when the reflex is not triggered anymore (Porr and Wörgötter, 2003a). The convergence properties of this kind of closed loop learning are discussed in Porr and Wörgötter (2006) and Porr and Wörgötter (2003b).

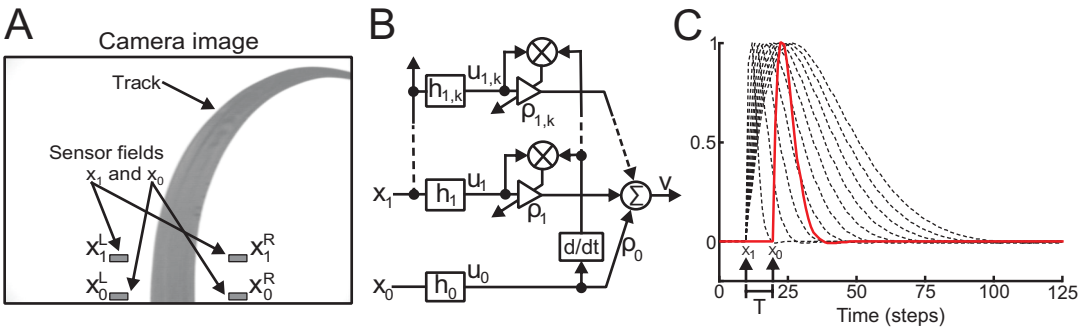

Figure 2.2: **A)** Camera image showing a track from the robot's perspective and relative positions of the predictive sensory inputs x_1 and reflexive sensory inputs x_0. **B)** Schematic diagram of the learning system. Inputs x, resonator filters h, connection weights ρ, output v. The symbol $\otimes$ denotes a multiplication, d/dt a temporal derivative. The amplifier symbol stands for a variable connection weight. Dashed lines indicate that input x_1 is fed into a filter-bank. **C)** Resonator filters h_0 (solid line) for the input signal x_0 and $h_{1,k}$ (dashed lines) for the x_1 given by parameters $f_{1,k} = 2.5/k\ Hz$, $k = 1, \ldots, 10$ for the filter-bank in the x_1 pathway. Frequency of the x_0 pathway was $f_0 = 1.25\ Hz$. Damping parameter of all filters was $Q = 0.6$.

2.4 Embedding learning in a closed-loop scenario

Fig. 2.3 shows how such a learning unit can be embedded in a closed-loop system. Initially (see panel A) the system is set up only to react to the near-sense x_0 by ways of a reflex. This reflex will after some behavioural delay reset the signal from the near-sensor again to its starting value (often zero) closing the loop. In more technical terms, this represents a negative feedback-loop controller. The learning system, however, contains a second, predictive loop (panel B) from a sensor x_1 that receives an earlier signal (far-sensor). At the beginning of the learning, synapses ρ_1 which convey information from the far-sense are zero and in Fig. 2.3 B only the inner loop x_0 is functioning. During learning, synapses ρ_1 will get strengthened and the system will increasingly better react to the far-sense. As a consequence reactions occur earlier and the reflex based on x_0 will not be triggered anymore. Effectively, the inner loop has functionally been eliminated after learning (see Fig. 2.3 C). A forward-model of the reflex has been built by Porr and Wörgötter (2003b). The learning of a forward model makes this approach appear similar to "feedback-error learning" as introduced by Gomi and Kawato (1993), but there are distinctive differences as will be discussed later (see section 2.13.3).

Intuitively the mechanism introduced in Fig. 2.3 will work with any aversive reflex.

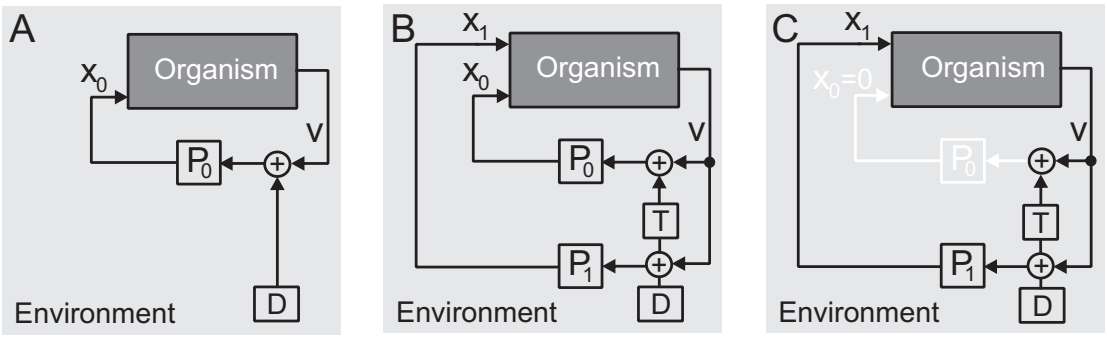

Figure 2.3: Schematic diagram of the control **(A)**, learning **(B)** and post-learning case **(C)**. Components of the learning system: sensor inputs x_0 and x_1, motor output v, P_0 denotes a reflexive pathway and P_1 a predictive pathway. D - disturbance, T - time delay between sensory inputs x_1 and x_0.

One should, however, note the same mechanisms can also be used to learn earlier attraction reactions. Already Braitenberg (1984) had nicely demonstrated that it is the sign-combination of the motor signals which determines the resulting reaction (aversion versus attraction) in his vehicles. Here, similarly, we can define the behavioural outcome by ways of the motor signals leaving the learning mechanism unaffected (see Porr and Wörgötter, 2003b, 2006 for examples of attraction reflexes). Regardless of the motor-signs, the learning goal is always to *avoid the earlier, near-sense-triggered reflex* leading to a situation where $x_0 = 0$. Porr and Wörgötter (2003b, 2006) were able to prove mathematically that synaptic weights will stop to change as soon as this condition ($x_0 = 0$) is fulfilled. Hence learning terminates as soon as the newly learnt behaviour is successful, which creates a nice self-stabilising property of such systems.

2.5 Simple learning architecture

2.5.1 Physical and neuronal setup of the system

A physical setup used for learning is presented in Fig. 2.4 A. A camera mounted at the front of the robot produces images of the track like the one shown. Since the robot drives forward, obviously sensor fields more at the top of the image ($x_1^{L,R}$) represent far-sensors (predictive inputs), while those at the bottom ($x_0^{L,R}$) can be regarded as near-sensors (reflexive inputs). Initially the robot reacts abruptly only to the near-sensors as soon as the image of the track moves over one of these near-sensor fields. The robot makes left turn if sensor on the left side is triggered and vice versa. As a consequence the robot will be brought back to a situation where the

track will remain mostly in the centre of the image. As mentioned before the learning goal is to learn predictive and smoother steering reactions. This can be achieved by changing the synaptic weights of the far-sensor fields in an appropriate way such that earlier and smoother steering reactions will be elicited leading to the situation that the near-sensor fields will never be triggered again, hence, avoiding the initial reflex.

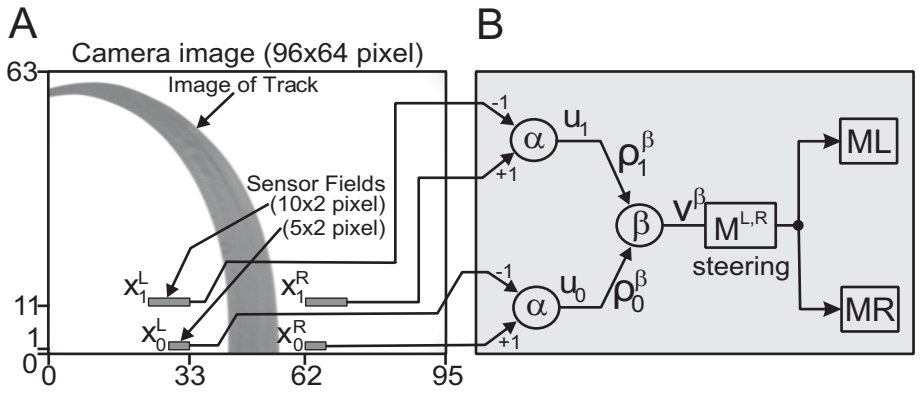

Figure 2.4: Physical and neuronal setup of the simple learning architecture. **A)** Camera image with left and right sensor fields marked by $x_1^{L,R}$ and $x_0^{L,R}$. **B)** The simple neuronal setup of the robot. Symbols α and β denote neurons, u denote filtered input signals x, ρ connection weights and v the output of the neuron β used for steering. v is calculated by the method shown in Fig. 2.2 B and its corresponding Eq. 2.1. $M^{L,R}$ is given in Eq. 2.4 and transforms v to the motor output.

A simple neuronal setup of the robot is presented in Fig. 2.4 B. It has three neurons, two are essentially only summation nodes, which we, for consistency, also call neurons α. They have fixed weights ($+1$ for right side inputs and -1 for left side inputs). In addition there is one neuron β with changing synapses on which all signals converge. Synaptic weights ρ_0^β are also set to a fixed value of 1 and only weights ρ_1^β of all ten filters (see Fig. 2.2 C) change. The output v^β is used to control the motor signals of the robot. Note, in this experiment the setup for the weight development is symmetrical but with inverted signs for left versus right curves. Hence only one set of weights ρ_1^β develops. This is motivated by the fact that, in a natural setup, left and right curves do not have any *a priori* bias. Situations were, for example, left curves are always on average sharper than right curves are not realistic. Hence, weights learnt for a left curve might as well be applied, with inverted sign, to a right curve (and vice versa), where learning will commence if the learnt weights are not sufficient. Given that the learning algorithm is linear, it would not make any difference if inputs were all converging directly onto β. Note, since the robot is continuously driving, we perform on-line and not batch learning.

2.6 Sensory-motor system

2.6.1 Sensory input

As described in the introduction, a far-sensor (predictive) pathway and a near-sensor (reflexive) loop can be defined from sensor fields in the image of a forward pointing camera on the robot.

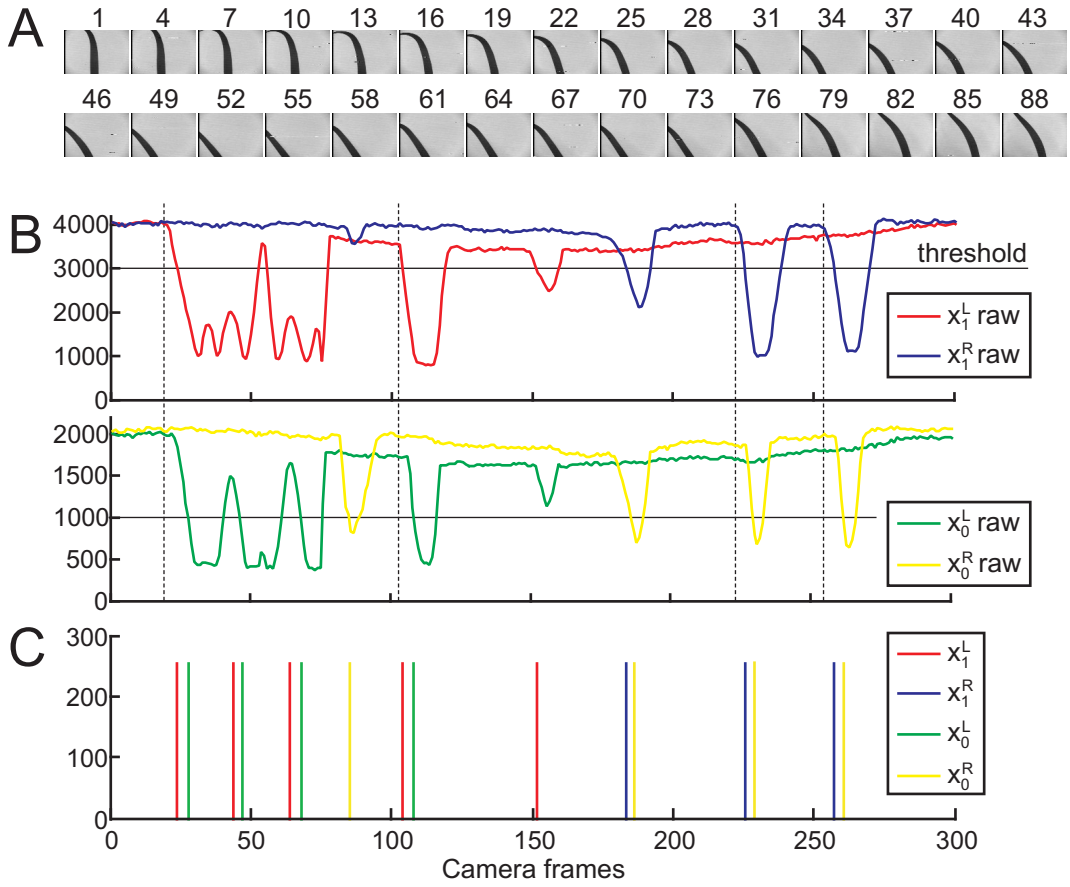

Figure 2.5: **A)** Sequence of camera frames taken from a left curve (here the number denotes the camera frame in a sequence). **B)** Raw signals $x_{0,1}\,raw$ obtained from sensor fields in the camera image. **C)** Preprocessed input signals $x_{0,1}$ of the learning system. Track layout is shown in Fig. 2.1 B. Signals before camera frame 150 come from the left turn, those after frame 150 from the right turn of the robot.

Fig. 2.5 A shows a sequence of camera frames obtained on a left-right track (see Fig. 2.1 B) during a left curve and the corresponding raw input signals (panel B)

obtained from the sensor fields $x_{0,1}^{L,R}$ (defined as the sum over all pixel values within the sensor field; pixel values are between zero (black colour) and 255 (white colour). The vertical dashed lines in panel B show that signals x_1 are indeed earlier than those at x_0. The sequence of camera frames in panel A demonstrates that the ego-motion of the robot creates quite some variability in the field of vision of the robot (see video camera.mpg[2]), for example the moving-out and moving-in of the bent line, clearly visible in the second row in panel A. This creates a temporally inverted sequence of input events. Learning needs to be robust against such effects as well as against other problems that arise from this behaviourally self-generated variability. Note that we use a threshold for raw input signals ($x_{0,1}^{L,R}\,raw$) before we feed them to the neuronal circuit to get rid of background noise. The sensory inputs $x_{0,1}^{L,R}$ could obtain binary value of 255 or zero depending on whether the raw inputs are below the given threshold or not (see Fig. 2.4 B and C). The threshold for the reflex $x_0\,raw$ and the predictor $x_1\,raw$ is 1000 and 3000 units respectively. We limited input activity of inputs x_0 and x_1 in our model by a time period T_r (similar to a refractory period in real neurons, Kandel et al., 2000), which means that if there was an input produced by triggering a sensory field at time moment t then there will be no inputs elicited for the next T_r time units. In our model we use $T_r = 20$ camera frames. In addition we also use lateral-temporal inhibition across the inputs of the left and the right side in order to avoid unwanted correlations. This means that whenever a sensor field on the left or the right side is firstly triggered it will inhibit inputs coming from the other side for the next T_i time units. Here we use $T_i = 20$ camera frames.

2.6.2 Motor output

The robot has a left and a right motor, which both receive a certain forward drive leading to a constant speed of 0.125 m/s in all experiments. This signal is modified by braking ($|v^\beta|$) and steering ($\pm v^\beta$). So, for the left M^L and the right motor M^R we get:

$$\begin{aligned} M^L &= 1.1905 \times 10^{-4}\,(3097 - |v^\beta| - v^\beta) - 0.2437\ m/s, \\ M^R &= 1.1905 \times 10^{-4}\,(3097 - |v^\beta| + v^\beta) - 0.2437\ m/s. \end{aligned} \tag{2.4}$$

Numerical constants are determined by the 12-bit resolution of the used DA-converter, where zero corresponds to *maximal reverse* and 4095 corresponds to *maximal forward* speed. For the chained architectures, introduced later (see Fig. 2.29 B, C), we use v^γ instead of v^β in the Eq. 2.4.

[2]Videos can be downloaded at *http://sites.google.com/site/ktomsite/driving-robot*

2.7 Learning with the simple architecture

2.7.1 Basic behaviour of the simple architecture

The simple learning architecture (see Fig. 2.4) was applied on the line following task and three different tracks (intermediately steep, shallow and sharp track) were used in this experiment. Results for the intermediately steep track are presented in Fig. 2.6 where we show sensory input of the left side x_0^L (panel A), synaptic weights ρ_1^β (panel B) and motor output v^β (panel C). Driving trajectories of the robot for the control case (i.e. before learning, reflexive behaviour) are shown in panel D and the trajectory after learning is shown in panel E. As we can see the late and weak reflex response by itself is not enough to assure line-following behaviour; therefore the robot misses the line whenever it drives without learning (see panel D and also video control.mpg[3]). In panels A-C two learning trials (separated by a vertical dashed line) are shown, between which connection weights were frozen and the robot was manually returned to its starting position. A rather high learning rate $\mu = 3 \times 10^{-6}$ was chosen to demonstrate fast learning. The cumulative action of reflex and predictive response allows the robot to stay on the line already *during* the first learning trial (trajectory not shown, but similar to the trajectory T_2, see panel E). In the first learning trial the motor signal (panel C) shows three leftward cumulative reflexive-predictive reactions (large troughs) and seven (two leftward and five rightward) non-reflexive (predictive) reactions. Note that cumulative responses consist of two components: the first component, smaller in amplitude, is the predictive response, whereas the second, larger in amplitude, is the reflexive response (see inset in panel C). In the second trial only predictive leftward and rightward steering reactions occurred and the reflex was not triggered anymore. An appropriate steering reaction was learnt after three learning experiences (later on referred to LE) reflected by the three peaks in the weight-curve in panel B, during the first learning trial corresponding to about 50 cm of the track (total length of the track was approximately 1.7 m). The left reflex signal x_0^L is shown in panel A where we observe that the reflex was triggered three times (three troughs below the threshold) which corresponds to three LEs. To ensure weight stabilisation we employed a threshold where values of x_0 above the threshold were set to zero (similar to the mechanical arm experiment in Porr and Wörgötter, 2006). Due to the symmetry of this setup (see Fig. 2.4 B), learnt synaptic weights from the left curve could be equally applied to the right curve and no more reflexes were triggered after these first three LEs. We can also observe, that after learning the robot elicits steering reactions that are wider and much smaller in amplitude (compared to the steering reactions during learning) which as a consequence leads to smoother driving behaviour (for the whole learning process see video middle.mpg).

In addition two more extreme tracks were chosen to demonstrate the robustness

[3]Videos can be downloaded at *http://sites.google.com/site/ktomsite/driving-robot*

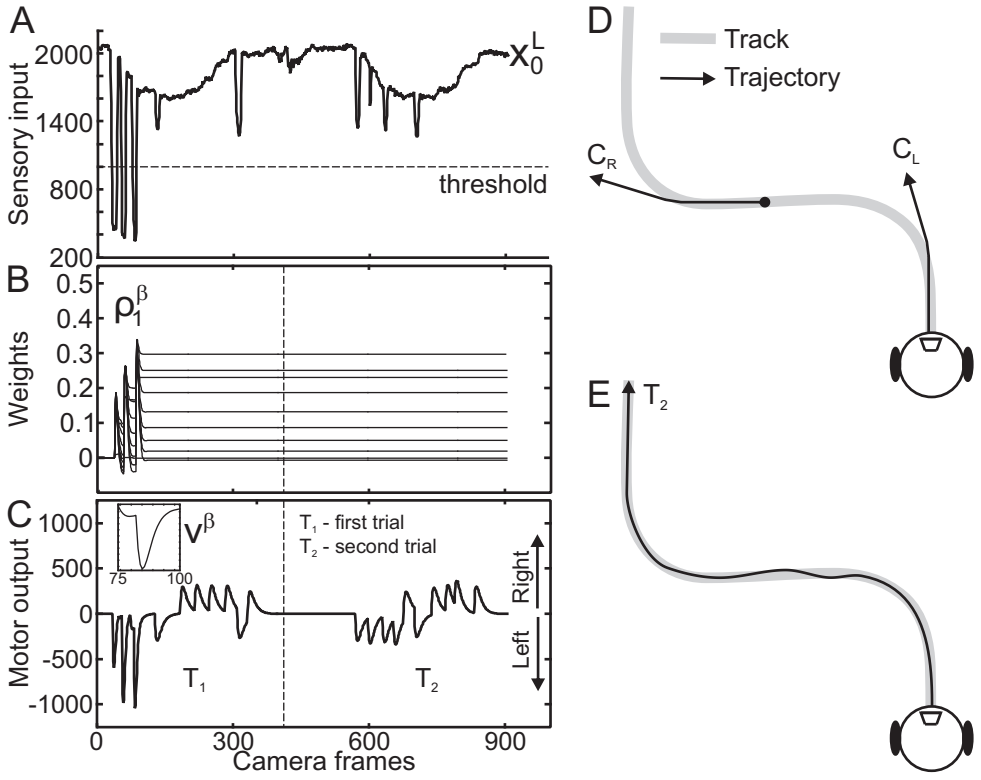

Figure 2.6: Results for the line following experiment using the simple architecture (see Fig. 2.4 B) on the intermediately steep track. Learning rate was $\mu = 3 \times 10^{-6}$. **A)** Reflex signal x_0^L, **B)** connection weights ρ_1^β, **C)** motor output v^β, **D)** driving trajectories for the left (C_L) and the right turn (C_R) for the control case (before learning). **E)** Driving trajectory for the second trial (after learning).

of these findings. The results for a shallower track (total length of the track was approximately 1.5 m) are presented in Fig. 2.7 and are similar to those from the previous experiment, but for this track learning stopped already after two experiences even with a lower learning rate of $\mu = 2.5 \times 10^{-6}$ as compared to the previous experiment where the learning rate was $\mu = 3 \times 10^{-6}$. As expected smaller synaptic weights (panel A) and a much weaker steering reaction (panel B) was learnt and weights (panel A) are smaller. For a movie of the whole learning process see video shallow.mpg.

The third experiment was performed using a track with very sharp corners (total length of the track was approximately 1.5 m) and a relatively higher learning rate $\mu = 6.5 \times 10^{-6}$ was used (see Fig. 2.8 C). This was done to demonstrate that fast and stable learning is possible even for such a sharp track. The results of three

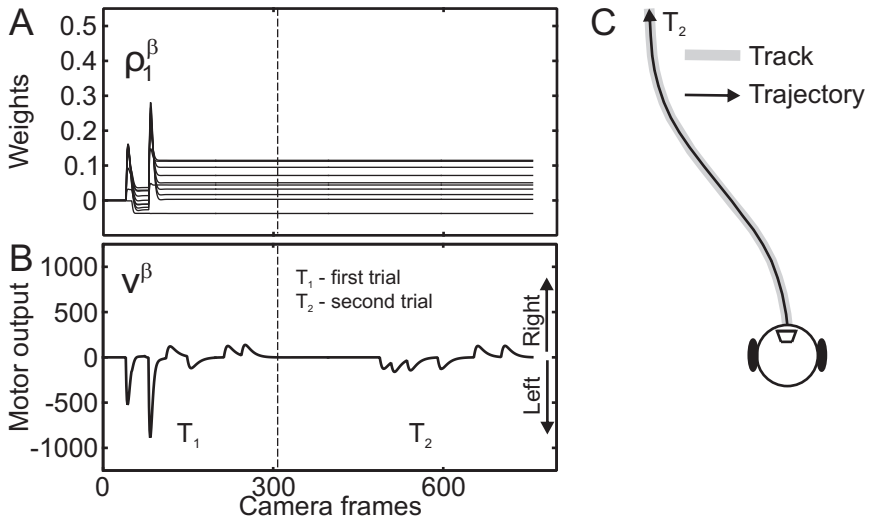

Figure 2.7: Results for the line following experiment using the simple architecture (see Fig. 2.4 B) on the shallow track. Learning rate was $\mu = 6.5 \times 10^{-6}$. **A)** Connection weights ρ_1^β, **B)** motor output v^β, **C)** driving trajectory for the second trial (after learning).

learning trials (separated by vertical dashed dashed lines) are presented in Fig. 2.8. The robot missed the track twice and finally succeeded in the third trial (see also video sharp.mpg). Learning stopped after three experiences. As before, it could use the learnt weights also for the right curve. Note, however, as a consequence of the general arrangement, the robot now "cuts corners". This is a result of the fact that the predictive sensor field is at some distance from the bottom of the camera image. Because steering necessarily always consists of a sequence of short straight trajectories, the robot will always take shortcuts if the curves are too sharp and/or if the predictive sensor field is high up in the camera image.

In general we observed that the robot can learn the task fast even with a low learning rate as long as the track is shallow but needs higher rates to be able to follow the sharp track after about the same number of reflexes. If the same learning rate is chosen for all tracks then more reflexes are needed for the sharp track than for the shallow one.

Fig. 2.9 shows results for two control experiments with a shallow left and an increasingly sharper right curve (see Fig. 2.9 C). Connection weights ρ_1^β (panel A) and motor output v^β (panel B) of four learning trials (separated by dashed lines) are shown for a relatively low learning rate $\mu = 0.4 \times 10^{-6}$. At the beginning, the low learning rate prevents the robot even from following the very shallow left curve (see trajectory T_1 in Fig. 2.9 C). In the second trial, the robot succeeded for the left

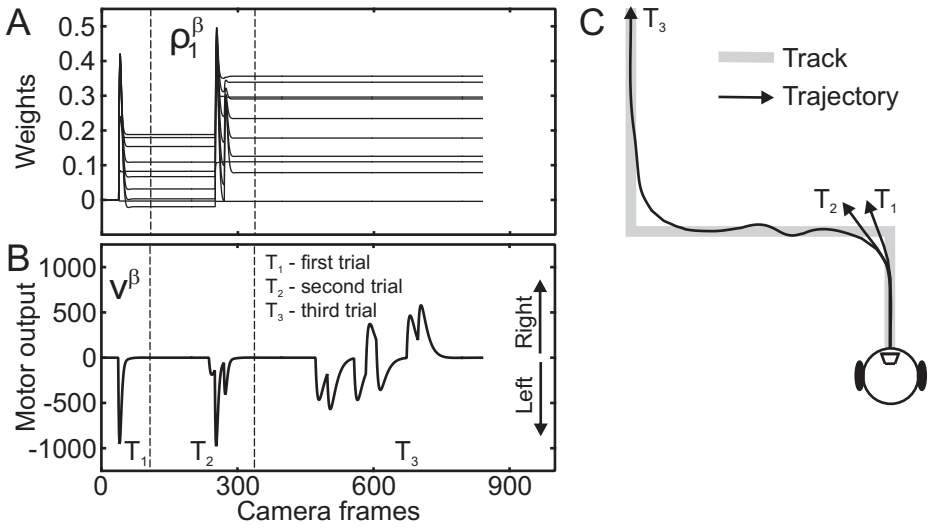

Figure 2.8: Results for the line following experiment using the simple architecture (see Fig. 2.4 B) on the sharp track. Learning rate was $\mu = 2.5 \times 10^{-6}$. **A)** Connection weights ρ_1^β, **B)** motor output v^β, **C)** driving trajectories two learning trails T_1 and T_2, and T_3 for the post-learning trial.

curve at the beginning of the right curve but the learnt steering reaction still was not enough to allow it to follow the sharper parts of the right curve at the end of the spiral track (see trajectory T_2 in Fig. 2.9 C). In the third learning trial the robot succeeded to follow the whole trajectory completely (see trajectory T_3 in panel C) but still most of the time a mix of predictive and reflexive (large peaks) steering reactions occurred. The robot continued to improved its steering reactions in the fourth trial (trajectory not shown, but see video of whole experiment: spiral-low.mpg) where one can see more non-reflexive reactions (smaller peaks) and less predictive+reflexive reactions than in the third trial. As expected from the linearity of our learning rule, in the right curve the system can use the weights learnt during the left curve up to the point where the right curvature remains below the left curvature (three leftward reactions and then two rightward reactions in the fourth trial) after which weights will continue to grow (large peaks). However, learning is not yet finished at this stage and would need more trials until weights finally stabilise.

To speed-up the learning process a higher learning rate of $\mu = 1.5 \times 10^{-6}$ was used and three learning trials are presented in Fig. 2.9 D-F. In this case, the robot is able to stay on the line already during the first learning trial (trajectories not shown but see video spiral-high.mpg) but still more predictive+reflexive (large peaks) than non-reflexive steering reactions occurred (see panel E). In the second trial only two predictive+reflexive reactions occurred whereas in the last trial only non-reflexive

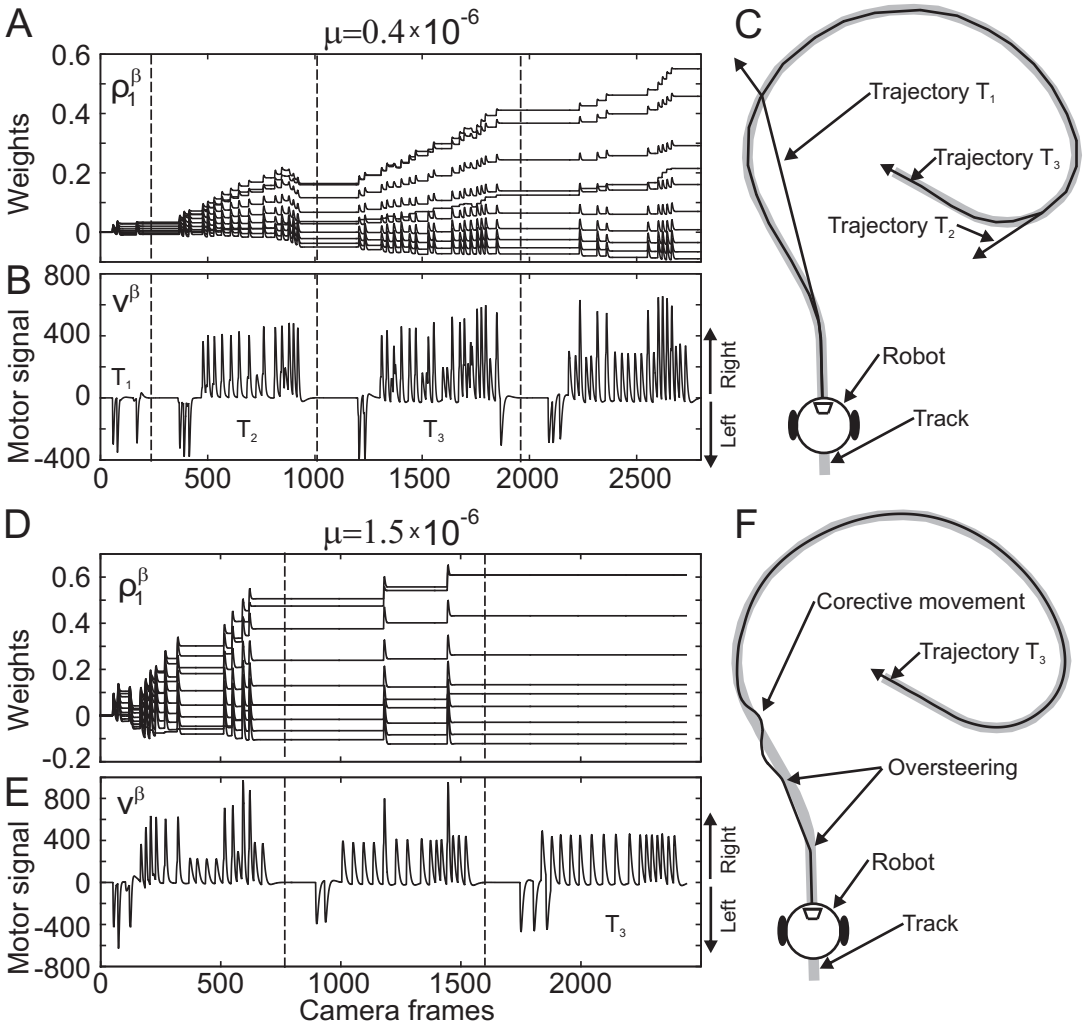

Figure 2.9: Results of the driving robot experiment using the simple architecture (see Fig. 2.4 B) on a spiral track. **A-C)** Results for a learning rate of $\mu = 0.4 \times 10^{-6}$. **A)** Connection weights ρ_1^β, **B)** motor output v^β, **C)** corresponding driving trajectories T_1, T_2 and T_3 during learning process. Note, learning has not yet finished after T_3, but improves gradually towards a smooth trajectory. **D-F)** Results for a learning rate of $\mu = 1.5 \times 10^{-6}$. **D)** Connection weights ρ_1^β, **E)** motor output v^β, **F)** Final driving trajectory T_3 reached after two, not-shown learning trajectories, when using the higher learning rate of $\mu = 1.5 \times 10^{-6}$. In this case we find weight stabilisation after two trials (see panel C), but learnt weights will lead to too strong reactions (over-steering) for shallow curves which are compensated by corrective movements.

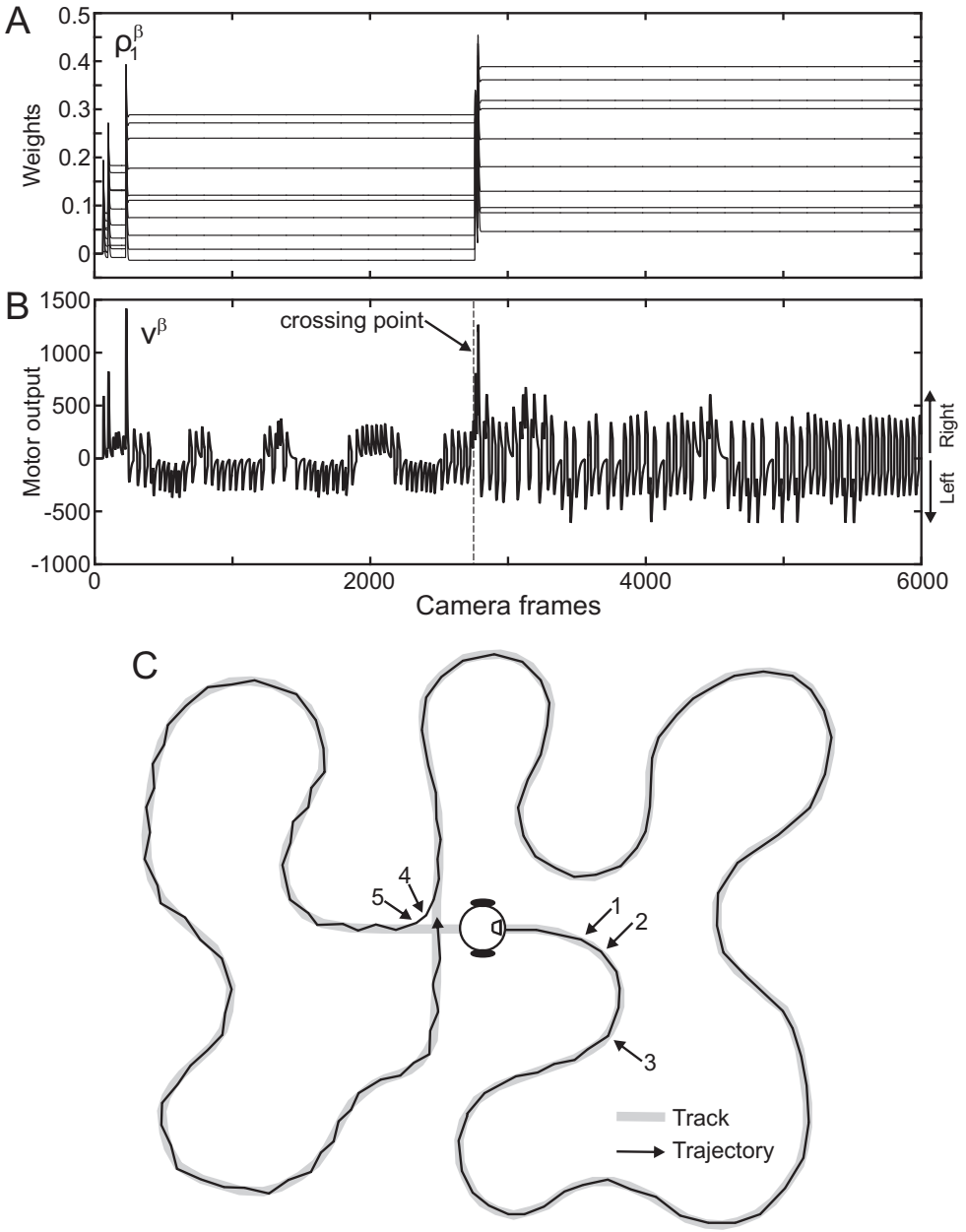

Figure 2.10: Results for the line following experiment using the simple architecture (see Fig. 2.4 B) on the maze track. Learning rate was $\mu = 3 \times 10^{-6}$. **A)** Connection weights ρ_1^β, **B)** motor output v^β, **C)** driving trajectory.

steering reactions occurred and weights did not change anymore. When we use the final weights learnt with the sharp curve afterwards for driving a through the shallow

left curve in a third trial the robot over-steers slightly the left curve and then makes an oscillatory corrective movement, however, without triggering reflexes, in order to remain on the line (see trajectory T_3 in Fig. 2.9 F).

We also did an experiment to see how the robot behaves on a difficult track with different kinds of curvatures (see Fig. 2.10). The total length of the track was ≈ 14 m. Connection weights and the motor output are shown in panel A and B. The robot had three learning experiences at the beginning (see panel A and arrows in panel C) while turning to the right and after that the reflex input was not triggered till the robot approached the crossing point where the robot turned to the right (see trajectory in panel C) and the reflex was triggered twice again. As expected from the linearity of our learning rule, the robot can use the learnt weights up to the point where the curvature remains below the experienced curvatures after which weights will continue to grow. After ≈ 2740 camera frames (crossing point) the reflex was not triggered anymore and weights stopped changing. When the robot approached the crossing point for the second time it went straight and for the third time (trajectory not shown) it turned to the left (see video maze.mpg). In general we obtained that the robot uses the final weight learnt for the sharpest curve and over-steers when driving on the shallower curves which leads to the oscillatory driving behaviour (compare motor output signals before and after crossing point). Note, as the robot does not use any assumptions about track smoothness, for the machine both solutions, driving straight or turning, are equivalent at the crossing point in the centre of the track and the selection of a certain behaviour only depends on the status of its sensory inputs.

2.7.2 Statistical evaluation of the simple architecture

In the experiments above it has become clear that our system performs on-line (and not batch) learning. Hence the most critical parameter affecting the convergence of learning is in which way the momentary behaviour will influence, or rather generate, the next learning experience. Ultimately this is given by the sequence of viewing angles which the robot creates due to its own driving. As a consequence an investigation of the influence of the viewing angle on the learning should provide the most relevant information about the robustness of this system. Other relevant parameters are learning rate as well as relative placement of the different sensor fields.

Thus, to investigate the robustness against these parameters we used a simulation and performed a set of experiments where we let the simulated robot learn to follow left-right tracks with angles of 20, 45 and 90 *degrees* (see Fig. 2.11 A). The total length of all tracks was 360 units while its thickness was 1 unit. The radius of the robot was $r = 20$ units and the size of the sensory fields $x_{0,1}^{L,R}$ was 1×1 unit. Positions of sensory fields were defined as shown in Fig. 2.11 B. We used the neuronal setup as presented in Fig. 2.4 B. The output of the neuron v^{β} modified by transformation function $P_{x,y}$ (Eq. 2.6) instead of function $M^{L,R}$ (Eq. 2.4) was used here to change the position of the robot in the environment. The position of the robot $P_{x,y}$ was defined

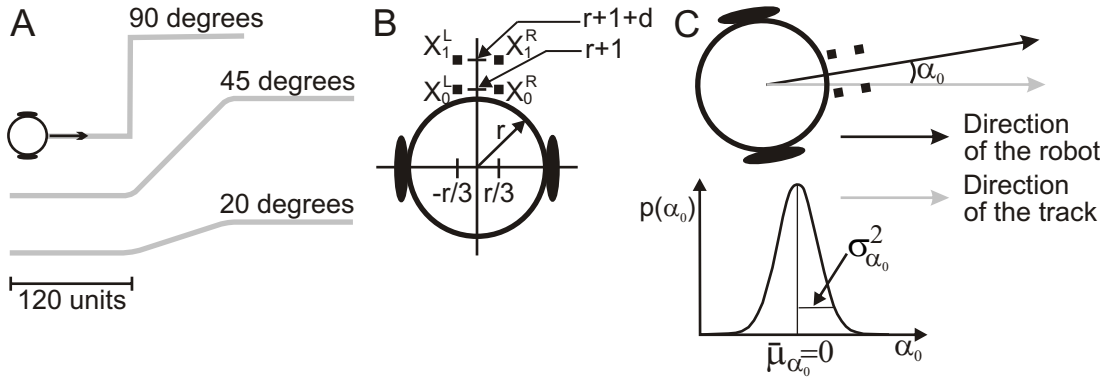

Figure 2.11: Setup of the simulated line following experiment. **A)** Tracks with curvatures of 20, 45 and 90 *degrees*. **B)** Setup of the simulated robot. Dots represent positions of the sensor fields $x_{0,1}^{L,R}$, $r = 20$ units is the radius of the robot, $d = [2, 3 \ldots 10]$ units is the distance between sensors x_1 and x_0. **C)** Direction angle α_0 of the robot at its starting position, given by the deviation from the direction of the track when placing the robot at the starting position. In the experiments a Gaussian distribution of α_0 has been used with mean $\overline{\mu}_{\alpha_0} = 0$ and different variances $\sigma^2_{\alpha_0} = [1, 4, 9]$.

by the x and y coordinates of the robot's mass centre. The signal v^β is then directly used to change the robot's driving angle α, i.e. v^β directly corresponds to the change of the turning angle:

$$\frac{d\alpha}{dt} = -G_{st}\ v_t^\beta, \tag{2.5}$$

where $G_{st} = 0.01$ is the steering gain. The change of the robot's position is calculated as follows:

$$\begin{array}{l} \frac{dP_x}{dt} = (\nu - G_{br}\ |v_t^\beta|)\ cos(\alpha_t), \\ \frac{dP_y}{dt} = (\nu - G_{br}\ |v_t^\beta|)\ sin(\alpha_t), \end{array} \tag{2.6}$$

where $\nu = 1$ is the constant default velocity and $G_{br} = 0.001$ is the breaking gain.

The sensory inputs $x_{0,1}^{L,R}$ can take binary values 255 or zero depending on whether the sensor field is triggered or not. We used a filter bank of ten filters to prolong inputs $x_1^{L,R}$ given by parameters $f_1 = 0.5/k, k = 1 \ldots 10$, for x_1, wheras for x_0 we used one filter with the parameter $f_0 = 0.25$. Damping parameter of all filters was $Q = 0.6$.

To evaluate the robot's performance we define three (AND-connected) conditions to measure success:

1. The correlation coefficient between robot's trajectory and the whole track is > 0.90.

2. The reflex is not triggered in three consecutive trials after connection weights stopped changing.

3. The robot completed the task within maximally 20 trials.

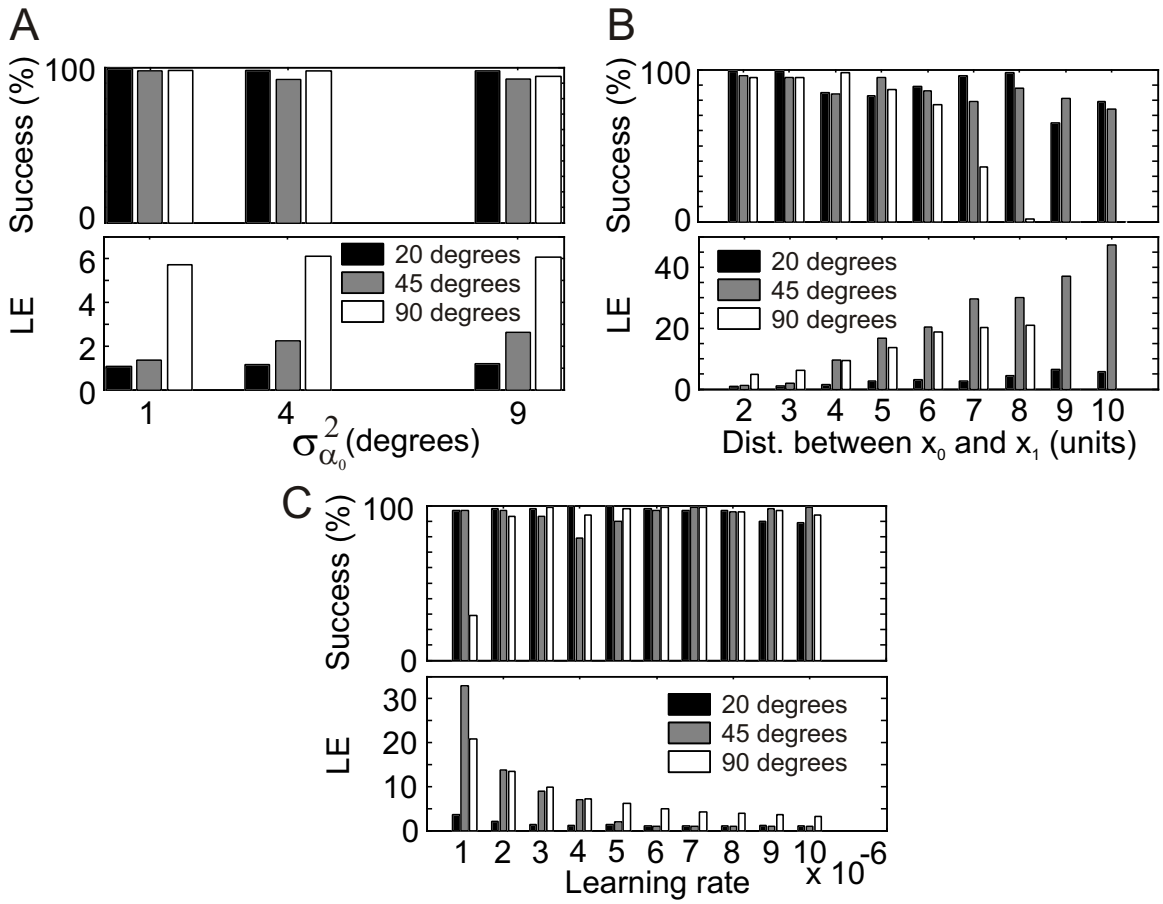

Figure 2.12: Results of the simulated line following experiment using the simple learning architecture (see Fig. 2.4 B). **A)** Success in 1000 experiments and average number of learning experiences (LE) needed to accomplish the task within successful experiments are plotted against the variance $\sigma^2_{\alpha_0}$ of robot's direction at the starting position. Learning rate was $\mu = 5 \times 10^{-6}$ and distance between sensor fields x_1 and x_0 was $d = 3$. **B)** Success in 100 experiments and average number of LE plotted against the distance between sensor fields x_1 and x_0. Learning rate was $\mu = 5 \times 10^{-6}$ and variance $\sigma^2_{\alpha_0} = 4$. **C)** Success in 100 experiments and average number of LEs plotted against the learning rate μ. The variance was $\sigma^2_{\alpha_0} = 4$ and the distance between sensor fields x_1 and x_0 was $d = 3$.

If these three conditions are not fulfilled at the same time then we count an

experiment as a failure.

Results demonstrating the influence of the robot's position angle (α_0, see Fig. 2.11 C) on placing the robot at the starting position are presented in Fig. 2.12 A. We plot the success rate in 1000 experiments and the average number of learning experiences (LE) needed to accomplish the task within successful experiments against the variance of the distribution of the starting angle $\sigma^2_{\alpha_0}$. The success is slightly decreasing if we increase the variance of the starting angle distribution σ^2_{α}, but we still get high performance and the success rate is $0.92 < success \leq 0.99$ for all tracks. More learning experiences are needed to accomplish the task if $\sigma^2_{\alpha_0}$ is increased. Also, as expected, more LEs are required for the sharp track as compared to shallower ones.

Results of 100 experiments for different positions of the predictor sensor x_1 are shown in Fig. 2.12 B. Success rate decreases if the distance between inputs is getting larger for the sharp track whereas for the shallower tracks decrease is less significant when the distance is very large ($d = [9, 10]$). The number of necessary LEs is increasing if the distance between x_1 and x_0 is getting larger. This is due to the weight change curve of the ICO learning rule (Porr and Wörgötter, 2006). If the inputs are spaced further apart in time then correlations are weaker, the connection weights do not change so fast, and more repetitions are needed to complete learning. Due to this the robot never succeeded to steer the sharp track within 20 trials when the distance between x_1 and x_0 was $d > 8$.

We also investigated the influence of the learning rate and results of 100 experiments are presented in Fig. 2.12 C. The learning rate does not affect the performance except for the sharp track. When the learning rate is relatively low the robot does not succeed learn steering the curve within 20 trials. As expected we find that with a higher learning rate less LEs are needed to complete the task, because with a higher learning rate weights are growing faster and the task is learnt quicker.

2.8 Development of receptive fields with the simple architecture

In the following we would like to show the development of receptive fields (RF) with the simple architecture. In saying RF-development we are specifically focusing on the development of the spatial features of an initially unresponsive (all weights zero) field. Also we will average over all synapses coming from the same input pixel via the used filter bank. Hence the spatio-*temporal* structure of the plotted field is more complex. Filters are the same for all pixels and we are in this study more interested in the general spatial structure of the RFs and in their stability. Therefore we will neglect the temporal domain.

2.8.1 Physical and neuronal setup of the system

Physical and a neuronal setup used for the receptive field (RF) development are shown in Fig. 2.13 and are similar to the setups presented above (see Fig. 2.4). Here predictive sensor fields $x_1^{L,R}$ with size of 10×2 px are replaced by receptive fields with size of 15×15 px (Fig. 2.13 A) where each pixel within the receptive field represents an individual input $x_{1_{i,j}}$ (in total 255 inputs) with corresponding plastic synapse $\rho^{\beta}_{1_{i,j}}$ with which the input converges onto the neuron β. As before we have a symmetrical setup where inputs from the left side have the fixed weight -1 and inputs from the right side $+1$. Synaptic weight ρ_0^{β} is also set to a fixed value of 1 and only weights ρ_1^{β} of all ten filters (see Fig. 2.2 C) change.

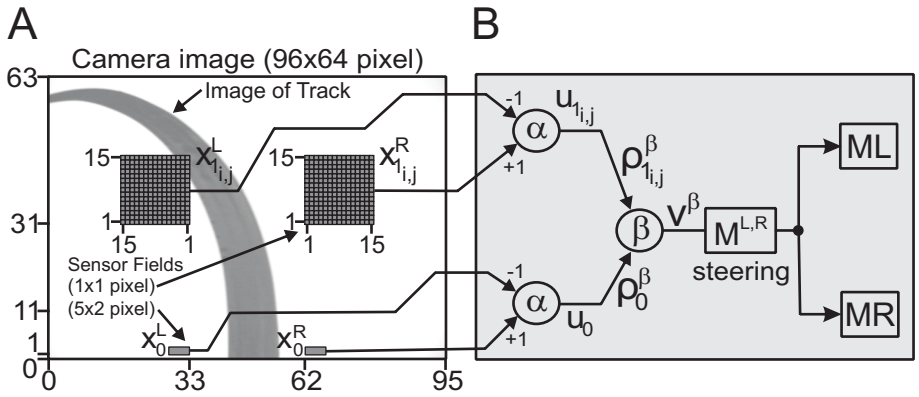

Figure 2.13: Physical and neuronal setup of the receptive field (RF) development using the simple learning architecture. **A)** Camera image with left and right sensor fields. The receptive filed positions are denoted by $x_{1_{i,j}}^{L,R}$, where $i = 1 \ldots 15, j = 1 \ldots 15$ are the indices of the RF pixels, and sensor field positions $x_0^{L,R}$. **B)** The simple neuronal setup of the robot. Symbols α and β denote neurons, u denote filtered input signals x, ρ connection weights and v the output of the neuron used for steering. v is calculated by the method shown in Fig. 2.2 B and its corresponding Eq. 2.1. $M^{L,R}$ is given in Eq. 2.4 and transforms v to the motor output.

2.8.2 Learning primary receptive fields

Results of the receptive field (RF) development using the simple neuronal setup on three different tracks (shallow, intermediately steep and sharp) are presented in Fig. 2.14 where we plot weights of the right receptive field x_1^R. The left receptive field is the mirror image of the right one due to the symmetry of the learning setup. The obtained RFs have a line like structure with stronger and weaker sub-fields, where

low weight values correspond to the weak steering reactions and high weight values to the strong steering reactions. Note, in order to obtain structured receptive fields, a smaller learning rate and a few more repetitions had to be used. As expected, we obtained different RFs for different tracks. RF developed on the sharp track (panel C) have higher weights (total sum of all weights is 0.553) compared to RFs obtained from the shallower tracks shown in panel A and B (total sum of all weights is 1.068 and 2.304 respectively). For the shallow track (panel A) we obtained also negative weights which is due to the fact that some inputs experience temporally inverted correlations (predictor comes *after* reflex, see also Fig. refRFF C).

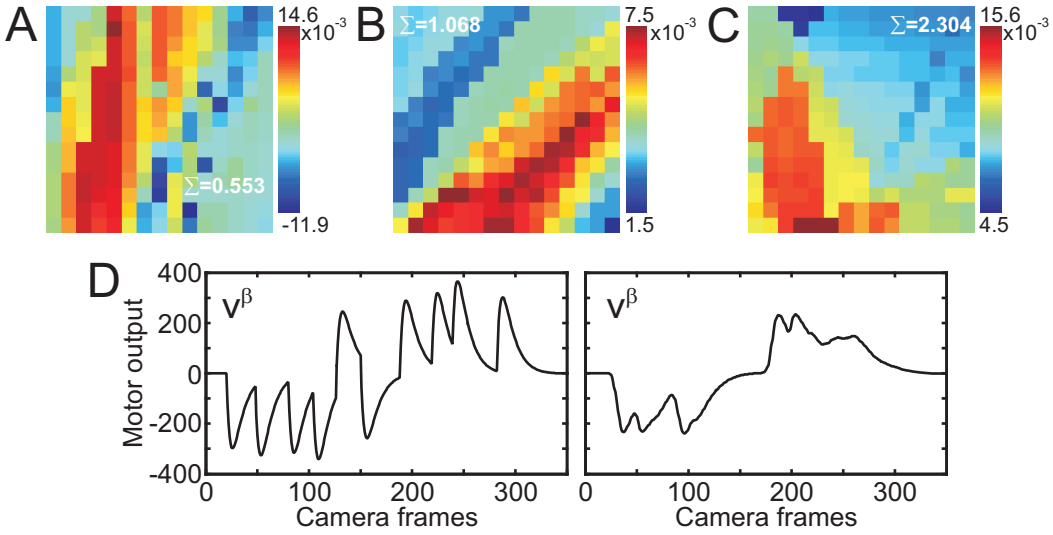

Figure 2.14: **A-C)** Results of the receptive field development using the simple neuronal setup (see Fig. 2.13). The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^{R}_{1_{i,j}}$ (the right receptive field). The number in the RF denotes the total sum of all weights ($\sum$). **A)** Results for the shallow track (see Fig. 2.7 C). Learning rate was $\mu = 1.7 \times 10^{-8}$. Learning stopped after three trials (see video rf-shallow.mpg). **B)** Results for the intermediately steep track (see Fig. 2.6 D). Learning rate was $\mu = 10^{-8}$, learning stopped after four trials (see video rf-middle.mpg). **C)** Results for the sharp track (see Fig. 2.8 C). Learning rate was $\mu = 1.7 \times 10^{-8}$. Learning stopped after six trials (see video rf-sharp.mpg). **D)** Comparison between learnt motor outputs obtained on the intermediately steep track by using single sensor fields (left) versus receptive fields (right). For more details please read the main text.

In Fig. 2.14 D we compare motor outputs learnt on the intermediately steep track (Fig. 2.6) by using single sensor fields (for the setup see Fig. 2.4) versus the receptive

field (RF) approach (for the setup see Fig. 2.13). We can see that the motor output obtained by using RFs (right panel) is smaller in amplitude and not as spiky as compared to the output obtained by using single sensor fields (left panel), which, as a consequence, corresponds to much smoother and more accurate line following behaviour compared to the single sensor field approach (see also videos middle.mpg and rf-middle.mpg[4]).

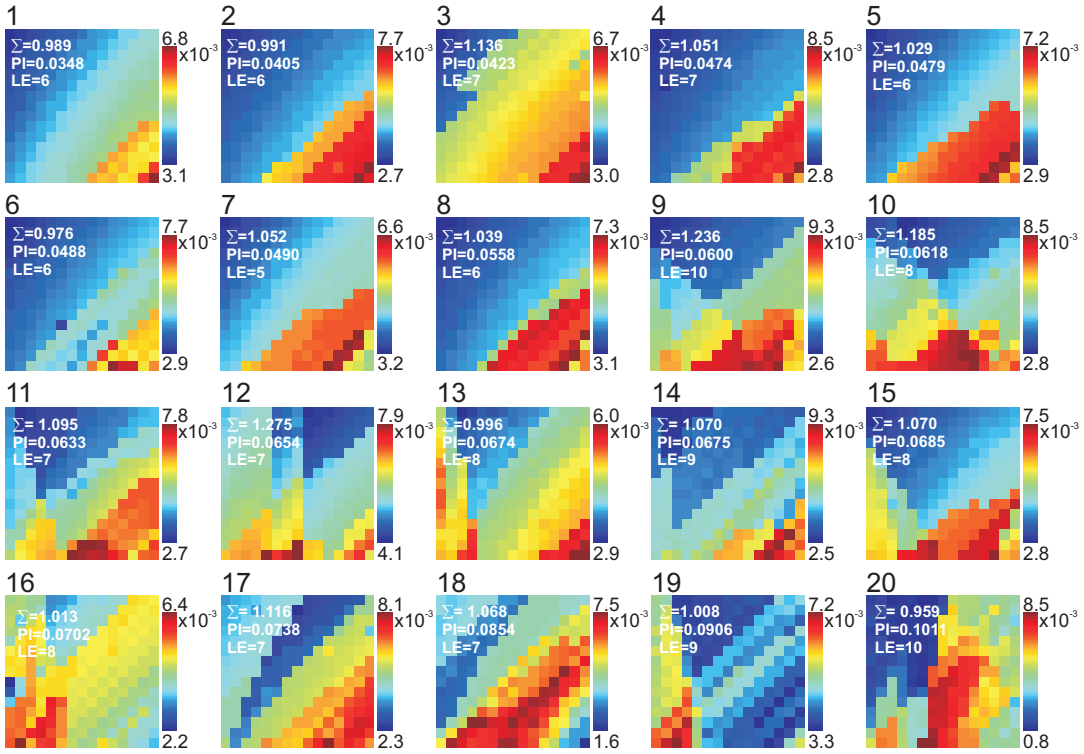

Figure 2.15: Results of the receptive field development using the simple neuronal setup (see Fig. 2.13) on the intermediately steep track (see Fig. 2.6 D) obtained from 20 experiments. The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^{R}_{1_{i,j}}$. Size of the RF (15x15 pixels) and position was as shown in Fig. 2.13 A. Learning rate was $\mu = 10^{-8}$. Numbers in the receptive fields correspond to the total sum of all weights ($\sum$), pattern inconsistency (PI) and the number of learning experiences (LE), respectively. Receptive fields are ordered according to the pattern inconsistency measure PI as given in the RF. For more details read the main text.

[4]Videos can be downloaded at *http://sites.google.com/site/ktomsite/driving-robot*

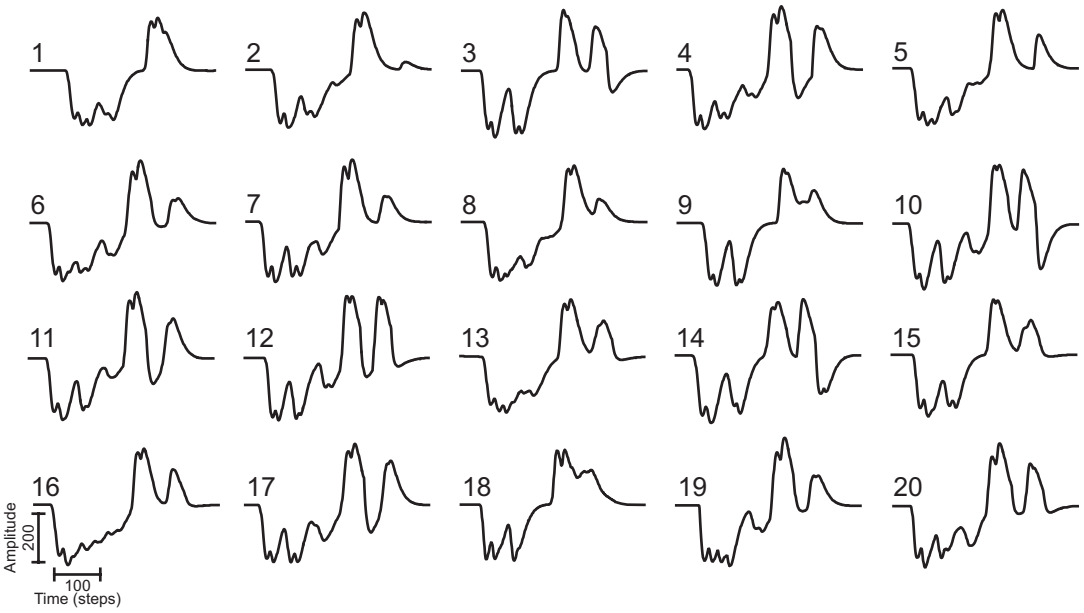

Figure 2.16: Corresponding motor outputs v^{β} obtained from the receptive fields shown in Fig. 2.15.

For experiments with the same parameters we would nonetheless expect to receive slightly differing RFs, because of the remaining contingencies mainly due to the manual placement of the robot and the noise in the system. To investigate the variability of the RFs we performed 20 experiments where we developed receptive fields on the intermediately steep track using the same system parameters (position of the RFs and learning rate). Obtained receptive are shown in Fig. 2.15 and corresponding motor output for each receptive field are shown in Fig. 2.16. Learning stopped after ≈ 7 learning experiences (LE) on average. RFs are grouped according to the structure of the RF. Here we represent the structure of RF by the *pattern inconsistency* measure which is defined as the average distance to the neighbouring weights computed over all weights within the receptive field (similar to the grey scale values used in self-organising maps, Kohonen, 2001). The equation for the pattern inconsistency (PI) measure is given in appendix A.1. The PI measure shows the dissimilarity between neighbouring weights, thus, receptive fields with gradient-like structure (see 1st RF in Fig. 2.15) lead to small pattern inconsistency values, whereas alternating or noisy RFs (17th and 20th RF respectively) lead to the higher PI values. Note that for a homogeneous RF one gets $PI = 0$ whereas for an RF with random structure of values from a uniform distribution we find $PI \approx 0.33$. We observe that some of the fields are relatively noisy which is mainly due to the reasons stated above, but, nonetheless,

behaviour is successfully learnt in all cases. Note that completely random receptive fields would lead to the value of $PI \approx 1$, whereas for our RFs we obtained much smaller PI values (for all RF we obtained $PI < 0.32$), thus, developed RFs are not random.

2.9 Analysis of the receptive field formation

2.9.1 Experimental setup

In the following we will analyse how system parameters influence formation of the receptive field pattern. To do so, we simulated the development of receptive fields on three different tracks as shown in Fig. 2.17 B. The setup of the robot used for simulation of the RF development is shown in Fig. 2.17 A. As before we used the simple neuronal setup as shown in Fig. 2.13) where the neuronal output v^β is directly used to change the robot's driving angle α and position as given in Eq. 2.5 and Eq. 2.6 with the steering gain $G_{st} = 0.005$ and breaking gain $G_{br} = 0.001$. If not mention elsewhere we used the following default system parameters. We used a filter-bank of ten filters to prolong receptive field inputs $x_1^{L,R}$ given by parameters $f_1 = 0.1/k, k = 1 \ldots 10$, whereas for reflexive input x_0 we used one filter with the parameter $f_0 = 0.05$. Damping parameter of all filters was $Q = 0.6$. The distance between RF position and reflex position was $d = 8$ and the learning rate was $\mu = 0.5 \times 10^{-8}$.

2.9.2 Dependence on the track and RF position

First of all we analysed how the structure (pattern) of the receptive field depends on the curvature of the track. We simulated the RF development on three different tracks (see Fig. 2.17 B) with the same system parameters as given above. The resulting receptive fields are shown in Fig. 2.18 A left column) and have different pattern location and orientation for a specific curvature. This is due to the fact that location and orientation of the pattern is determined by the input location within the receptive field at which the RF inputs correlate best with the reflex. In the middle and the right column of panel A we show the input intensity map of the RF during and after learning, respectively. This map shows the input activity for each RF pixel (input). It equals zero if there was no input at that pixel location and one if it was triggered most often relative to the number of inputs at the other pixels. The equation for the calculation of the input intensity map is given in appendix A.2. Intensity maps during learning show which inputs contribute to the RF development, and intensity maps after learning show which regions (inputs) of the RF drive the steering behaviour of the robot after learning. We observe that after learning for the shallow and the intermediately steep track there is only one dominant region within the RF, whereas for the sharp track there are two regions which are most active. This is because the

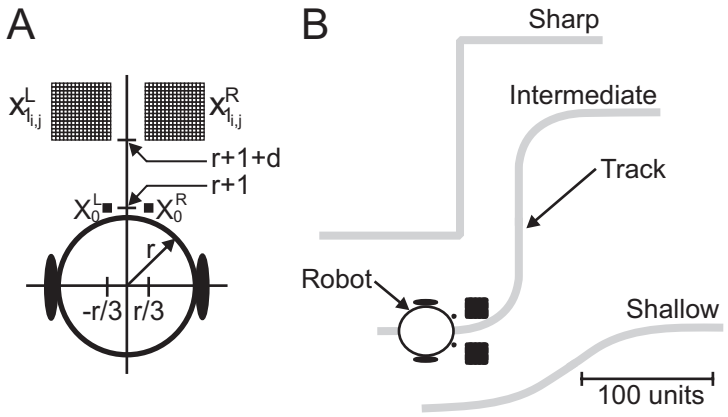

Figure 2.17: Setup of the simulation experiments for receptive field development. **A)** Setup of the simulated robot. Dots represent positions of the reflexive sensor fields $x_0^{L,R}$ (1×1 unit) and square grids represent positions of receptive fields $x_1^{L,R}$ (15×15 units). $r = 20$ units is the radius of the robot, d denotes the distance between position of the RF (x_1) and the reflex (x_0). **B)** Image of tracks with curvatures of different steepness: a shallow track, an intermediately steep track and a sharp track. The thickness of all tracks was 1 unit.

robot over-steers on the sharp track (see Fig. 2.17 C) which causes activity in the second region within the receptive field (on the side of the RF) in order to bring the robot back to the track. In Fig. 2.18 C driving trajectories of the robot after learning are shown for each track. As expected, we can see that the amplitude and width of the response curve is increasing as the curvature of the track gets steeper.

Results of the receptive field development for different RF positions are presented in Fig. 2.19 where we show RFs obtained on three different tracks. Here we varied the position of the RF by changing the distance d between the position of the reflex sensor field and the RF as shown in Fig. 2.17 A. From the results we can see that the location of the pattern is shifting downwards as we increase the distance d between reflex and RF position, i.e. shift the position of the RF upwards. This is expected as shifting the RF upwards causes a change in the input location at which RF inputs correlate best with the reflex. Thus, moving the RF downwards or upwards will change the location of the RF pattern while maintaining more or less the same pattern orientation for a particular track. We can also observe that for the shallow track faster learning is obtained (less LEs are required) when the RF is further away from the reflex ($d = 15$) whereas for the sharper tracks faster learning is obtained when the RF is closer to the reflex ($d = 9$ for the intermediately steep track and $d = 6$ for the sharp track).

Motor outputs together with driving trajectories of the robot for three different RF positions obtained on the intermediately steep track are shown in Fig. 2.20. Ob-

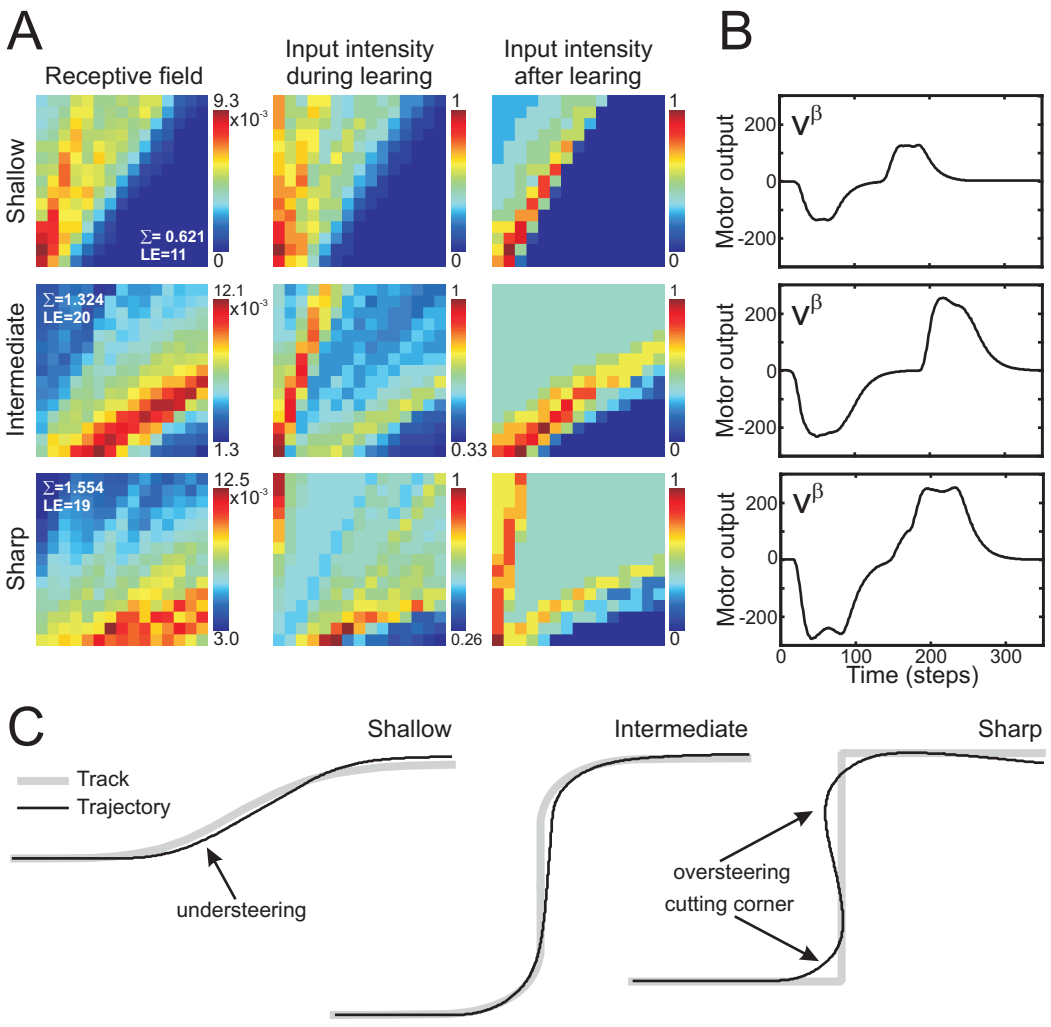

Figure 2.18: Results of the simulated receptive field development for three different tracks as shown in Fig. 2.17 B. **A)** The diagrams in the first column show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^{R}_{1_{i,j}}$ (the right receptive field). Numbers in the RF denote the total sum of all weights ($\sum$) and the number of learning experiences (LE). The diagrams in the second and the third column show input intensity maps of the RF during and after learning, respectively. For more details please read the main text. **B)** Learnt motor outputs v^{β} generated by the corresponding receptive fields (panel A). **C)** Driving trajectories of the robot shown for each track.

tained motor outputs are different which, as a consequence, leads to different driving behaviour. If the RF is too close then a relatively narrow motor response is generated

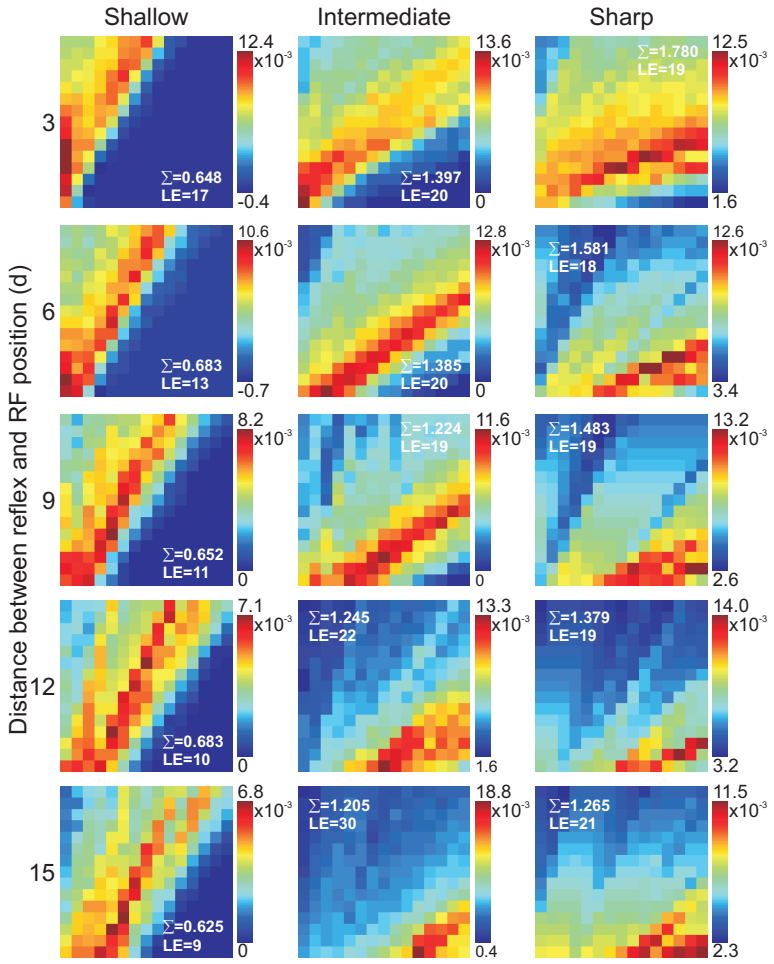

Figure 2.19: Results of the simulated receptive field development for different RF positions (defined by the distance d between reflex and RF position) obtained from three different tracks (Fig. 2.17 B). The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^{R}_{1_{i,j}}$ (the right receptive field). Numbers in the RF denote the total sum of all weights ($\sum$) and the number of learning experiences (LE).

which leads to slight under-steering (panel A). In case the RF is positioned too far away then a much wider response is generated which leads to an over-steering (panel C). This suggests that the position of the RF position affects not only the learning speed (number of required learning experiences) but also the driving behaviour of the robot. We tested this hypothesis by performing 100 experiments were we varied the position of the receptive field, i.e. the distance d between reflex and RF posi-

tion. In order to introduce some variability in the data we changed the direction angle α_0 of the robot every time when replacing the robot to its starting position (see Fig. 2.11 C). Values for α_0 were chosen randomly from a Gaussian distribution with mean $\overline{\mu}_{\alpha_0} = 0$ and variance $\sigma^2_{\alpha_0} = 4$.

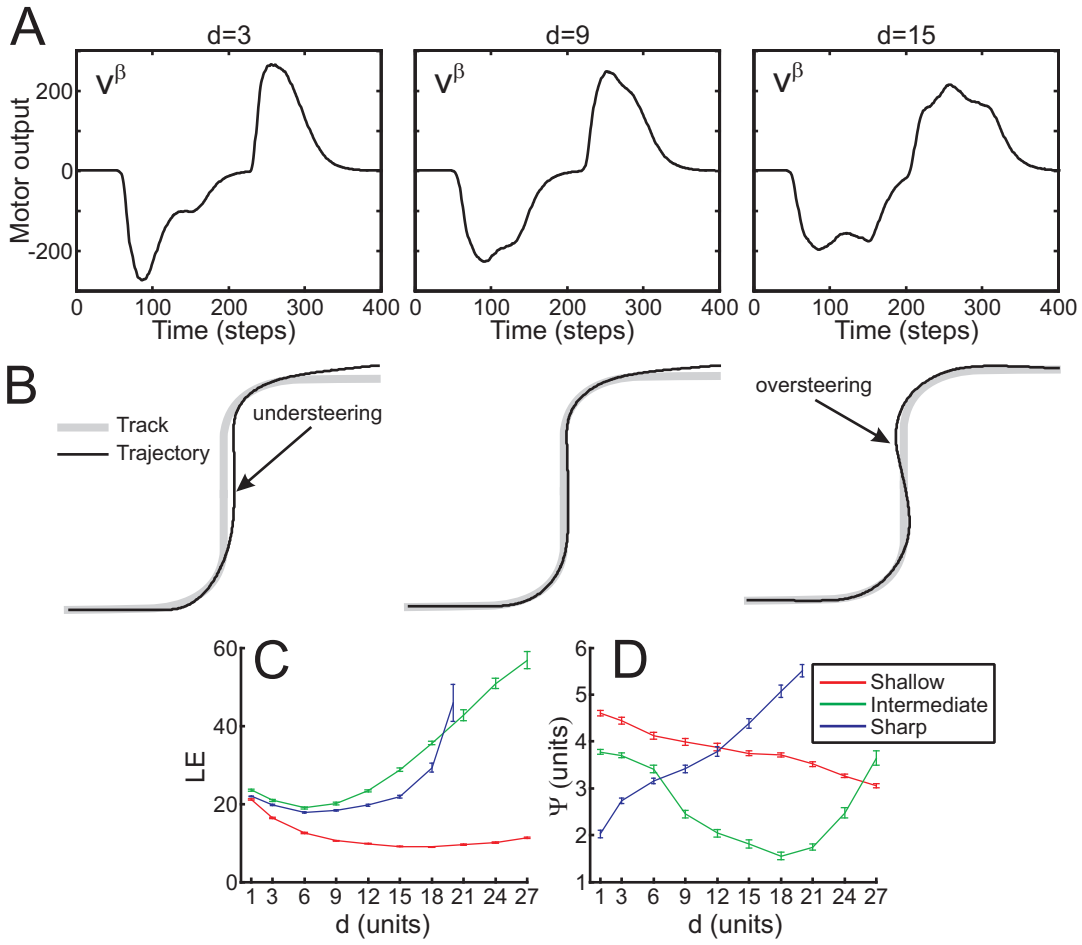

Figure 2.20: **A)** Learnt motor outputs v^β (left side) generated by the corresponding receptive fields (see Fig. 2.19) obtained on an intermediately steep track for different RF positions. Position of the RF is defined by the distance d between reflex and RF position (Fig. 2.17 A). **B)** Corresponding driving trajectories of the robot are shown for each case (panel A). **C-D)** Statistics for different RF positions obtained from 100 experiments. **C)** Number of required learning experiences (LE) and **D)** robot's deviation from the track after learning Ψ. Average together with confidence intervals (95%) is plotted for each case.

Results of the experiment as described above are presented in Fig. 2.20. Average

number of required learning experiences (LE) is shown in panel C where we can see that, indeed, there is an optimal RF position for a specific track with respect to the number of LEs. The fastest learning for the shalow track is obtained when d is between 15 and 18 whereas for the intermediately steep track and the sharp track the optimum is at $d = 6$. This is due to the fact that for the shallow track better correlations between RF inputs and reflex is obtained when the RF is further away from the reflex whereas for the sharper tracks better correlations are obtained when RF is placed closer to the reflex. Note that for the sharp track there were no more correlations obtained between RF and reflex inputs for $d > 20$, so learning was not possible anymore.

The influence of the RF position on the driving accuracy is shown in Fig. 2.20 D. Here we define accuracy Ψ as the average deviation of the robot's position from the track and it is calculated as shown in appendix A.3. We observe that the best accuracy with respect to track following for the shallow track is obtained when the RF is far away from the reflex ($d = 27$) whereas for the sharp track the best accuracy is obtained when the RF is placed as close as possible to the reflex ($d = 1$). This is due to the fact that on the sharp track the robot is cutting corners (see Fig. 2.18 C), and the further away the RF is from the reflex the more the robot cuts the corners since it starts to turn earlier. In case of the shallow track the robot is reacting to late and it is under-steering (see Fig. 2.18 C) if the RF is placed too close to the reflex. On the intermediately steep track the minimal deviation from the track is obtained then the RF is placed at the distance $d = 18$. Results suggest that there is an optimal RF position with respect to learning speed and driving accuracy for a specific track.

2.9.3 Dependence on the input filter

In the next step we looked at how receptive field pattern depends on the input filters. As described above we use filters to prolong our inputs in order to enable correlations between predictive (x_1) and reflexive inputs (x_0). Our filters $h_{0,1}$ are characterised by two parameters: the frequency $f_{0,1}$ and the damping Q. Here we analysed only the influence of the filter frequency, where lower frequencies correspond to the wider filter responses and higher frequencies correspond to the narrower filter responses. Note that we kept the reflex filter h_0 always the same and only varied the filter-bank h_1 of RF inputs. Results of the receptive field development for different input filters (for the parameters see caption of Fig. 2.21) obtained on the intermediately steep track are presented in Fig. 2.21 A-B. Receptive fields are shown in panel A where we can observe that the filters influence only the width of RF pattern, but have no impact on location and orientation of the RF pattern. Filters with the relatively narrow response ($f = 0.2$) lead to narrower RF pattern compared to the wider response filters ($f = 0.075$) which produce wider RF pattern. As a consequence RF pattern will shape the motor output in a similar way such that in case of narrow filters the motor output decays immediately after it reaches its maximum value whereas for

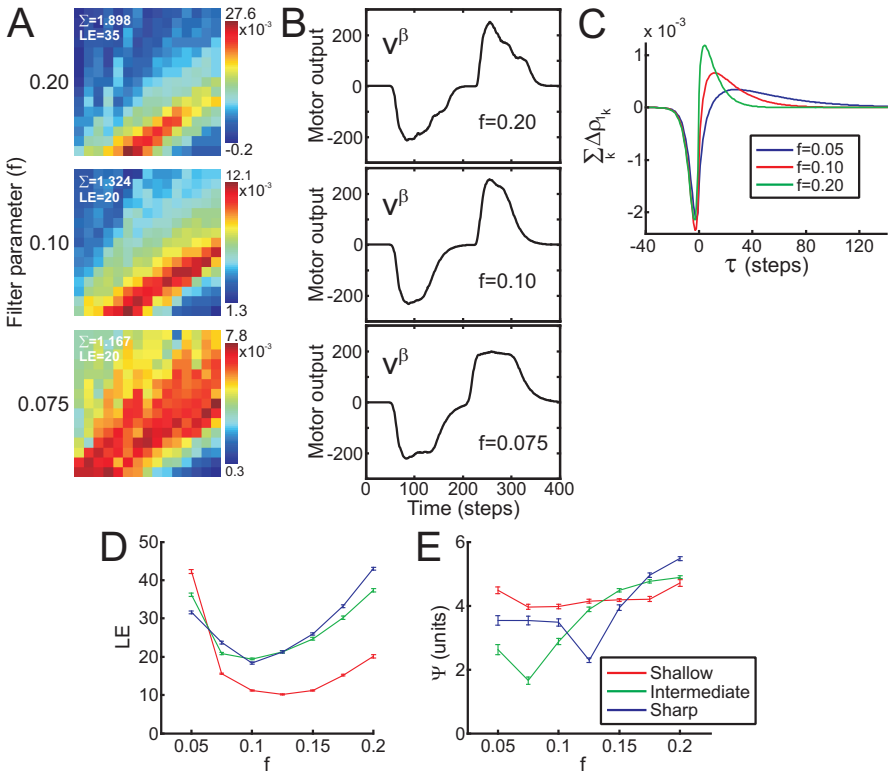

Figure 2.21: Results of the simulated receptive field development for different input filters obtained on the intermediately steep track. We used a filter bank h_1 of ten filters to filter receptive fields inputs $x^{L,R}_{1_{i,j}}$ given by parameters $f_1 = f/k, k = 1 \ldots 10, f = [0.2, 0.1, 0.075]$, whereas for the reflexive input x_0 we used one filter h_0 with the parameter $f_0 = 0.05$ for all three cases. Damping parameter of all filters was $Q = 0.6$. **A)** The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^{R}_{1_{i,j}}$ (the right receptive field). **B)** Learnt motor outputs v^{β} generated by the corresponding receptive fields (see panel A). **C)** Weight change curves for different filter parameters $f = [0.05, 0.1, 0.2]$. Here we plot the total weight change over all ten filters in the filter-bank versus time difference τ between inputs x_1 and x_0, where $\tau > 0$ if x_1 comes before x_0, and $\tau < 0$ if x_0 precedes x_1. Statistics for different input filters obtained from 100 experiments. **D)** Number of required learning experiences (LE), **E)** robot's deviation from the track after learning Ψ. Average together with confidence intervals (95%) is shown for three different tracks.

wide filters the motor output stays relatively flat for a while until it starts going back to zero (see top and bottom panel in Fig. 2.21 B). The RF pattern dependence on

the input filters can be explained by the weight change curve (Porr and Wörgötter, 2003a,b) which is shown in panel C, where we plot weight change against the time difference τ between inputs x_0 and x_1 for three different filter-banks. Note that here we plot the total weight change $\sum \delta_{1_k}$ over all ten filters in the filter-bank. From the weight change curve we can see that for $\tau > 0$ (x_1 precedes x_0 in time) we obtain positive weight change whereas for $\tau < 0$ (x_0 precedes x_1) negative weight change occurs. We also observe that the interval of positive τ values, where a positive weight change is obtained, is increasing if we use filters with wider response (lower frequency) which as a consequence results in a wider RF pattern.

Statistics from 100 experiments showing the impact of the input filter on the speed of learning (number of required learning experiences) and the driving accuracy are shown in Fig. 2.21 D, E. As in the previous experiments we varied the direction angle of the robot at its starting position in order to introduce some variability in the data. Values of α_0 were chosen randomly from a Gaussian distribution with mean $\overline{\mu}_{\alpha_0} = 0$ and variance $\sigma^2_{\alpha_0} = 4$. The fastest learning for the shallow track is obtained when $f = 0.125$, and for the intermediate and sharp track the fastest learning is obtained when $f = 0.1$ (panel D). This is due to the fact that there exists an optimal filter for a given time difference τ which gives the maximal weight change $\delta\omega$ per learning experience as shown in Porr and Wörgötter (2003a). This can also be seen in Fig. 2.21 C where we can observe that for a given τ for some specific filter-bank we get bigger weight changes as compared to the other filters. In Fig. 2.21 E the influence of the filter on the driving behaviour of the robot is shown. The best driving accuracy for the middle track is achieved when wider filters ($f = 0.075$) are used whereas for the sharp track minimal deviation from the track is achieved when narrower filters are used ($f = 0.125$). This is because the narrower filters produce sharper motor response which leads to a sharper driving trajectory and, as a consequence, the robot does not cut corners so much as compared to wide filters. For the shallow track in this case (given default RF position $d = 8$) filters were not so crucial with respect to driving accuracy and the best driving is obtained when f equals 0.075 and 0.1.

2.9.4 Dependence on the robot's initial position and learning rate

In the third stage we analysed how the robot's initial position influences the formation of receptive field pattern and how this depends on the learning rate. Results showing the influence of the robot's initial position (direction angle α_0) while placing the robot at its starting position are shown in Fig. 2.22. In the experiments a Gaussian distribution of α_0 has been used with mean $\overline{\mu}_{\alpha_0} = 0$ and different variances $\sigma^2_{\alpha_0} = [1, 4, 25]$. Here we plot the resulting receptive fields from 10 experiments for each case obtained on the intermediately steep track. Note that we used a relatively low learning rate $\mu = 0.5 \times 10^{-8}$ leading to ≈ 20 learning experiences on average. Results

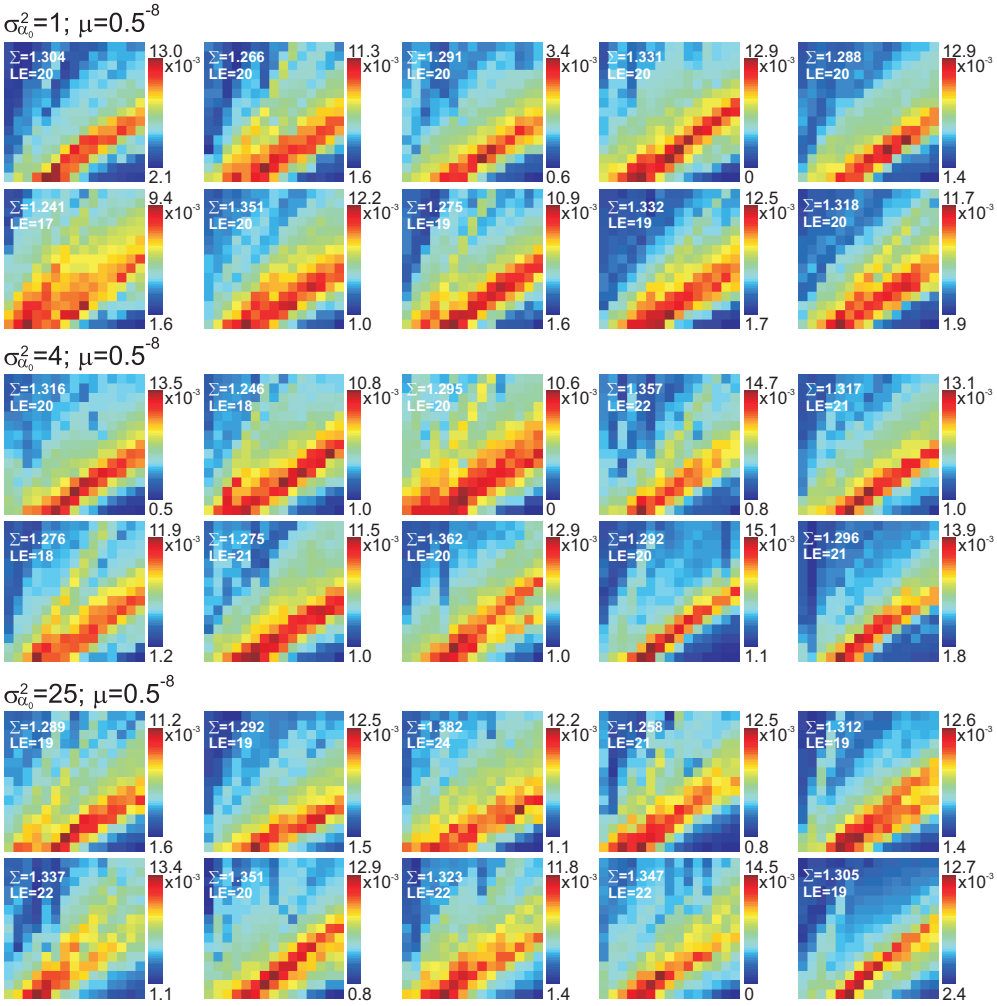

Figure 2.22: Results of simulated receptive field development showing the influence of the robot's direction angle α_0 at it's starting position (see Fig. 2.11 C). Gaussian distribution of α_0 has been used with mean $\overline{\mu}_{\alpha_0} = 0$ and different variances $\sigma^2_{\alpha_0} = [1, 4, 25]$. Learning rate was $\mu = 0.5 \times 10^{-8}$. The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^R_{1_{i,j}}$ (the right receptive field). Numbers in the receptive fields correspond to the total sum of all weights ($\sum$) and the number of learning experiences (LE), respectively.

show that the robot's initial position does not influence RF pattern if relatively low learning rate is used and that after learning most of the RFs have similar structure with more or less the same location, orientation and width of the pattern. By contrast,

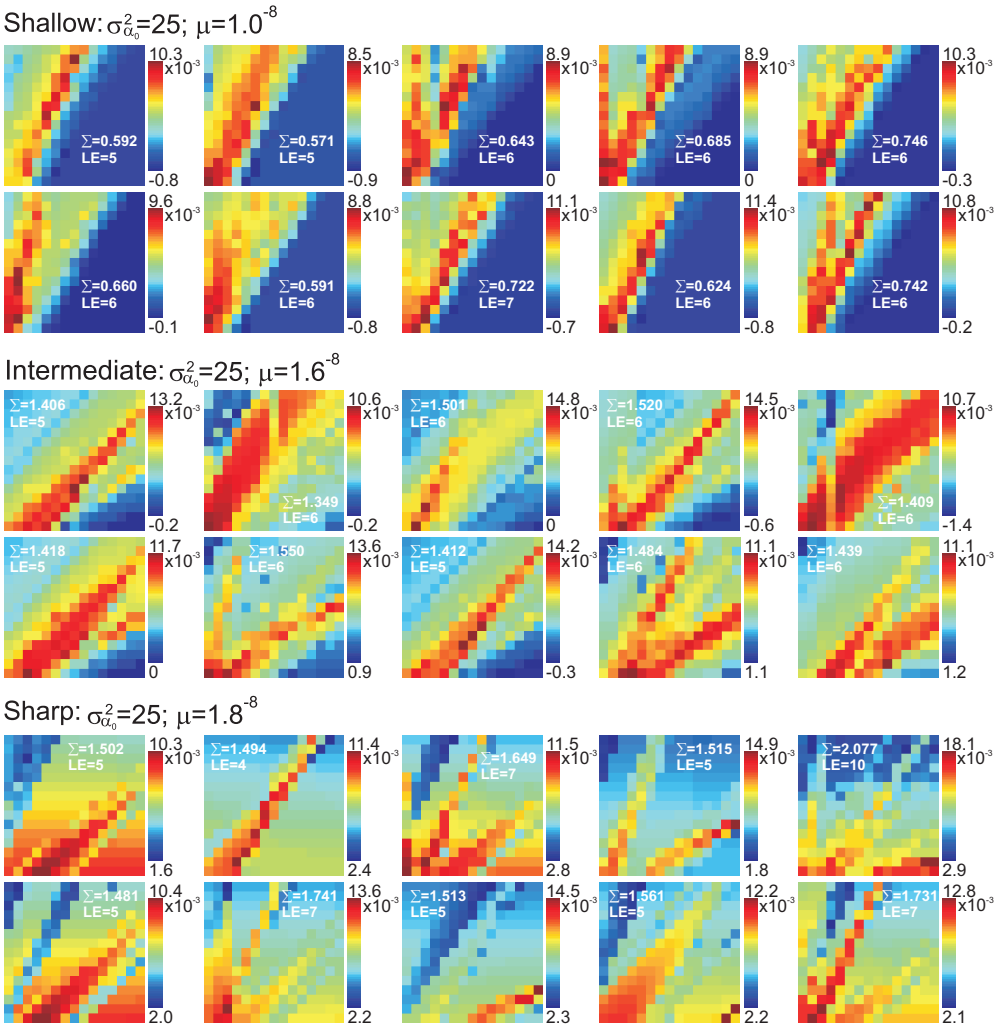

Figure 2.23: Results of simulated receptive field development showing the influence of fast learning obtained on three different tracks. The learning rate was tuned manually for each case in order to achieve more or less the same learning speed (number of required LEs). The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^R_{1_{i,j}}$ (the right receptive field). Numbers in the receptive fields correspond to the total sum of all weights ($\sum$) and the number of learning experiences (LE), respectively.

in Fig. 2.23 we show RFs developed on three different tracks by using relatively high learning rates leading to ≈ 6 LEs on average and high variance ($\sigma^2_{\alpha_0} = 25$). Note that we tuned learning rates manually in order to arrive at more or less the same

learning speed. Here we can see that patterns of the receptive fields vary quite a lot. This can be explained by the fact that in case of rapid learning already the very first experience shapes the pattern of the RF and influences the behaviour of the robot.

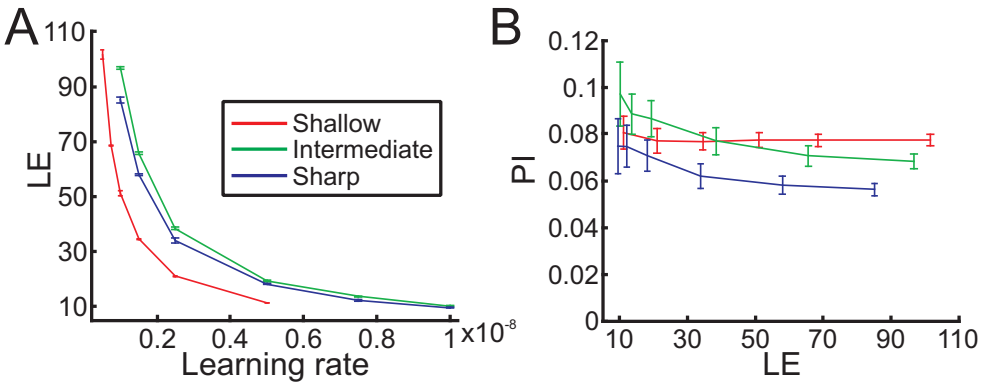

Figure 2.24: Results of simulated RF development obtained from 100 experiments on three different tracks. Values of α_0 are from a Gaussian distribution with mean $\overline{\mu}_{\alpha_0} = 0$ and variance $\sigma^2_{\alpha_0} = 4$. **A)** Number of required learning experiences (LE) versus learning rate μ. Average together with confidence intervals (95%) is shown for each track. **B)** Pattern inconsistency (PI) versus average number of required learning experiences (LE). Average together with standard deviation (SD) is shown for each track.

We also evaluated this statistically and results from 100 experiments are shown in Fig. 2.24. In panel A we plot the number of required learning experiences (LE) against the learning rate μ. For the shallow track we used learning rates $\mu = [0.05, 0.075, 0.1, 0.15, 0.25, 0.5]\times10^{-8}$, and for the intermediately steep and the sharp track we used $\mu = [0.1, 0.15, 0.25, 0.5, 0.75, 1]\times10^{-8}$, since different learning rates are required in order to achieve the same learning speed. As expected we can see that number of LEs is decreasing as the learning rates increases. The pattern inconsistency PI plotted against the number of required LEs is shown in Fig. 2.24 B where we plot the average together with its standard deviation (SD) for each track. The average value corresponds to the structure of the pattern and, as expected, is different for all curvatures. We can observe, as demonstrated qualitatively, that in all cases the variance is decreasing if we use more learning experiences. This suggests that in case of a slow learning process the system's noise is averaging out and a reproducible pattern is obtained.

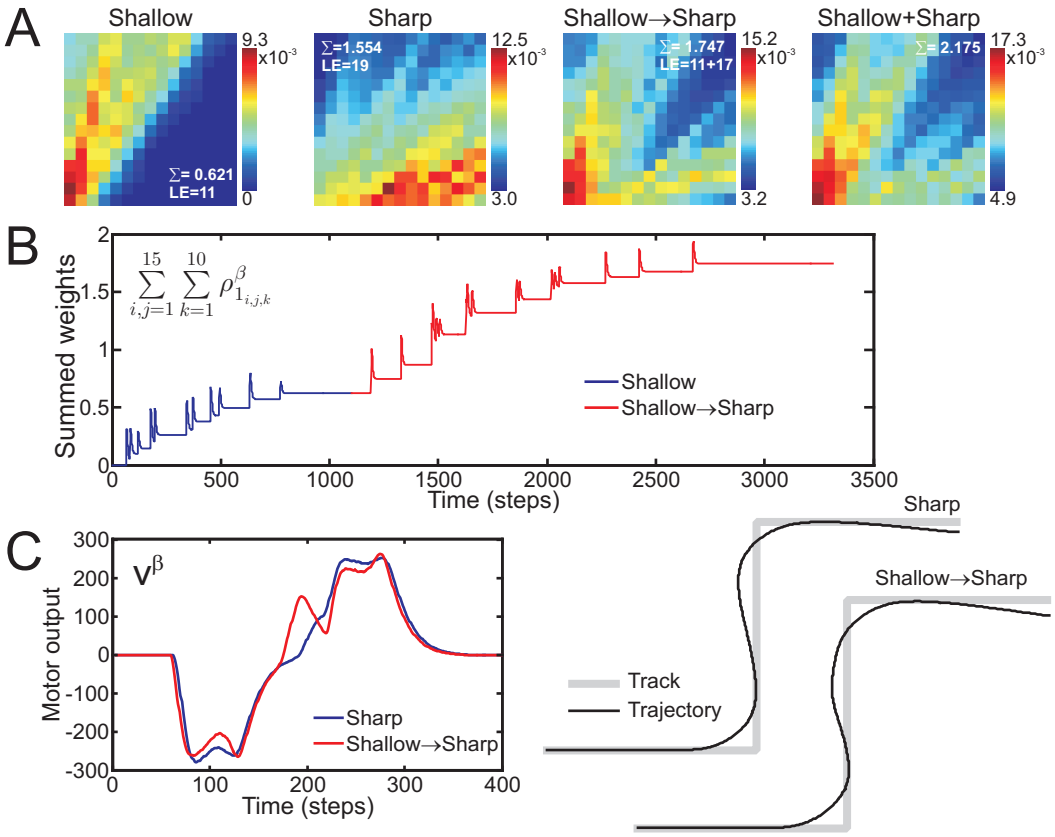

Figure 2.25: **A)** Examples of receptive fields: "Shallow" - RF obtained on a shallow track (see Fig. 2.17 B for different tracks), "Sharp" - RF obtained on a sharp track, "Shallow→Sharp" - RF obtained when the RF learnt on the shallow track was transferred to the sharp track. "Shallow+Sharp" - sum of RFs obtained on the shallow and the sharp track. The diagrams show the summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^{R}_{1_{i,j}}$ (the right receptive field). Numbers in the RF denote the total sum of all weights ($\sum$) and the number of learning experiences (LE). **B)** Development of weights where at first the RF was learnt on the shallow track (blue curve) and was later transferred for the use on the sharp track (red curve). Here we show summed weights over all ten filters in the filter-bank and over all inputs in the RF. **C)** Motor outputs after learning and corresponding driving trajectories: "Sharp" - RF developed and used on the sharp track (control case), "Shallow→Sharp" - RF developed on the shallow track and later used on the sharp track. Default system parameters were used as given in section 2.9.1.

2.9.5 Transfer of RFs from one track to an other

Finally, we checked what happens in case of a transfer of the RF, learnt on one track, for later use on a different track. Here we looked at two extreme cases where at first we developed the RF on the shallow track (see Fig. 2.17 B) and then used it on the sharp track (case 1), and vice versa (case 2). Results for the first case are presented in Fig. 2.25 where weight development is shown in panel B. As expected, weights continue growing when transferred to the sharp track as the reflex is triggered again, because of too small weights and inappropriate RF structure. Additionally, seventeen more learning experiences were required in order to adapt to the sharp track. The resulting RF after such procedure is shown in panel A ("Shallow→Sharp"). The obtained RF structure ("Shallow→Sharp") is similar to the joint structure (sum of two RFs) of RFs obtained on the shallow and sharp tracks independently which means that the composite RF contains patterns inherited from both tracks which overlap each other. We can also observe that a total sum of weights of the composite RF is bigger than that obtained on the sharp track alone which is due to the weight overlapping. As a consequence, the robot over-steers slightly more as compared to the control case (see panel C).

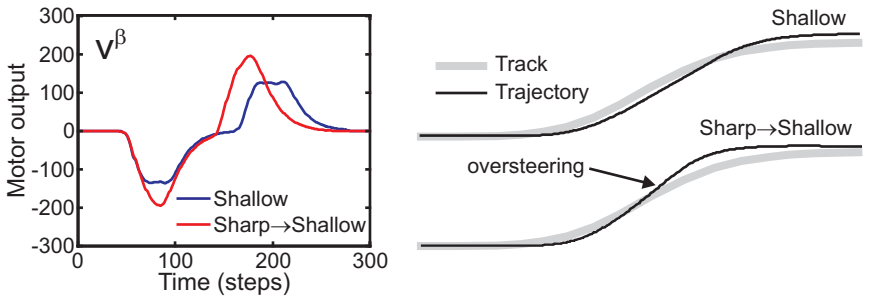

Figure 2.26: Motor outputs after learning and corresponding driving trajectories: "Shallow" - RF developed and used on the shallow track (control case), and "Sharp→Shallow" - RF developed on the sharp track and later used on the shallow track. Default system parameters were used as given in section 2.9.1.

Results for the second case are shown in Fig. 2.26. If we apply the RF learnt on the sharp track to the shallow track then the reflex will not be triggered anymore and the RF structure will not change (even if have non optimal structure) since weight values are relatively high and will cause stronger motor response which will lead to over-steering.

In summary, we have demonstrated that the receptive field pattern adapts to the specific track and depends on many system parameters. Results suggest that there exist optimal system parameters with respect to learning speed and driving accuracy.

We have also shown that the receptive field structure is more robust to noise in case of slow compared to rapid learning.

2.10 Chained learning architectures

Two types of chained learning architectures, namely *linear-chain* and *honeycomb-chain*, were developed by modifying the simple neuronal setup and were simulated and analysed in the open loop case before applying them in the line following task (closed loop case). In the current section we are going to explain both chained architectures and in the next section we will present results when using those architectures in the open- and closed-loop case.

2.10.1 Linear-chain architecture

The schematic of the first type of chained learning architecture, called the *linear-chain*, is presented in Fig. 2.27 A. There is one reflex input x_0 and two predictive inputs x_1 and x_2. The output v^β is used as the reflex input of the neuron γ. The final output v^γ is calculated by

$$v^\gamma = \rho_0^\gamma u^\beta + \rho_1^\gamma u_2, \tag{2.7}$$

where $u^\beta = h_0 * v^\beta$ is a temporal convolution of the output v^β with a resonator h_0. Resonator filters for the x_1 and x_2 are determined by parameters $f_{1,k} = 2.5/k\ Hz$, $k = 1, \ldots, 10$ for the filter-bank h_1. For x_0 and v^β we used one filter given by $f_0 = 1.25\ Hz$. Damping parameter for all filters was $Q = 0.6$. The weights $\rho_0^{\beta,\gamma}$ are set to a fixed value 1, all other weights are initially zero.

2.10.2 Honeycomb-chain architecture

The second type of chained architecture (Fig. 2.27 B) is named *honeycomb-chain*, to which its structure resembles. Output $v^{\beta,1}$ is used as the reflex input of the neuron γ and output $v^{\beta,2}$ as its predictive input. Note, the output $v^{\beta,2}$, similarly to inputs x_1 and x_2, is fed into a filter-bank h_1 of ten different filters as indicated by the dashed lines in Fig. 2.2 C. The final output v^γ is calculated by

$$v^\gamma = \rho_0^\gamma u^{\beta,1} + \rho_1^\gamma u^{\beta,2}, \tag{2.8}$$

where $u^{\beta,1} = h_0 * v^{\beta,1}$ and $u^{\beta,2} = h_1 * v^{\beta,2}$ is a temporal convolution of the output $v^{\beta,1}$ ($v^{\beta,2}$) with a resonator h_0 (h_1). Resonator filters $h_{1,k}$ for $v^{\beta,2}$ are determined by $f_{1,k} = 2.5/k\ Hz$, $k = 1, \ldots, 10$ for the filter-bank h_1, whereas resonator filter h_0 for the x_0 and $v^{\beta,1}$ is given by $f_0 = 1.25/k\ Hz$. Damping parameter for all filters was $Q = 0.6$. Note that we used the same filter parameter f_0 to compute the signal $u_0^{\beta,2}$.

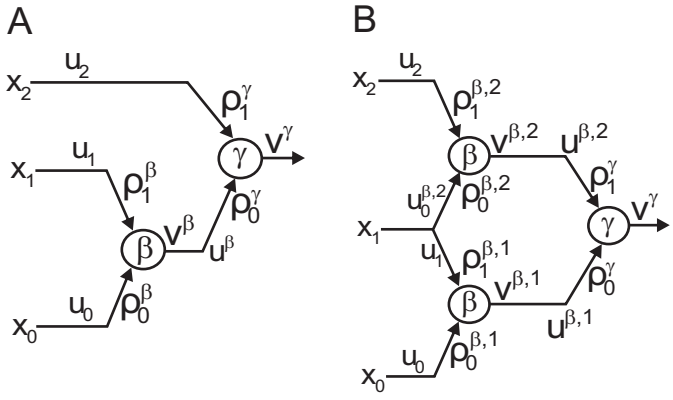

Figure 2.27: Schematic diagrams of chained learning architectures: **A)** linear-chain architecture and **B)** honeycomb-chain architecture. Symbols α, β and γ denote neurons, u denote filtered input signals x, ρ connection weights and v outputs of the neurons.

Weights $\rho_0^{\beta,1}$ and ρ_0^{γ} are set to a fixed value 1, all other weights are initially zero. The connection weight $\rho_0^{\beta,2}$ is given by

$$\rho_0^{\beta,2} = \sum_{k=1}^{10} \rho_{1,k}^{\beta,1}, \tag{2.9}$$

where k denotes the number of the filter in the filter bank. Note that both architectures (Fig. 2.27 A, B) are identical if we set $\rho_0^{\beta,2} = 0$ and $\rho_1^{\beta,2} = 1$.

2.11 Learning with chained architectures

In the following we will present the basic behaviour of the chained learning architectures. First off all we will show simulation results in an open-loop case and later on we will test the chained architectures on a line following task (closed-loop case).

2.11.1 Open-loop case

Inputs for the open loop case were generated as follows. Input x_2 occurs 20 time steps earlier than input x_0 with a variability of up to $\pm$ 5 time steps and x_1 occurs 10 time steps earlier than x_0 with the same variability. This impulse sequence has been repeated every 50 time steps.

Simulation results for the linear-chain (Fig. 2.27 A) are presented in Fig. 2.28 A, B. The variability in the pulse sequences leads to uneven growth. In the open loop

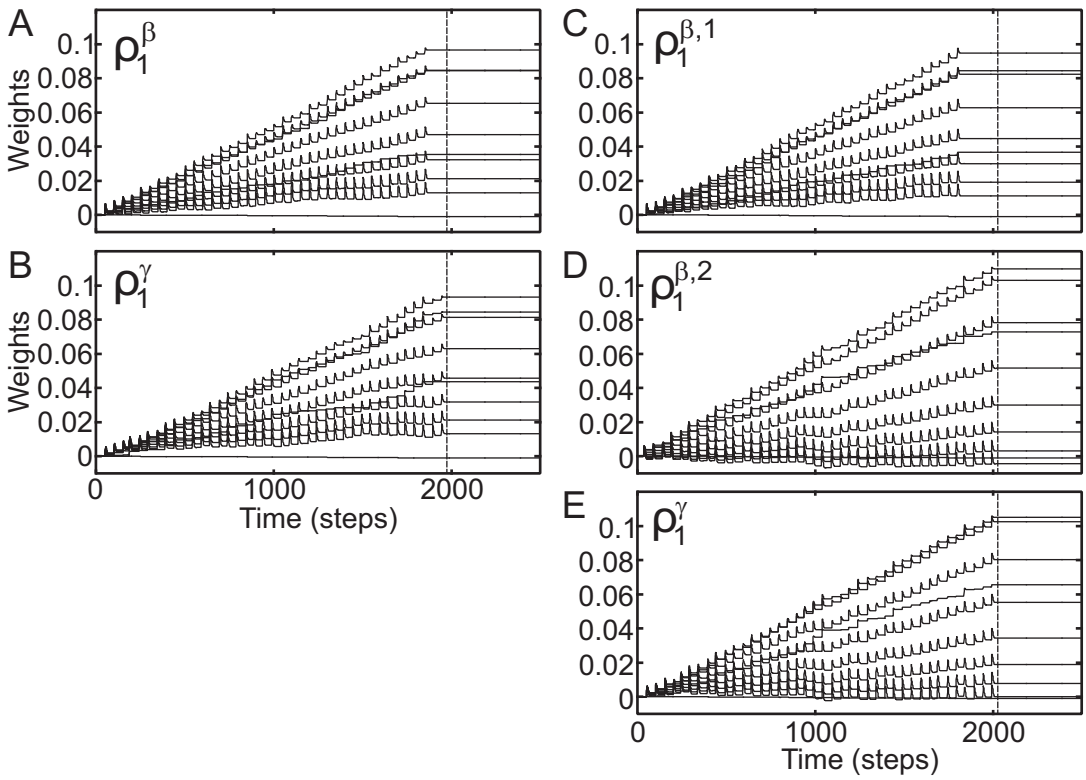

Figure 2.28: Simulation results for chained learning architectures in the open-loop case. Learning rate for both architectures was $\mu = 10^{-7}$. **A, B)** Results for the linear-chain (see Fig. 2.27 A). Connection weights ρ_1^β and ρ_1^γ. Weights ρ_1^β stop growing at the condition $x_0 = 0$ and ρ_1^γ stop growing when $x_1 = 0$. **C-E)** Results for the honeycomb-chain (see Fig. 2.27 B). Connection weights $\rho_1^{\beta,1}$, $\rho_1^{\beta,2}$, ρ_1^γ. Weights $\rho_1^{\beta,1}$ stop growing at the condition $x_0 = 0$, $\rho_1^{\beta,2}$ and ρ_1^γ stop growing when $x_1 = 0$.

case we have to enforce weight stabilisation by setting the respective inputs x_0 and x_1 to zero at some points. We have set $x_0 = 0$ when the sum of weights over all ten filters is

$$\sum_{k=1}^{10} \rho_{1,k}^\beta \geq 0.5, \tag{2.10}$$

and $x_1 = 0$ when

$$\sum_{k=1}^{10} \rho_{1,k}^\gamma \geq 0.5. \tag{2.11}$$

Using this criterion, first the connection weights ρ_1^β stabilise and after some time ρ_1^γ stop to change (as indicated by vertical dashed lines).

Results for the honeycomb-chain (Fig. 2.27 B) are presented in Fig. 2.28 C-E. Similar to the linear-chain architecture to assure weight stability we have set $x_0 = 0$ when the sum of weights over whole filter-bank is

$$\sum_{k=1}^{10} \rho_{1,k}^{\beta,1} \geq 0.5, \tag{2.12}$$

and $x_1 = 0$ when

$$\sum_{k=1}^{10} \rho_{1,k}^{\beta,2} \geq 0.5. \tag{2.13}$$

In this situation first the connection weights $\rho_1^{\beta,1}$ stop to change and later both weights $\rho_1^{\beta,2}$ and ρ_1^{γ} stabilise (as indicated by vertical dashed lines).

2.11.2 Closed-loop case

The chained architectures were applied in the line following task and results similar to those in the simulated open loop case were obtained for both architectures. The physical and neuronal setup of the robot for the chained architectures are presented in Fig. 2.29. The neuronal setup for the linear-chain architecture is presented in panel B and for the honeycomb-chain in panel C. It is similar to those above (see Fig. 2.27), only that we add left and right inputs with inverted signs before this signal finally arrives at neurons β.

Results for the learning task using the linear-chain (Fig. 2.29 B) are presented in Fig. 2.30 A-C and for the honeycomb-chain (Fig. 2.29 C) in Fig. 2.30 D-G. In the first learning trial the motor signal (Fig. 2.30 C) shows three leftward cumulative reflex-predictive reactions and two non-reflexive reactions, as well as two cumulative rightward reactions and three non-reflexive reactions. Note, by chance in this trial the three leftward reflexes were elicited by triggering x_0^L, whereas the two rightward reflexes came from x_1^R. Hence the leftward reflexes where contributing to the change on ρ_1^{β} *and* ρ_1^{γ} (Fig. 2.30 A, B) but not the rightward reflexes, which only contributed to the change of ρ_1^{γ}.

In the second trial only non-reflexive leftward and rightward steering signals occurred and the reflex was not triggered anymore. The driving trajectory is not shown but similar to that obtained by the simple architecture (see Fig. 2.6 D and also video linear-chain.mpg[5]). Weights at a certain neuron stabilise as soon as their corresponding reflex input remains silent. For the linear-chain (panels A-C) this happens earlier for ρ_1^{β} where x_0 becomes zero after about 150 camera frames and later for ρ_1^{γ}, because its reflex input v^{β} remains longer active. Essentially the same is true for the

[5]Videos can be downloaded at *http://sites.google.com/site/ktomsite/driving-robot*

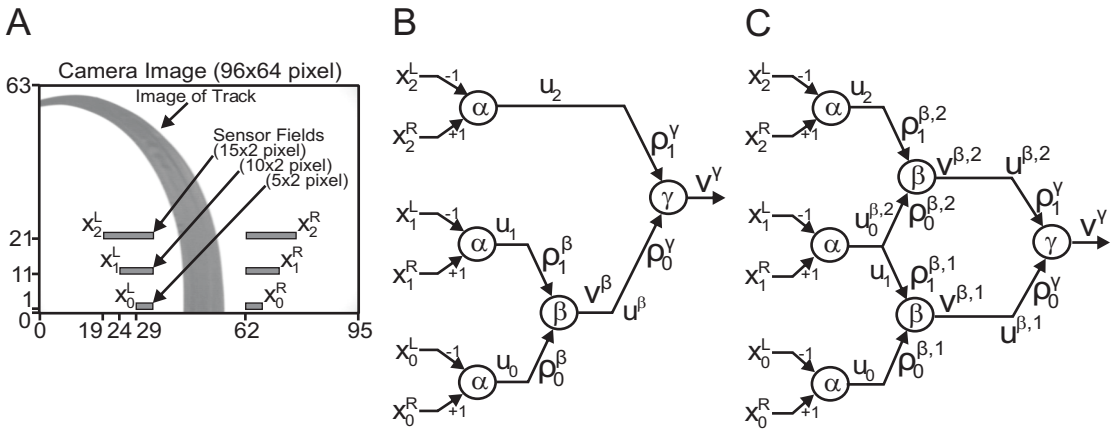

Figure 2.29: Physical and neuronal setup of chained learning architectures. **A)** Camera image with left and right sensor fields marked by $x_2^{L,R}$, $x_1^{L,R}$ and $x_0^{L,R}$. **B, C)** Neuronal setup of the robot for linear-chain and honeycomb-chain architectures respectively. Symbols α, β and γ denote neurons, u denote filtered input signals x, ρ connection weights, and v outputs of the neurons.

honeycomb-chain (D-G). Here $\rho_1^{\beta,1}$ stops growing first, which gets the same reflex input u_0 as ρ_1^{β} in the linear-chain. Convergence of the weights $\rho_1^{\beta,2}$ is controlled by reflex input u_1 which also contributes to the signal $v^{\beta,1}$, being the reflex input to neuron γ. Hence weights $\rho_1^{\beta,2}$ and ρ_1^{γ} behave in the same way and stabilise later (similar to ρ_1^{γ} of the linear-chain).

2.11.3 Statistical evaluation of the chained architectures

We also did simulations using chained architectures in order to make a comparison with the simple setup. The simulation setup for the chained architectures is shown in Fig. 2.31 A. Positions of sensor fields $x_1^{L,R}$ and $x_0^{L,R}$ were fixed (distance was 3 units) and we only varied the position of sensor fields $x_2^{L,R}$ ($d_2 = [2, 3 \ldots 7]$ units).

The influence of the robot's position angle while placing the robot at the starting position is presented in Fig. 2.32 A, B. We plot the success rate in 1000 experiments and the average number of learning experiences (LE) needed to accomplish the task (in successful experiments) against the variance of the distribution of the starting angle σ_α^2. We obtained similar results using the linear-chain (panel A) as compared to the simple architecture where success is slightly decreasing and more reflexes are needed to accomplish the task if we increase the variance σ_α^2. We get a slightly reduced performance as compared to the simple setup (success rate $0.86 < \mathit{success} < 0.96$ for all tracks). Also, as for the using simple setup, more learning experiences are

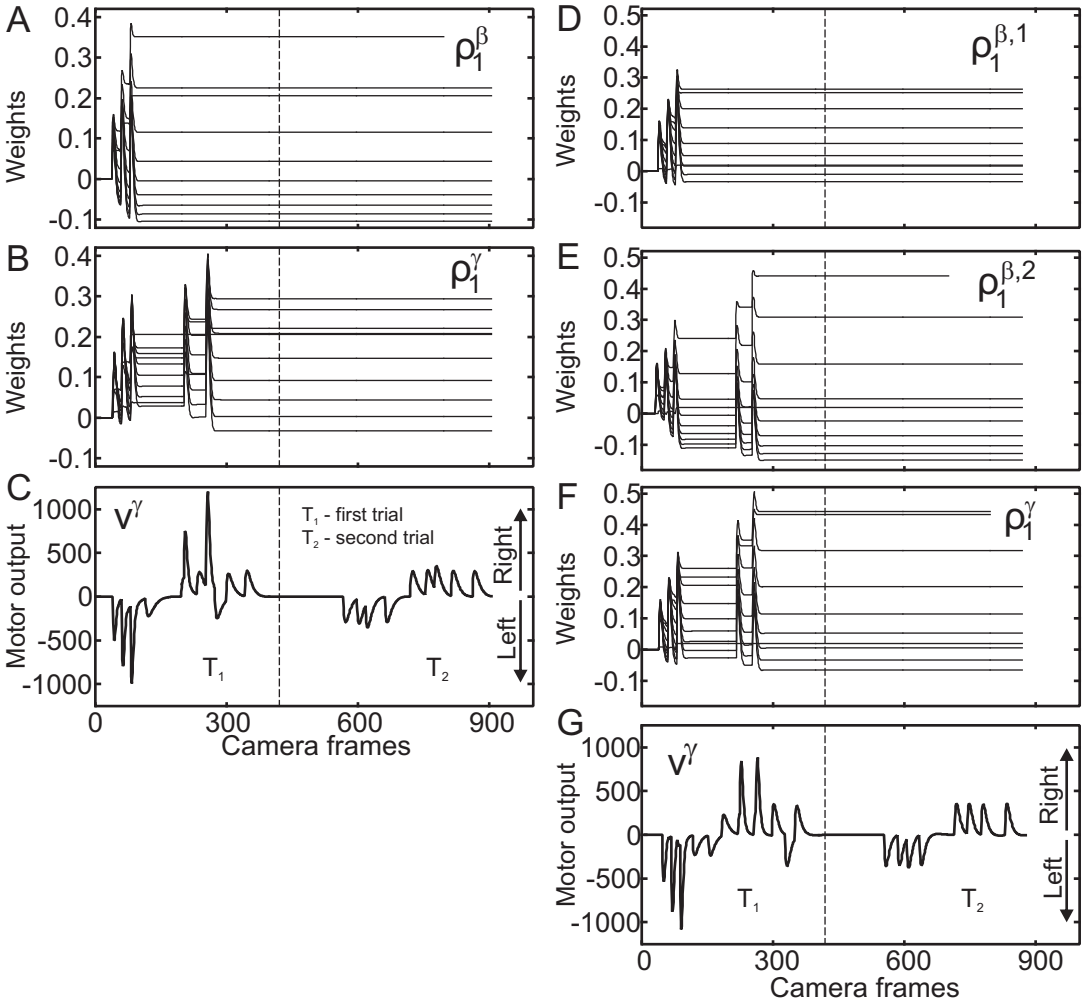

Figure 2.30: Results of the driving robot experiment using chained learning architectures on the intermediately steep track. Learning rate for both experiments was $\mu = 2.5 \times 10^{-6}$. **A-C)** Results for the linear-chain (see Fig. 2.29 B). **A, B)** Connection weights ρ_1^β and ρ_1^γ. **C)** Motor output v^γ. **D-G)** Results for the honeycomb-chain (see Fig. 2.29 C). **D-F)** Connection weights $\rho_1^{\beta,1}$, $\rho_1^{\beta,2}$ and ρ_1^γ. **G)** Motor output v^γ.

required for the sharp track as compared to shallower ones. For the honeycomb-chain (panel D) performance was again lower: success rate $0.71 < success \leq 0.94$ for the shallow and intermediately steep track where for the sharp track we got very low performance (success rate $success < 0.1$). This is due to the fact that the honeycomb-chain architecture is sensitive to the position of the sensor fields. We plot the results of 100 experiments for different positions of the predictor sensor x_2 (we

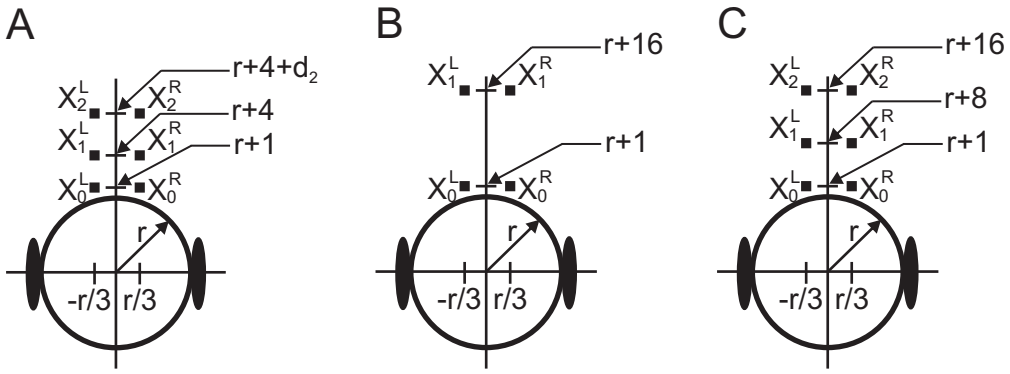

Figure 2.31: **A)** Setup of the simulated robot used for the chained learning architectures. **B, C)** Setups of the simulated robots used for the comparison between simple (B) and chained (C) learning architectures. Dots represent positions of the sensor fields $x_{0,1,2}^{L,R}$, $r = 20$ units is the radius of the robot, $d_2 = [2, 3 \ldots 7]$ units is the distance between sensors x_1 and x_0 (A).

kept positions of x_1 and x_0 fixed) in panel D. Here we can see that we get the best performance for the shallow and sharp track when the distance between x_2 and x_1 is $d_2 = 5$ units (success rate $0.70 \leq success \leq 0.96$ for all tracks) where for the middle track the importance of the position of the sensor fields is not significant (except for the smallest distance between x_2 and x_1 given $d_2 = 2$ units). For the linear-chain setup (panel C) we obtained the same results as for the simple one. Success rate decreases if the distance between inputs is getting larger only for the sharp track whereas for the shallow and intermediately steep track decrease is not significant. We also observed that the number of necessary learning experiences (see panel C, D) is increasing if the distance between x_1 and x_0 is getting larger except for very small distances between x_2 and x_1 when using the honeycomb-chain setup (panel D).

We can summarise that better performance is obtained with the simple setup as compared to the chained architectures. The performance does not crucially depend on the starting angle position. It decreases only slightly if the variance of the starting angle position increases. In general we observed that only for the honeycomb-chain architecture performance depends on the position of the sensor fields (distance between sensor fields). The learning rate does also not affect performance itself. The robot only needs more reflexes to learn the task if we use relatively low learning rates.

2.11.4 Simple vs. chained learning architecture

Previously we summarised that with the simple setup we get better performance as compared to the chained architectures. This is true only for cases where we have good

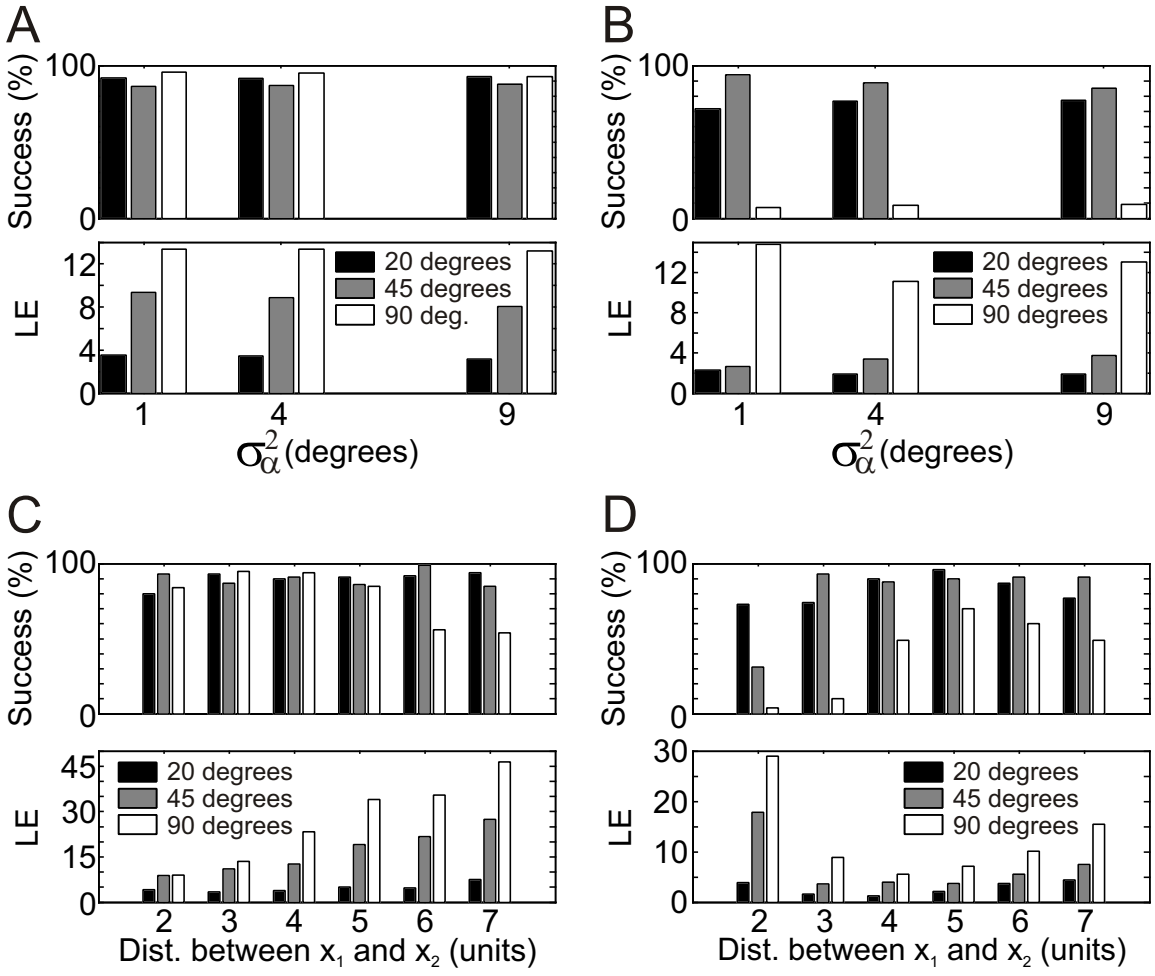

Figure 2.32: Results of the simulation experiments using chained architectures. **A-B)** Success in 1000 experiments and average number of learning experiences (LE) at the motor output neuron γ needed to accomplish the task within successful trials are plotted against the variance σ_α^2: **A)** linear-chain, **B)** honeycomb-chain. Learning rate was $\mu = 5 \times 10^{-6}$ and distance between x_1 and x_0 and between x_2 and x_1 was $d = d_2 = 3$. **C-D)** Success in 100 experiments and average LE plotted against the distance between x_2 and x_1: **C)** linear-chain, **D)** honeycomb-chain. Distance between x_1 and x_0 was fixed and was d=3. Learning rate was $\mu = 5 \times 10^{-6}$ and the variance was $\sigma_\alpha^2 = 4$.

input correlations (small distances between inputs) in the simple setup. Performance decreases if the distance between inputs is very large (see Fig. 2.12 B) for the shallow and intermediately steep track and the robot never managed to steer the sharp curve when the distance between inputs x_1 and x_0 was $d > 8$. However, the robot managed

to steer the sharp curve when chained architectures were used (see Fig. 2.32 C, D) where the distance between inputs x_2 and x_1 was $d_2 > 5$ and between x_1 and x_0 was $d = 3$ (total distance between x_2 and x_0 was > 8).

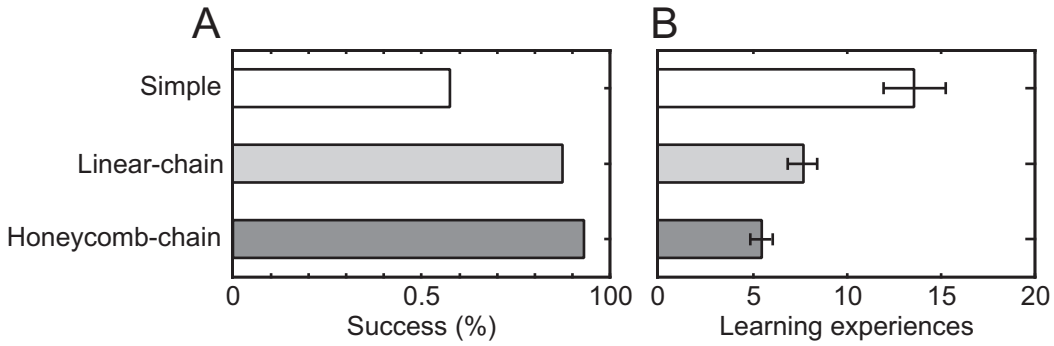

Figure 2.33: Results of the simulation experiments using different learning architectures on the intermediately steep track (45 *degrees*). **A)** Success in 500 experiments. **B)** Average number and confidence intervals (95%) of learning experiences (LE) needed to accomplish the task within successful experiments. Learning rate for all experiments was $\mu = 5 \times 10^{-6}$. Distance between x_1 and x_0 in the simple setup was 15 units whereas distance between x_1 and x_0 and between x_2 and x_1 in chained architectures was 7 and 8 units, respectively.

To test the hypothesis whether chained architectures are advantageous for bad correlations because of sparse inputs we did an experiment where we compared the performance of all three architectures on the intermediately steep track (45 *degrees*). The setup of the input configuration for the simple architecture is shown in Fig. 2.31 B and for the chained architectures in Fig. 2.31 C. Distance between inputs x_1 and x_0 in the simple setup was 15 units and in the chained architectures it was 8 between inputs x_2 and x_1 and 7 between x_1 and x_0 (total distance between x_2 and x_0 was 15 units). A comparison between all three architectures is presented in Fig. 2.33 where we plot the success rate in 500 experiments (panel A) and the average number of learning experiences (LE) within successful experiments together with confidence intervals (95%) needed to accomplish the task (see panel B). From the results we can conclude that chained architectures indeed perform better in this case (success rate for the linear-chain 0.87 and for the honeycomb-chain 0.92) whereas for the simple architecture we obtained a success rate of only 0.57 (see panel A). We also needed less trials to complete learning when using chained architectures as compared to the simple setup (see panel B).

2.12 Development of receptive fields with chained architectures

In the following we will present results on receptive field development using previously introduced chained learning architectures and will show that stable secondary receptive fields can be obtained by using these architectures.

2.12.1 Physical and neuronal setup of the system

The physical and a neuronal setup used for the receptive field (RF) development using chained learning architectures are shown in Fig. 2.34 and are similar to the setup presented above (see Fig. 2.29). Only that here predictive sensor fields $x_1^{L,R}$ and $x_2^{L,R}$ with size of 10×2 px and 15×2 px respectively (Fig. 2.29 A) are replaced by receptive fields with size of 10×10 px (Fig. 2.13 A) where each pixel within the receptive field represents an individual input $x_{1_{i,j}}$ and $x_{2_{i,j}}$ with corresponding plastic synapse $\rho^{\beta}_{1_{i,j}}$ and $\rho^{\gamma}_{1_{i,j}}$ for the linear chain, and $\rho^{\beta,1}_{1_{i,j}}$ and $\rho^{\beta,2}_{1_{i,j}}$ for the honeycomb-chain, respectively. As before we have a symmetrical setup where inputs from the left side have the fixed weight -1 and inputs from the right side $+1$. Synaptic weights ρ_0^{β}, ρ_0^{γ}, $\rho_0^{\beta,1}$ are set to a fixed value of 1 and only weights ρ_1^{β}, ρ_1^{γ}, $\rho_1^{\beta,1}$ and $\rho_1^{\beta,2}$ of all ten filters (see Fig. 2.2 C) change. Note that in the honeycomb-chain $u_0^{\beta,2} = h_0 * \frac{1}{n} \sum_{i,j} x_{1_{i,j}}$, and $\rho_0^{\beta,2} = \sum_{k=1}^{10} \rho^{\beta,1}_{1_{i,j},k}$, where i, j represent indices of active (non zero) input signals and n is the number of active signals.

2.12.2 Learning secondary receptive fields

Results of the receptive field (RF) development using the chained learning architectures on an intermediately steep track are shown in Fig. 2.35. Receptive fields obtained from the linear-chain (Fig. 2.34 A) are shown in panel A, where results for the development of the RF of predictor x_2^R is presented in the top panel and for x_1^R in the bottom panel. We had to place the primary RF closer to the reflex sensor field (at pixel-line 7) and to keep the secondary RF further from the primary RF (at pixel-line 31) in order to achieve appropriate correlation between inputs and stabilise the receptive fields (positions have been found experimentally). One can see that the obtained fields are different. The secondary field x_2^R is much noisier than the primary field x_1^R (pattern inconsistency PI for the secondary and primary RF is 0.1394 and 0.0736 respectively) and has only positive weights whereas the primary RF has positive and negative weights. Note that after learning only the secondary receptive field will drive the steering behaviour of the robot. Connection weights stabilised after five trials (in total nine learning experiences).

Results of the RF development using the honeycomb-chain (Fig. 2.34 B) are shown in Fig. 2.35 B. In this case, in order to achieve appropriate behaviour and stabilise

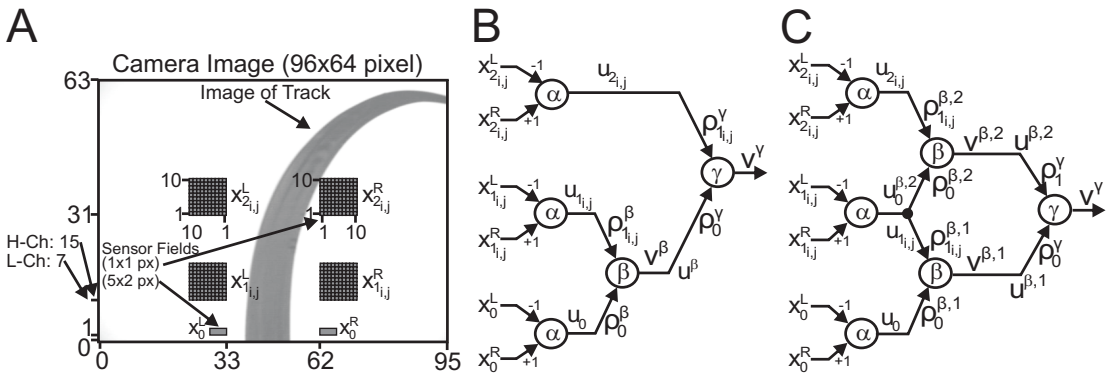

Figure 2.34: Physical and neuronal setup of the receptive field (RF) development using chained learning architecture. **A)** Camera image with left and right sensor fields. The receptive filed positions are denoted by $x^{L,R}_{1_{i,j}}$ and $x^{L,R}_{2_{i,j}}$, where $i = 1 \ldots 10, j = 1 \ldots 10$ are the indices of the RF pixels, and sensor field positions $x^{L,R}_0$. **B, C)** Neuronal setups of the robot for linear-chain and honeycomb-chain architectures, respectively. Symbols α, β and γ denote neurons, u denote filtered input signals x, ρ connection weights and v outputs of the neurons. v is calculated by the method shown in Fig. 2.2 B and its corresponding Eq. 2.1. v^γ is used for steering control and is transformed to the motor output $M^{L,R}$ as given in Eq 2.4.

receptive fields, we placed the primary RF further from the reflex sensor field but closer to the secondary RF (at pixel-line 15). The secondary RF was placed at the same position as for the linear-chain (at pixel-line 31). Both fields have negative and positive weights. In contrast to the linear chain the secondary RF is less noisy than the primary RF (pattern inconsistency PI for the secondary and primary RF is 0.0651 and 0.0914 respectively. Connection weights stabilised after six trials (in total 18 learning experiences).

Because more care has to be taken in positioning the RF, in general, we have found that chained architectures are harder to stabilise as compared to the simple one-neuron architecture (see Fig. 2.13). This is not unexpected and results from the influence of the indirect inputs. Hence the more indirect a signal becomes the more does its correlation structure deteriorate, leading to problems in stabilising the corresponding receptive fields. This cries out for more advanced sensory pre-processing by ways of neurons that are already able to extract more stable correlation patterns. We will discuss possible implications of this statement in section 2.13, but in the current study we are not concerned with this pre-processing problem.

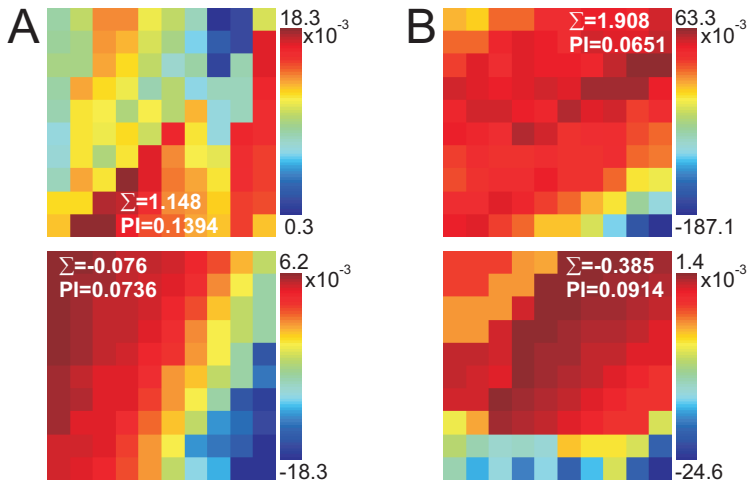

Figure 2.35: Results of the receptive field (RF) development using the chained learning architectures on a intermediately steep track. **A)** Right RFs obtained from the linear-chain (see Fig. 2.34 B). Diagrams show summed weights $\sum_{k=1}^{10} \rho^{\gamma}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^R_{2_{i,j}}$ (top) and summed weights $\sum_{k=1}^{10} \rho^{\beta}_{1_{i,j},k}$ from the corresponding predictor $x^R_{1_{i,j}}$ (bottom). Learning rate was $\mu = 2 \times 10^{-8}$. **B)** Right RFs obtained from the honeycomb-chain (see Fig. 2.34 C). Diagrams show summed weights $\sum_{k=1}^{10} \rho^{\beta,2}_{1_{i,j},k}$ over all ten filters in the filter-bank which receive inputs from the corresponding predictor $x^R_{2_{i,j}}$ (top) and summed weights $\sum_{k=1}^{10} \rho^{\beta,1}_{1_{i,j},k}$ from the corresponding predictor $x^R_{1_{i,j}}$ (bottom). Learning rate was $\mu = 5 \times 10^{-8}$. Note that the values in the receptive field denote the total sum of all weights ($\sum$) and the pattern inconsistency (PI) respectively.

2.13 Discussion

In this chapter we have introduced a specific closed loop robotics system which can adaptively improve its line following behaviour, performing reflex-avoidance learning by ways of replacing late responses to sensor fields at the base of a camera image with earlier ones triggered by sensors higher up in the field of vision. A new learning rule (ICO) has been employed which is able to correlate sequences of temporal events and the system has been tested in a restricted set of scenarios far less complex than those in a real-world navigation task. Thus, the system has been specifically designed for this task and cannot easily be compared with more general navigation systems (see section 2.13.4 below). These restrictions, however, are justified by the focus of this chapter which is two-fold. We have focused on: 1) The development of receptive fields in a closed loop perception-action system and 2) the question of how to chain

temporal sequence learning architectures. Note, more general applications of single module (no chaining) ICO learning are found in (Porr and Wörgötter, 2006, 2003a,b; Manoonpong et al., 2007). These studies should support the general versatility of this type of learning. In the following we would like to discuss how the open- and closed-loop situation compares to biological and other artificial systems, how our methods relate to other approaches for RF-development, and where there are relations to some aspects of reinforcement learning.

2.13.1 Relation of ICO-learning to synaptic plasticity at real neurons

The ICO-learning rule has been chosen because of its robust convergence properties (Porr and Wörgötter, 2006) even with high learning rates. ICO learning changes its weights by correlating inputs, only. This can be interpreted as heterosynaptic plasticity or as modulatory plasticity. In biological systems, pure heterosynaptic learning is only found at a few specialised synapses (mossy fibre, amygdala, see Humeau et al., 2003; Tsukamoto et al., 2003), where the mossy fiber synapse between dentate gyrus and CA3 in the hippocampus can indeed create fast and strong changes similar to those induced by ICO-learning with a high learning rate. More often, however, heterosynaptic influences are thought to be mainly modulatory (Kelley, 1999; Ikeda et al., 2003; Bailey et al., 2000; Jay, 2003). Here we are not really concerned with the possible biological implications of such a learning rule (for a more detailed discussion see Wörgötter and Porr, 2005; Porr and Wörgötter, 2006). Instead we have used it as a tool to employ fast learning in a difficult scenario. This property is visible when learning succeeds after the first trial in keeping the robot on track for an intermediately sharp track, while it does not follow the line if only the reflex alone is employed. Hence, already *during* the first learning reflex synaptic weights adjust quickly and, in turn, immediately influence the output leading to successful behaviour. This behaviour is generically observed for the ICO-rule, which thereby approaches the limit of one-shot learning in a stable behavioural domain (Porr and Wörgötter, 2006), provided the input correlations are robust enough. With noisier inputs, a lower learning rate needs to be applied, which will lead to a more gradual development of the weights, required also in the case of RF-development in order to arrive at structured receptive fields.

2.13.2 Relation to secondary conditioning

In this chapter we were concerned with designing simple chained architectures of our learning modules. This was inspired by second-order conditioning in animals (Rescorla, 1980; Gewirtz and Davis, 2000) and humans (Jara et al., 2006). Secondary conditioning requires a similar situation where the primary correlation between conditioned (early, CS) and unconditioned (late, US) stimulus is first learnt and then in

a second learning stage is replaced by a newly learnt correlation between secondary conditioned stimulus (yet again earlier) and CS. This situation is conceptually similar to our chained learning units and the same problems, for example less reliable correlation patterns, arise in both situations.

2.13.3 Closed-loop context: combining control and learning

Biological systems are generally operating in close conjunction with their environment. This so-called ecological embedding has already been discussed by theoreticians very early as also essential for autonomous artificial agents (Ashby, 1956; McFarland, 1971; Wiener, 1961). On the more practical side the work of W.G.Walter was probably the first to create an operational, autonomous cybernetic control system when he built his two robots Elmer and Elsie. These machines could already perform homing as well as different forms of photokinesis (Walter, 1950). In the following the ecological perspective had been widened most notably by the work of Braitenberg (1984) on his "vehicles" and for invertebrates by Webb (2002).

In most of the older work typical feedback loop control systems had been built, which do not adapt but instead react to a stimulus by ways of reflex-like behaviour. Stable feedback loop control is in itself a difficult problem in particular when there are multiple inputs and outputs. It is however known that even very simple animals can learn and adapt to new situations. Hence we are now faced with the augmented problem of how to combine Control with Learning in a stable way. Specifically we are confronted with the question how animals arrive at useful, reproducible and, hence, stable behavioural patterns, while they are at the same time able to learn "something new". Recently Verschure suggested that such systems should contain several layers for control and learning: At the bottom a "reactive layer" performing pure reflex-based control, one above an "adaptive layer" performing predictive learning much in the sense of classical or operand conditioning and finally on top a "contextual layer" for higher level adaptation (DAC-architecture, see Verschure and Althaus, 2003). Here we are concerned with the first two layers only.

There is another class of learning setups, called feedback-error learning (FEL, see Gomi and Kawato, 1993; Nakanishi and Schaal, 2004), which appear to be related to closed-loop ICO. However, in contrast to ICO learning, FEL does not use additional predictive inputs $x_1, x_2, \ldots$ to compensate for a disturbance. It rather improves the feedback loop itself by using the signals which are available to the (late) feedback system. A simple example is feedback loop which is set up as an over-damped system (PI controller) so that the reaction of the loop to a disturbance or a change in the set-point leads to a low pass filtered impulse response of the system. With the help of FEL the reaction could be made faster by adding an adaptive controller which receives a copy of the disturbance itself or the output of the feedback controller. Because we have got an over-damped system, FEL would learn to become the derivative of the disturbance. In other words, FEL would adaptively learn to add the "D" to a PI

controller. ICO or ISO learning, however, are fundamentally different because they use the derivative as a predictor to learn another predictive input which is then used to eliminate the disturbance and eventually eliminates the feedback loop itself. FEL on the other hand does not replace the feedback loop by a forward controller but rather improves the performance of the feedback controller as such.

In all such architectures, however, one must ask how in the process of learning synaptic weights are stabilised in conjunction with behavioural success. Stability in our approach rests on the assumption that the reflex eliciting signal (x_0) really represents an error signal. Hence, ICO-learning stabilises as soon as this error signal is eliminated as has been rigorously shown in Porr and Wörgötter (2006). On the behavioural side this, however, means that the reflex has been functionally eliminated and has now been successfully replaced by an earlier anticipatory action. This property allows controlling the homoeostasis of learning and behaviour at the same time, which is difficult to achieve with most other architectures.

2.13.4 Comparison to other approaches on receptive field development

The development of visual receptive fields has been in the centre of research interest during the last decade and it had been shown that cortical receptive fields can develop following a sparseness principle and essentially implementing independent component analysis (Olshausen and Field, 1996; Bell and Sejnowski, 1997). These studies have been followed by many others focusing on specific sub-aspects in the receptive field development (Blais et al., 1998; Weber and Obermayer, 1999; Hurri and Hyvärinen, 2003; Körding et al., 2004). However, only very few attempts exist to develop RFs from signals of a behaving agent. Einhäuser et al. (2002) had shown that receptive fields develop from natural stimuli by employing an objective function to control the development (Vinje and Gallant, 2000). Recently this work has been extended employing exploration in a Khepera robot to drive the development of receptive fields in a complex multi-layered neuronal system (Wyss et al., 2006). At the different layers receptive fields are generated using an objective function to control their development which also leads to stability of the RF structures after about three days of real-time exploration. However also in this study the responses from the network are not used to drive the behaviour. Hence these approaches, while conceptionally elegant, remain in the tradition of the older studies (Olshausen and Field, 1996; Bell and Sejnowski, 1997) and do not employ direct behavioural feedback.

Pomerleau (1991, 1995) in his ALVINN system (Autonomous Land Vehicle In a Neural Network) learnt to steer a vehicle in response to visual input from a forward facing camera. The ALVINN system uses a single hidden layer feed-forward neural network which applies the back-propagation learning algorithm to learn an appropriate behaviour similar to and taught by human reactions. It differs from our approach

because it is based on supervised learning and the learning also does not take place in a complete closed loop setting. Obtained receptive fields (weight matrices of hidden layers) do not get finally stabilised with this type of learning.

For a different system, Arleo and Gerstner (2000) have shown that place fields, similar to those in the hippocampus (O'Keefe and Dostrovsky, 1971), can be developed in a robot. This robot does also explore its environment, but the motor control is again not derived form the network.

Hence, these models differ strongly from our approach because they are still open loop. This is different for a recent study by McKinstry et al. (2006) who were able to close the loop and derive path-following behaviour in a robot that is driven by a complex multi-layer neuronal system supposed to mimic parts of the cerebellar system. The system learns, as in our case, reflex avoidance. This is done by neurons in the simulated Inferior Olive which adapt following a Hebbian learning rule. Synaptic weight matrices (receptive fields), develop at several stages in the network, but it appears that this type of learning will not lead to their final stabilisation.

2.13.5 Limitations of our approach

We are, however, also facing some problems with our simple setup. First and foremost, we found it difficult to assure convergence when receptive fields are to far up or down in the camera field. A somewhat more detailed analysis of this problem showed that in these cases the correlation structure between the signals x_0 and x_1 is distorted. If the fields are too high up, correlations become weak, if it is too far down, they become often temporally inverted. To be able to obtain convergent learning one has to make sure that the correlations between the inputs are reliable enough, which is not the case in these situations. On similar grounds we found it much harder to stabilise chained receptive fields. It is clearly an overly simplified approach to use raw camera signals to develop receptive fields, which was here meant to demonstrate basic principles of closed loop learning. Hence in a more realistic setup one should already start with an input pre-processing system that better extracts the necessary correlations. This could, for example, be achieved by already starting with network units that have a predefined spatio-temporal receptive field. It seems that visual receptive fields in the cortex are specifically designed to optimally represent natural images (Simoncelli and Olshausen, 2001; Felsen et al., 2005) and that essentially a spatial input decorrelation is performed by the cortical population response (Daugman, 1989; Vinje and Gallant, 2000). Hence, untroubled by spurious spatial correlations, such cortical receptive fields might be very well suited to encode stimulus driven *temporal* correlations, which arise from scanning over a visual scene.

2.13.6 Some relations to reinforcement learning

Parts of this chapter were concerned with designing simple chained architectures of our learning modules. This was motivated by the fact that the sensor information in animals internally progresses along many stages until a motor output is generated. It is unknown, how such complex sensor-motor loops maintain behavioural stability, let alone behavioural stability during changing synaptic strengths between these different stages. Our approach is to some degree related to reinforcement learning, not so much to machine learning methods like Q-learning (Watkins, 1989; Watkins and Dayan, 1992), but rather to Actor-Critic loop architectures (Witten, 1977; Barto et al., 1983; Barto, 1995), which have been employed in simulated neural systems. Indeed, if one uses the x_0 signal as a reward one can create a structural similarity between some of these algorithms and our ICO-rule (for a detailed comparison see Kolodziejski et al., 2009). Also, we note that the strict state- and action space tiling used in traditional Q-learning approaches has in some approaches been replaced by more adaptive self-defining processes, which span the state- and action space through exploration (Jodogne et al., 2005; Agostini, 2004) making these algorithms better compatible to neuronal architectures.

Indeed, some Actor-Critic algorithms have been also used to guide the learning of biologically-inspired agents (Montague et al., 1995; Suri and Schultz, 1998; Schultz and Suri, 2001; Niv et al., 2002) but – to our knowledge – it has not been attempted to chain Actor-Critic loops so far. Apart from the fact that there is no generic "recipe" existing, the problem may be even more fundamental. Actor-Critic architectures usually rely (in their Critic) on the TD-algorithm (Sutton, 1988; Sutton and Barto, 1998) to assess the value of an action of the Actor. The prediction error δ in TD-learning equals zero as soon as the output v accurately estimates the future expected reward $r(t+1)$ using: $\delta(t) = r(t+1) + v(t+1) - v(t)$. To fulfil this convergence condition, the output v needs to take on a certain value (output control). In any single loop architecture, outputs will be fairly directly transferred to inputs by ways of the environment (e.g. Fig. 2.3). In a nested or chained loop, however a problem may arise. To guarantee the convergence of each individual stage of the chain its output needs to be directly conveyed backward to compare it to the reward, which, necessarily is an input to the regarded stage. Effectively this amounts to some kind of error back-propagation, a commonly used principle in artificial neural networks (McClelland et al., 1987), but hard to justify in biological networks, where the role of internal feedback does not seem to be related to any error back-propagation mechanism. Architectures based on our correlation based learning rule(s) perform strict input control, because they converge as soon as the error signal of the reflex, x_0, equals zero, regardless of the value of the output. This condition, hence, does not require error back-propagation and may prove to be easier to handle for the design of more complex nested of chained loops as compared to Actor-Critic architectures (Wörgötter and Porr, 2005).

3

Behavioural Analysis of Closed-loop Learning Systems

3.1 Introduction

In the previous chapter we talked about closed-loop learning systems were we introduced simple layered learning architectures. By applying such learning architectures on a driving robot we were able to develop primary and secondary receptive fields, at the same time guaranteeing stable behaviour of an agent. In this chapter we will continue to further investigate closed-loop learning systems. Here we are concerned with an analysis focusing on the dynamics of learning systems, where we will also apply some of the now introduced system measures to analyse receptive fields from chapter 2.

As discussed earlier, behaving systems form a closed loop with their environment where sensor inputs influence motor output, which, in turn, will create different sensations. Simple systems of this kind are reflex based agents which react in a stereotyped way to sensory stimulation, either by a retraction or an attraction reflex (Braitenberg Vehicles, Braitenberg, 1984). If the environment is not too complex, one can describe (linear) systems of this kind also in the closed loop case by methods from systems theory. For this, the transfer functions of agent and environment need to be known and the characteristics of the control-loop also needs to be taken into account.

The situation becomes much more complicated as soon as one allows the controller to adapt, for example by learning. Now the transfer function of the agent changes over time and thereby its interaction with the world, which not only influences its behaviour but also the learning, resulting in an ongoing change of the behaviour. It is exceedingly difficult to describe such non-stationary situations.

Two very general questions arise here. 1) To what degree is it possible to describe the temporal development of such adaptive systems using only knowledge about their initial configuration, their learning mechanism and knowledge about the structure of the world? and 2) Given a certain complexity of the world can we predict which system from a given class would be the best (in some well defined sense)? Clearly these questions are too general to be answered without constraining "system" and

"world" much more. But even when doing so, the problem remains intricate due to the non-stationary closed loop configuration.

As in the previous chapter, we will focus on systems that perform differential hebbian learning, related to spike-timing dependent plasticity (Markram et al., 1997; Saudargiene et al., 2004, 2005), for the learning of temporal sequences of paired sensor events. Differently from the previous chapter here we are going to use an agent that learns to avoid obstacles as shown in Fig. 3.1 B. Fig. 3.1 A shows the general control diagram for such system which is similar to the one introduced earlier (see Fig. 2.3 B). In chapter 2 we assumed that transfer functions of the closed-loop system are constant, however, while the agent interacts with its environment these transfer function change. For example, the question arises whether the delay τ (Fig. 3.1 A) between the predicting event x_1 and the reflex trigger x_0 would change while learning to avoid obstacles. Thinking of an obstacle-avoiding robot, intuitively one would expect τ to get longer as the growing influence of x_1 should lead to a later and later triggering (by x_0) of the reflex until it is finally fully eliminated. This is shown in Fig. 3.1 B trajectory (1) versus trajectory (2), where the robot beetle depicted uses two sets of antennas (short and long) for near (x_0) and far (x_1) sensing, respectively. The intuition of a growing τ is alluring, but just shows how even in the simplest cases our understanding of such adaptive systems can go astray. Because, as shown below and contrary to our naive intuition, such systems are better described by a τ which first grows and then shrinks again back to essentially its starting value. A deeper look into the development of these systems allows understanding why this happens and we can even derive an analytical approximation for the weight development in these cases.

In the second part of this chapter we define energy, input/output ratio and entropy measures for these systems and measure them in environments of different complexity. Using these measures we will first show that learning equalises the energy uptake across agents *and* worlds. Strong differences which initially exist are being levelled out during learning. However, when judging learning speed and complexity of the resulting behaviour one finds a trade-off and some agents will be better than others in the different worlds tested.

The analysis of closed-loop systems is a well established field in the engineering sciences, which also investigates "adaptive controllers". Very little, however, is known about adaptive controllers which interact with their environment by shaping non-stationary dynamics through their own learning (see section 3.10). It had been shown that Shanon's Information Theory (Shannon, 1948) can be applied for perception-action loops (Ashby, 1956; Tishby et al., 1999; Touchette and Lloyd, 2000). Few other studies exist that try to analyse closed loop systems from an agent-perspective asking about the information processing properties of such system in the context of what would be beneficial for the agent itself (Klyubin et al., 2007, 2008; Lungarella et al., 2005; Lungarella and Sporns, 2006; Prokopenko et al., 2006). Even fewer attempts exist that consider learning (Lungarella and Sporns, 2006; Porr et al., 2006). The work

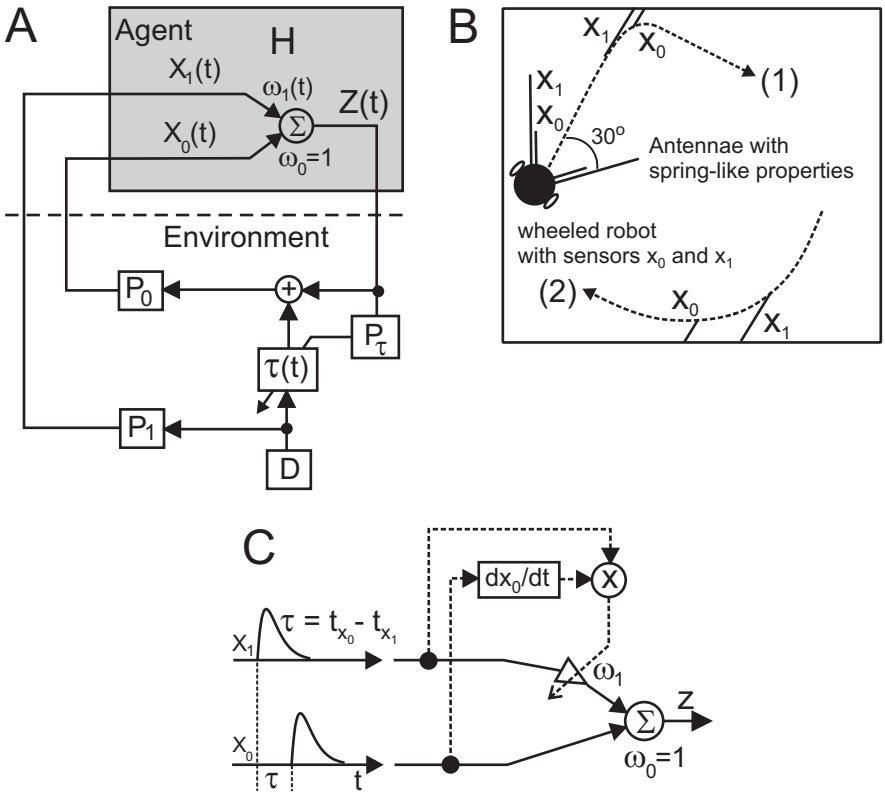

Figure 3.1: **A)** Schematic diagram of the closed-loop learning system with inputs x_0 and x_1, connection weights ω_0 and ω_1 and neuronal output z. P_0 and P_1 denote the reflexive and the predictive pathway respectively. D defines the disturbance, where τ is the time difference between inputs x_0 and x_1 as shown in panel C. **B)** Robot setup with short antennas (reflexive inputs, x_0) and long antennas (predictive inputs, x_1). The diagram shows a situation with an increase of the time difference between far- and near-sensor events during learning process ($\tau_2 > \tau_1$), depicted by the respective distance between the little (solid) triggering lines $x_{0,1}$ from trajectory (dashed) to wall. **C)** Schematic diagram of the input correlation learning rule and the signal structure (ICO, Porr and Wörgötter, 2006; Kulvicius et al., 2007).

presented in this chapter is, to our knowledge, one of the first, which tries to address these issues in closed-loop systems. While our scenarios have strong constrains, the newly introduced information measures can be applied to a wide range of adaptive predictive controllers.

The chapter is organised in the following way. First off all we will describe the environment and the adaptive controller of our system and define several system measures. Then we will present results from single experiments to demonstrate the basic

behaviour of the system and provide an analytical solution of its temporal development. Afterwards we will show results for the different system measures showing a statistical analysis for different agents and different environments. Finally we will discuss the question of "optimal robots" and will conclude this chapter with a Discussion section relating our work to other approaches.

3.2 Experimental setup

Note, all spatial measures in the following are given in arbitrary "size units" (short "units"), time is measured in "steps".

3.2.1 Agent

The structure of the simulated agent used for these simulations is shown in Fig. 3.1 B. It is a Braitenberg Vehicle of diameter 40 units with two lateral wheels. It operates in a square arena of 400×400 units or a circular arena with diameter of 400 units, which can be empty ("simple world") or contain different numbers of obstacles ("complex worlds"). By default, the agent drives straight forward (dashed arrow) with speed $\nu = 1$ units per time step. It has two sensor-pairs, near-sensors and far-sensors, at the front; each sensor resembling a beetle's antenna, albeit here with ideal spring-like properties. Short near-sensors elicit the reflex signal x_0 and long far-sensors the predictive signal x_1. Triggering of a sensor will happen as soon as the agent gets close enough to an obstacle. Then sensor signal x will be elicited according to:

$$x(t) = \beta\, x(t-1) + (1-\beta)\frac{\lambda_t}{\Lambda}, \tag{3.1}$$

where λ_t is the part of the antenna bent by an obstacle at time point t and Λ is the length of the antenna. The constant $\beta = 0.6$ defines the decay rate of the first order low pass implemented by the feedback $x(t-1)$. We use a fixed reflex antenna length of $\Lambda_0 = 10$ units and different antenna lengths for the predictive sensor of $\Lambda_1 = 40, 50, 60, 70, 80, 100, 120, 150, 200$. In the following we will use the antenna ratio $\frac{\Lambda_1}{\Lambda_0}$ to specify different robots.

From this the agent computes its output z as:

$$b_{0,1} = sign(x^R_{0,1} - x^L_{0,1}),$$

$$z = \omega_0\, b_0\ \max\{|x^R_0|, |x^L_0|\} + \omega_1\, b_1\ \max\{|x^R_1|, |x^L_1|\}, \tag{3.2}$$

where $x^{L,R}_{0,1}$ are sensory inputs from the left and right side obtained by Eq. 3.1 above. This corresponds to a linear summation neuron with an added-on winner-takes-all mechanism so that the reaction z will follow the strongest of the left and right sensor signals in case both are triggered at the same time.

Signal z is then directly used to change the robot's driving angle α. For the remainder of this chapter it is important to remember that z directly corresponds to the *change of the turning angle* $d\alpha/dt$:

$$\frac{d\alpha}{dt} = g_\alpha z(t), \tag{3.3}$$

where g_α is the steering gain. From this the change of the robot position can be calculated for each time step and as a result of this setup the agent will avoid obstacles when moving through its arena.

3.2.2 Learning rule

For learning we use the ICO (input correlation) rule (Porr and Wörgötter, 2006), because of its intrinsic stability, given by (Fig. 3.1 C):

$$\frac{d\omega_1}{dt} = \mu x_1 \frac{dx_0}{dt} \tag{3.4}$$

Note, that the typical low pass filtering of the input signals for ICO learning (Porr and Wörgötter, 2006) is performed by the environment itself and by Eq. 3.1.

3.2.3 Closed-loop system

The general structure of the closed loop has been presented in Fig. 3.1 A and was discussed in the introduction so that only a few explanations need to be added here.

In general we denote the transfer function of the agent by H and those of the environment by P. In Fig. 3.1 A we have added the time variable t to all those components of which the temporal development is of interest in the context of this study: $x_0,\ x_1,\ z,\ \tau$ and ω_1. The other synaptic weight ω_0 is kept constant at 1.0.

3.2.4 Experimental procedure

We tested nine robots with different antenna ratio $\Lambda_1/\Lambda_0 = 4, 5, 6, 7, 8, 10, 12, 15, 20$ in four environments of different complexity. We used a circular environment with a diameter of 400 units where complexity was defined by the number of obstacles (3, 7, 14, or 21). We used square shaped obstacles of size 20×20 units that were placed at random positions in a circular manner at the perimeter of three imaginary circles with radii of 50, 120 and 190 points. This way we avoid deadlock situations and assure a free path along the whole circular arena. Several examples of the simplest and the most complex environments are shown in Fig. 3.2.

Two different types of experiments are being made. (1) Normal learning experiments where the robots actually learn while driving and (2) steady state experiments

(called weight freezing), where we keep ω_1 for some time at a preset value for measuring the currently queried parameter(s) in a steady state situation. Then ω_1 will be increased and parameter(s) will be measured again and so on until we are reaching the final weight ω_1^f at which the reflex is not triggered anymore.

We used the following procedure for this. We set the weight ω_1 to specific values $(0, \Delta\omega_1, 2\Delta\omega_1, \ldots \omega_1^f$, where $\Delta\omega_1 = 10^{-3})$ and, for each ω_1, let the robot run for $N = 20000$ time steps without learning.

Such a procedure is motivated by the fact that actual runtime is irrelevant (as explained above). Thus, by setting weights we can probe the robot's behaviour for a longer period in a steady state situation in order to get more data for the analysis.

3.3 System measures

In the following we will present different measures used to evaluate temporal development and success of learning, and to find the optimal robot for a specific environment.

3.3.1 Temporal development

To analyse the temporal development we measure how the temporal difference τ between inputs x_1 and x_0 changes on average during learning. As events in these systems are very noisy we need to adopt a method by which the time-difference between two subsequent x_1, x_0 events is reliably measured. For this we use the weight freezing procedure and keep $\omega_1 = const$ for N time steps. We define a window $c_w = 300$ steps. Then we use a threshold with value 0.02 on the x_0 signal and determine the time t_k where the signal x_0 reaches the threshold ($c_w \leq t_k < N - c_w,\ N = 20000$). Finally we place a window c_w around these t_k values and calculate the cross-correlation between x_1 and x_0 by:

$$C^k(t) = \sum_{T=-c_w}^{T=+c_w} x_1(t_k) \cdot x_0(t_k + T), \tag{3.5}$$

We determine the peak location of the cross-correlation as:

$$\tau_k = \underset{t}{\operatorname{argmax}}\, C^k(t). \tag{3.6}$$

Finally we calculate the mean value of the obtained different time differences τ_k for the whole frozen time section (N steps) according to:

$$\overline{\tau} = \frac{1}{M} \sum_{k=1}^{M} \tau_k, \tag{3.7}$$

where M is the number of found threshold crossings. After increasing ω_1, this procedure is repeated until ω_1^f.

3.3.2 Energy

We measure how much energy the robot uses for a given task during the learning process. In physics the total kinetic energy of an extended object is defined as the sum of the translational kinetic energy of the centre of mass and the rotational kinetic energy about the centre of mass:

$$E_k = \frac{1}{2}m\nu^2 + \frac{1}{2}I\omega^2, \tag{3.8}$$

where m is the mass (translational inertia), I is the moment of inertia (rotational inertia), ν and ω are the velocity and angular velocity respectively. As we use a constant basic speed ν and our all robots have the same size we can simplify the previous equation and define the mean output energy as:

$$\overline{E_z} = \frac{g_\alpha^2}{2N}\sum_{t=0}^{N-1} z^2(t). \tag{3.9}$$

We note that the change of the turning angle $\frac{d\alpha}{dt} = g_\alpha z(t)$ is directly to be understood as the angular velocity w.

3.3.3 Path Entropy

The following measure quantifies the complexity of the agent's trajectory during the learning process. The function z determines the state of the orientation of both wheels (particles) relative to each other as the relative speed of one particle against the other determines the turn angle and hence the orientation of the robot. If the robot only makes sharp turns then we would find for z ideally only two values: zero for "no turn" and one other (high) value for "sharp turn". In defining the path entropy H_p in an information theoretical way by *number of states taken divided by number of all possible states* this would yield a very low entropy as only 2 states out of many possible turns are taken. On the other hand the path entropy would reach its maximum value if all possible steering reactions will be elicited with equal probability.

Thus, in order to calculate the path entropy we need to get probabilities $p(z_i)$ of the output function z for each value z_i. To do that, first we calculate a cumulative distribution function of z by:

$$F_c(z) = \sum_{z_i \leq z} f(z_i), \tag{3.10}$$

where $z = 0, \Delta z, \ldots 1$ (we used $\Delta z = 2 \times 10^{-3}$). Here $f(z_i) = 1$ if $z_i \leq z$, and $f(z_i) = 0$ otherwise. From the cumulative distribution function we calculate a probability distribution function to be able to calculate the probability of the different values of z given by $p(z)$:

$$p(z) = \frac{\Delta F_c(z)}{\Delta z}. \tag{3.11}$$

Then we can define H_p in the usual way as:

$$H_p = -\sum_z p(z)\ log_2\ p(z). \tag{3.12}$$

3.3.4 Input/Output Ratio

We define the input/output ratio H_z which measures the relation between reflexive and predictive contribution to the final output, and shows how this relation changes during the learning process. At the beginning of learning only the reflexive output will be elicited which would lead to zero value. With learning ratio should grow and reach a maximum when reflexive and predictive parts contribute to the output evenly. After that ratio should go down back to zero since the reflex is being avoided and at the end of learning only predictive reactions will be elicited.

We define the absolute value of the neuronal output for the x_0 pathway:

$$|z_0| = \sum_{t=0}^{N-1} |z_0(t)|, \tag{3.13}$$

$$z_0(t) = x_0(t) \cdot w_0,$$

and for the x_1 pathway:

$$|z_1| = \sum_{t=0}^{N-1} |z_1(t)|, \tag{3.14}$$

$$z_1(t) = x_1(t) \cdot w_1(t),$$

where N is the length of the sequence (here $N = 20000$ time steps). The total absolute value of neuronal output can be defined as:

$$|z| = \sum_{t=0}^{N-1} |z(t)|, \tag{3.15}$$

$$z(t) = z_0(t) + z_1(t).$$

Finally, the input/output ratio can be calculated by the following equation:

$$H_z = -\left(\frac{|z_0|}{|z|} log_2 \frac{|z_0|}{|z|} + \frac{|z_1|}{|z|} log_2 \frac{|z_1|}{|z|}\right). \tag{3.16}$$

Note that this measure would be similar to an entropy measure if one would use the probabilities that an output z is generated by the reflex x_0 or predictor x_1 instead of the integrals $|z_{0/1}|$.

3.3.5 Speed of learning

To evaluate the speed of learning we assess weight development and not time, noting that elapsed time is irrelevant. For instance, if the robot drives around a long time without touching obstacles (no learning events) this would not influence the weight. Learning is driven by events (x_1 and x_0 pairs) which is directly reflected by weight growth and this we relate to the speed of learning. Hence we can determine the speed of learning of a specific agent by measuring at which weight the agent reaches the maximum input/output ratio value, where reflex and predictor contribute equally to the output. Thus, we define the learning speed S as being inversely proportional to this weight:

$$S = \left(\underset{\omega_1}{\operatorname{argmax}} H_z(\omega_1) \right)^{-1}, \tag{3.17}$$

with $\omega_1 = 0, \Delta\omega_1, \ldots \omega_1^f$, where ω_1^f denotes the final weight at which the reflex x_0 is not triggered anymore.

Note in a given environment one finds that learning events can occur more or less often depending on the sensitivity of the reflex. In this case - to compare architectures at the reflex level - one would indeed want to measure time as such. We are, however, in the current study not concerned with this.

3.3.6 Optimality

In order to find an optimal robot for a specific environment we used an averaged optimality measure O which is a product of the speed of learning S and the final path entropy $H_p(\omega_1^f)$:

$$O = S \cdot H_p(\omega_1^f). \tag{3.18}$$

Note that we normalised values of S and $H_p(\omega_1^f)$ between zero and one before calculating the product in Eq. 3.18. With this measure we can find the optimal robot in a given world, which learns the task quickly and also produces relatively complex driving trajectories.

3.4 Basic behaviour of the system

The basic behaviour of the obstacle avoidance agent is presented in Fig. 3.2 where we show simulation results for a circular environment with 3 and 21 obstacles. In panels A and B we show weight development and corresponding driving trajectories (see insets) for the case where the robot was actually learning (no weight freezing here). The resulting weight curves for both cases are similar and we observe relatively rapid weight growth at the beginning of the learning and then slow saturation till the reflex is avoided and weights finally stabilise. Corresponding trajectories are

colour-coded where the blue colour corresponds to reflex-driven behaviour and the red colour corresponds to predictor-driven behaviour. Values for the colour-coding were calculated by a contrast measure given in appendix A.4. From the driving trajectories we can see that at the beginning the robots make sharp turns because of the initially built in strong reflex reaction whereby, as a consequence, the robot explores more or less the whole environment. With learning the predictor takes over which at the end leads to wall following behaviour since learnt steering actions are much weaker but are elicited earlier compared to the initially strong and late reflex reactions. Simulation results for the case where we used the weight freezing procedure are shown in panel C. This way we can, for different weights, show longer trajectories to better assess the robots' behaviour. Here we plot selected trajectories for two different environments (3 and 21 obstacles) and for two different robots (antenna ratio 6 and 15). Trajectories for each case are presented in rows where the first trajectory corresponds to the reflexive driving behaviour ($\omega_1 = 0$), the second and the third trajectory correspond to a mixture of reflexive and predictive behaviour, and the last trajectory corresponds to the predictive driving trajectory when the reflex is finally fully avoided ($\omega_1 = \omega_1^f$). Here we obtained similar driving behaviour as in the examples presented above. In general we observed that late, strong, and abrupt reflex reactions turn into early, weak and smooth predictive reactions whereby, as a consequence, a bouncing driving behaviour turns into a wall following behaviour.

3.5 Characterising the temporal development

Fig. 3.3 shows the results from one obstacle avoidance experiment in our standard empty square arena. Panels A and B show the development of the reflex (x_0) and predictor (x_1) signals over time (top panels), where the bottom panels show magnifications for the beginning and the end of the learning. As expected, x_0 shrinks substantially during learning, because the reflex signal is better and better avoided. It would finally fully vanish as theory predicts, leading to the stabilisation of weights (Porr and Wörgötter, 2006), only here – to be able to show how small x_0 signals look like – we have stopped the learning process before this final equilibrium had been reached (see Fig. 3.2 A for a completed process).

The predictor signal in panel B also gets smaller which is due to the fact that at the beginning of learning the predictive antennas are bent all the way until the reflex antennas finally also hit the wall whereas after learning the reflex is avoided and the predictive antennas are not so strongly bent anymore. Panel C finally shows the development of the output signal z, which shrinks in amplitude but gets wider over time.

Of special interest is the development of τ during the learning process. Therefore we carried out an experiment where we analysed how the time difference τ depends on the synaptic weight ω_1 and the angle at which the robot hits the obstacle. For

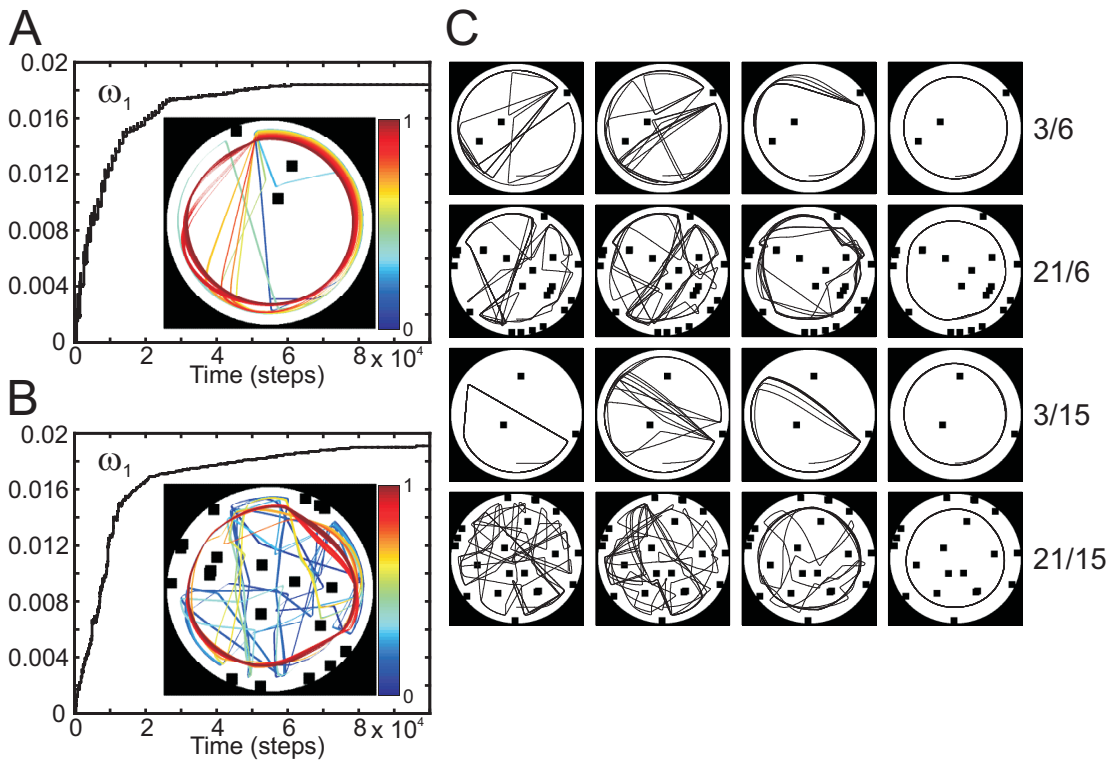

Figure 3.2: Driving trajectories from single experiments in a circular environment with obstacles. **A, B)** Weight development and corresponding driving trajectories obtained from individual experiments in an environment with 3 (panel A) and 21 obstacles (panel B). Trajectories are colour-coded where a zero value corresponds to reflex-driven behaviour and one corresponds to predictor-driven behaviour. The following parameters were used: antenna ratio $\Lambda_1/\Lambda_0 = 6$, steering gain $g_\alpha = 50$, learning rate was $\mu = 5 \times 10^{-3}$ for the case A, and $\mu = 10^{-3}$ for the case B. **C)** Driving trajectories obtained from individual experiments when using the weight freezing procedure in an environment with 3 (first and third row) and 21 obstacles (second and fourth row). For the two experiments shown in the first two rows we used a robot with antennae ratio $\Lambda_1/\Lambda_0 = 6$ whereas for the third and fourth experiment antenna ratio $\Lambda_1/\Lambda_0 = 15$ was used. The same steering gain $g_\alpha = 50$ was used for all four cases.

that we simulated our agent in a square and a circular environment without obstacles where we let the robot drive into a wall with different preset starting angles as shown in Fig. 3.4 (see insets). We varied the starting angle from 30 to 90 degrees in the square arena and from 40 to 90 degrees in the circular arena. Smaller angles were not possible here. In addition we also varied the weight ω_1 by setting it to a specific value

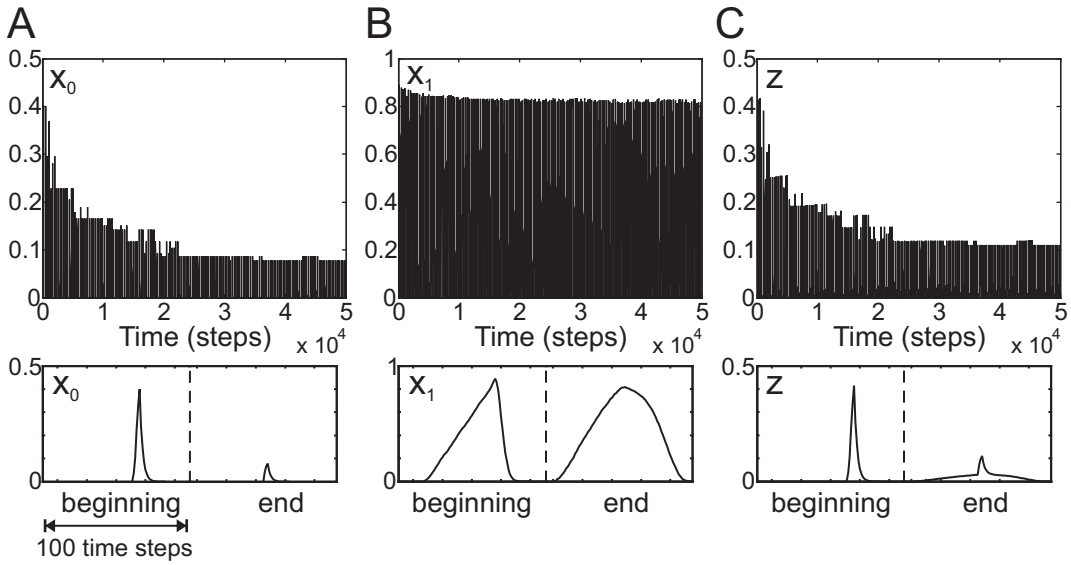

Figure 3.3: Results from one experiment in a square arena without obstacles. **A, B)** Inputs x_0 and x_1 respectively. **C)** Neuronal output z. Bottom panels show signal shapes at the beginning and at the end of the learning. The following parameters were used: antenna ratio $\Lambda_1/\Lambda_0 = 5$, steering gain $g_\alpha = 50$, learning rate $\mu = 0.06$.

$(0, \Delta\omega_1, \ldots$, where $\Delta\omega_1 = 10^{-3})$. Results for both environments are shown in Fig. 3.4 where we plot the time difference τ between inputs x_1 and x_0 against the synaptic weight ω_1. Here each curve shows time differences for one specific preset angle at which the agent drives towards the wall. The obtained results are very similar for both cases where we can see that the time difference increases for all given angles with increasing weights. We can also see that the increase for large angles is less pronounced than that for small angles. In general we observe that curves for small angles are shorter than those for larger angles, which is due to the fact that a less strong weight may suffice to avoid a wall when approaching under a small angle, but will not under a large angle. In a real learning situation this would lead to the fact that at the beginning all angles lead to learning, whereas at the end only large ones will. If we assume that there is no prior bias for any approach-angle (hence all angles will occur with equal probability without any learning), then this predicts that as soon as learning takes place an agent will *on average* experience τ values which follow (roughly) the average curve (grey) inside the "brushes" shown in Fig. 3.4.

To test this prediction, we analysed the development of τ statistically by testing nine different robots in four different environments. For the statistical evaluation we carried out 100 experiments for each specific case (in total 36 cases). All experiments were carried out by using the weight freezing procedure. Statistics are presented in

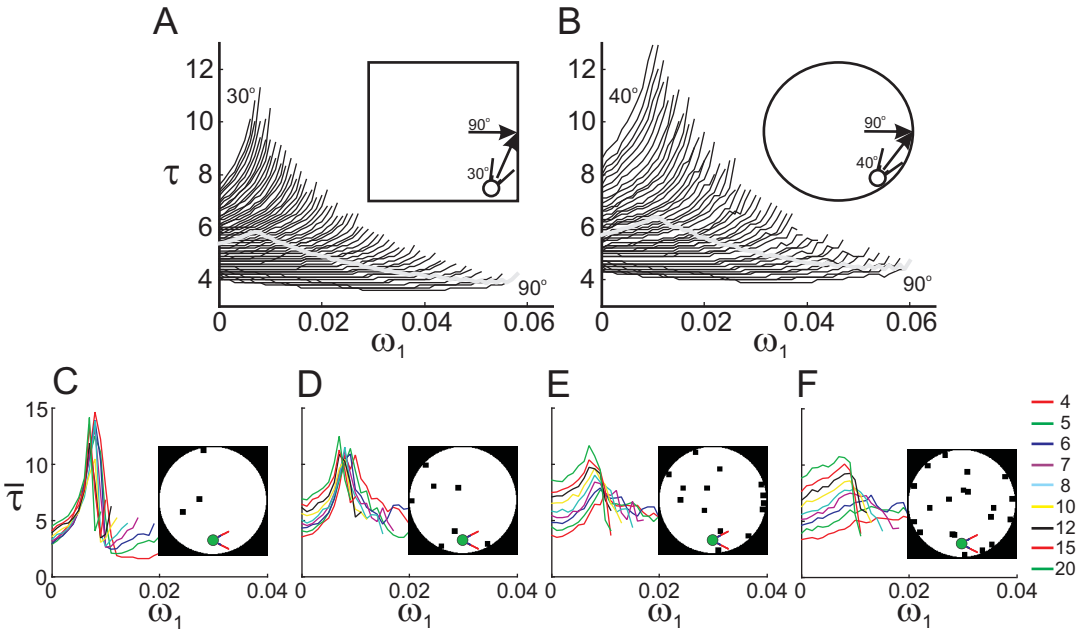

Figure 3.4: Time difference τ between far- and near-sensory inputs x_1 and x_0 for a wall avoidance task in a square (**A**) and a circular arena (**B**). τ is plotted against the weight ω_1 where each curve represents a certain angle with which the robot sets off to drive towards the wall of the specific arena as shown in the insets. Grey curve represents the average. The following parameters were used for all cases: antenna ratio $\Lambda_1/\Lambda_0 = 5$, steering gain $g_\alpha = 50$, weight change $\Delta\omega_1 = 10^{-3}$. **C-F**) Statistics for time difference τ between inputs x_0 and x_1 obtained from an obstacle avoidance task in a circular environment of different complexity with 3, 7, 14, and 21 obstacles (see insets for examples). Coloured curves in each panel show the averaged results plotted against the weight ω_1 obtained from 100 experiments where different colour represents results for the different robots defined by the antenna ratio Λ_1/Λ_0. The following parameters were used for all cases: steering gain $g_\alpha = 50$, weight change $\Delta\omega_1 = 10^{-3}$.

Fig. 3.4 C-F where we plot averaged results for all 100 experiments for each case. As discussed above we can see an increase of $\overline{\tau}$ at the beginning and then a decay later on. We can also observe that in general we get larger $\overline{\tau}$ values if we increase the antenna ratio which is obvious because longer antennas produce larger time differences between x_1 and x_0 events. In addition we observe that the time differences at the beginning of the development are smaller for simpler environments and are larger for more complex environments. The reason for this is that in a simple environment we get only those experiences where the robot drives into an obstacle placed close to

the wall with a sharp angle or into the opposite wall when it is repelled from the obstacle (for trajectories see Fig. 3.2 C cases 3/6 and 3/15), which leads to small, uniform values of $\overline{\tau}$ in panels C and D. In more complex environments the variety of experiences is much larger due to the more complex paths taken by the robot (see Fig. 3.2 C cases 21/6 and 21/15) and this leads to the larger and more dispersed $\overline{\tau}$ values in panels E and F.

3.6 Analytical closed-loop calculation of the temporal development

3.6.1 Definitions

The analysis of the different signals and their changes makes it now possible to provide an analytical approximation for the temporal weight development. To do so we need to simplify the observed signal structure. For the analytics, the reflexive signal x_0 consists of a linear rising and falling phase with identical slopes (see Fig. 3.5 A), where we allow the amplitude to diminish gradually towards zero. By contrast, for convenience, the predictive shape is described by a concave quadratic function (see Fig. 3.5 B). The definition of both, with their maximum being at $t = 0$, is as follows:

$$x_0(t) = \frac{A_0}{\sigma_0}\,(t+\sigma_0)\,\Theta(t+\sigma_0)\,\Theta(-t) + \frac{A_0}{\sigma_0}\,(\sigma_0 - t)\,\Theta(t)\,\Theta(\sigma_0 - t) \tag{3.19}$$

$$x_1(t) = A_1\left(1 - \frac{t^2}{\sigma_1^2}\right)\Theta(\sigma_1 + t)\,\Theta(\sigma_1 - t) \tag{3.20}$$

with Θ being the Heaviside step function.

We will see that this simple definition will lead to a very good approximation of the system's behaviour.

3.6.2 Weight change per learning experience

As the weight change per time step is defined by the ICO-learning rule, the weight change per experience k is the integral over a single x_1-x_0 experience using equation 3.4:

$$\omega'(k) := \frac{d\,\omega(k)}{d\,k} = \int_{-\sigma_1}^{\sigma_1} \mu\, x_1(t)\, \dot{x}_0(t-\tau)\, dt \tag{3.21}$$

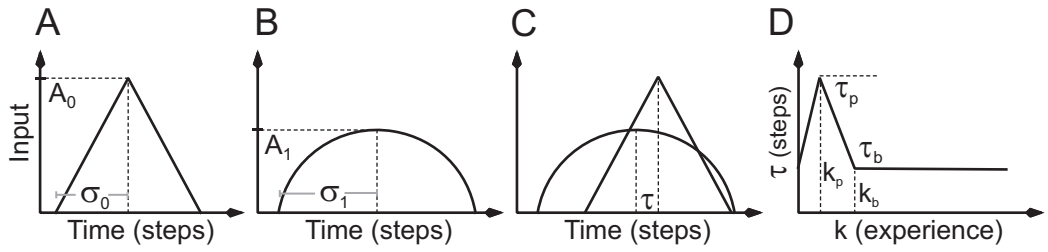

Figure 3.5: **A-C**) Structural simplifications of the input signals. **A**) Reflex signal. **B**) Predictive signal. **C**) The relation between both signals including the temporal difference τ. **D**) The development of τ-values over experience k.

where k is defined as experience and $\dot{x} = \frac{dx}{dt}$. Next we include the definition for $x_0(t)$ and $x_1(t)$ (i.e. equations 3.19 and 3.20) which allows us to integrate equation 3.21:

$$\begin{aligned}\omega'(k) =& \int_{\tau-\sigma_0}^{\tau} \mu\, A_1 \left(1 - \frac{(t-\sigma_1)^2}{\sigma_1^2}\right) \frac{A_0}{\sigma_0}\, dt - \int_{\tau}^{\tau+\sigma_0} \mu\, A_1 \left(1 - \frac{(t-\sigma_1)^2}{\sigma_1^2}\right) \frac{A_0}{\sigma_0}\, dt \\ =& \mu \frac{A_1\, A_0}{\sigma_0} \left(\left[t - \frac{1}{3} \frac{(t-\sigma_1)^3}{\sigma_1^2} \right]_{\tau-\sigma_0}^{\tau} - \left[t - \frac{1}{3} \frac{(t-\sigma_1)^3}{\sigma_1^2} \right]_{\tau}^{\tau+\sigma_0} \right) \\ =& \mu \frac{2\, A_0\, A_1\, \sigma_0}{\sigma_1^2}\, \tau. \end{aligned} \tag{3.22}$$

In order to avoid unnecessary complex case distinctions we used following constraints on τ: $|\tau| < \sigma_1 - \sigma_0$ given from the hindsight of the actual τ development we will encounter.

When looking at the data one finds that it is reasonable to keep most variables, especially A_1, σ_0 and σ_1 and some others (see Table 3.1), constant. Clearly the amplitude of the reflex A_0 should shrink as this leads to weight stabilisation. The parametrisation of A_0, thus, writes as $A_0 = a_0\,(1 - \frac{\omega}{\omega_f})$, were we use the final weight value ω_f as a control parameter for the shrinking of reflex amplitude A_0.

After including the parametrisation of A_0 into equation 3.22 we get:

$$\omega'(k) = \mu \frac{2\, a_0\, A_1\, \sigma_0}{\sigma_1^2}\, \tau \left(1 - \frac{\omega(k)}{\omega_f}\right) \tag{3.23}$$

Now the question arises whether a constant or a changing τ would be required for a good system description.

Analytical calculation of the weight development with constant τ

For a constant $\tau = \tau_b$ the solution of the first order differential equation Eq. 3.23 using the initial condition $\omega(0) = 0$ is

$$\begin{aligned}\omega(k) =& \omega_f \left(1 - \exp\left[-\mu \frac{2\, a_0\, A_1\, \sigma_0\, \tau_b}{\omega_f\, \sigma_1^2}\, k\right]\right) \\ =& \omega_f \left(1 - \exp\left[-\mu\, \lambda\, k\right]\right) \qquad (3.24)\end{aligned}$$

$$\text{with} \quad \lambda = \frac{2\, a_0\, A_1\, \sigma_0\, \tau_b}{\omega_f\, \sigma_1^2} \qquad (3.25)$$

Analytical calculation of the temporal development including the temporal dependence of τ on k

Different from above, here we start with equation 3.22 and add the parametrisation of A_0 and τ to this equation using:

$$\tau(k) = \begin{cases} \tau_b + (\tau_p - \tau_b)\, \frac{k}{k_p} & \text{if } 0 \le k \le k_p \\ \tau_p - \frac{\tau_p - \tau_b}{k_p - k_b}\, k_p + \frac{\tau_p - \tau_b}{k_p - k_b}\, k & \text{if } k_p < k \le k_b \\ \tau_b\, \frac{k}{k_f} & \text{if } k > k_b \end{cases} \qquad (3.26)$$

describing a linear increase in the beginning of learning which results in a τ-value of τ_p at experience k_p. It is followed by a linear decrease to the original τ-value of τ_b at experience k_b where it is kept fixed to the end (see Fig. 3.5 D). This gives us three second order differential equations, which we solve independently. Equations are structurally similar to Eq. 3.23 and their solutions are shown in appendix A.5.

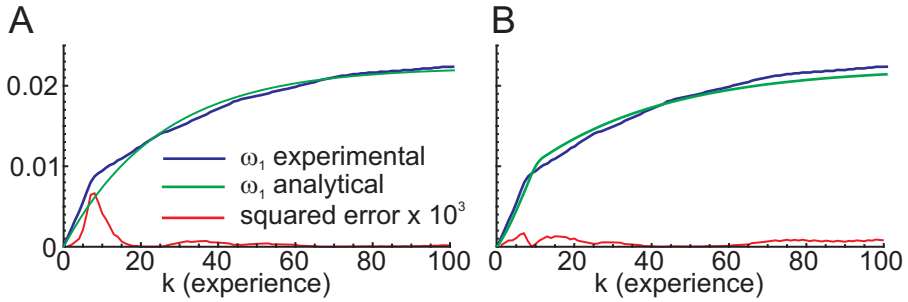

Figure 3.6: Comparison of experimental and analytical results of weight development when using constant τ (**A**) and variable τ (**B**). Parameters are given in parameters in Tab. 3.1. Additionally we show the squared error scaled by a factor of 1000.

Comparison of weight development with constant vs. changing τ

We can now extract the necessary parameters from the robot experiments and test to what degree the different situations (constant versus variable τ) describe the system correctly. Parameters are given in Table 3.1.

In Fig. 3.6 A and B we show the real weight change of the conducted experiment and the analytical solution for constant and variable τ. From the experimental data it can be seen that the weight ω_1 grows at two different rates. First, faster till experience $k = 10$ and then slower afterwards, which has been explained in sections 3.1 and 3.2, and is due to an initial increase in τ and then a decrease in τ to initial values. Consequently, the constant-τ solution (panel A) only captures the overall weight development, however, cannot reproduce the change in weight growth around experience $k = 10$. The fit for variable τ is substantially better (panel B) and the different weight growths are much better reproduced. The remaining error arises from the required simplifications used to arrive at the analytical solution.

Table 3.1: *Parameters extracted from an experiment. The first part states the parameters and their values needed for both, constant-τ and variable-τ, approximations, whereas the second and the third part give particular parameters used for the respective, constant-τ and variable-τ, cases. We additionally indicate the learning rate by μ_1 and μ_2, relating them to the equation used to fit the data.*

parameters	a_0	A_1	σ_0	σ_1	ω_f	τ_b	μ_1	τ_p	k_p	k_b	μ_2
values	0.6	0.85	43.75	5.75	0.0223	4	0.073	12	9	13	0.0523

3.7 Statistical evaluation of system measures

We analysed the development of the system measures during learning by testing nine different robots in four different environments. For statistical evaluation we carried out 100 experiments for each specific case (in total 36 cases). All experiments were performed by using the weight freezing procedure. Statistics are presented in Fig. 3.7 where we plot averaged results for all 100 experiments for each measure.

Results for the energy development are shown in panels A-D. We can see that energy is gradually decreasing as sharp reflexive steering reactions turn into smooth predictive reactions and less energy is needed for shallow turns compared to sharp turns. We also observe that the energy consumption at the end of the development is similar across all robots and all environments.

Development of the input/output ratio is presented in panels E-H. As expected, we observe that at the beginning of the development the ratio equals zero since only the reflex contributes to the neuronal output and then increases as the synaptic

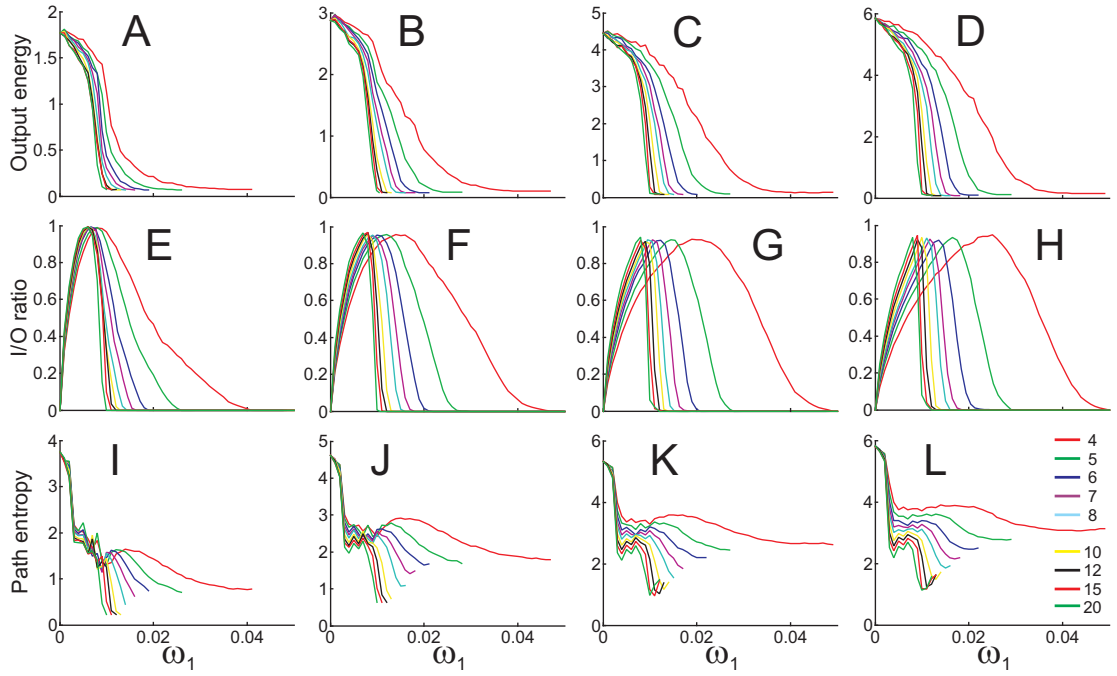

Figure 3.7: Statistics for different measures obtained from an obstacle avoidance task in a circular environment of different complexity with 3, 7, 14, and 21 obstacles (from left to right). **A-D**) - output energy E_z, **E-H**) - input/output ratio H_z, **I-L**) - path entropy H_p. Coloured curves in each panel show averaged results plotted against weight ω_1 for a specific measure obtained from 100 experiments where different colours represent results for a specific robot defined by the antenna ratio Λ_1/Λ_0 (see panel L). The following parameters were used for all cases: steering gain $g_\alpha = 50$, weight change $\Delta\omega_1 = 10^{-3}$.

weight of the predictive input grows. The ratio reaches a maximum when reflex and predictor contribute equally to the output. Thereafter the ratio decreases back to zero since with development we get less and less reflexive reactions and at the end of the development only predictive reactions are elicited. Different robots reach their maximum ratio at the different weights. Similarly, a different steepness of the decay after the maximum is found. Results suggest that in given environments, robots with longer antennas are quicker learners compared to robots with shorter antennas. We can conclude that the input/output ratio measure can be used to evaluate the success and speed of learning of a specific agent in a given environment.

Results for the path entropy are presented in panels I-L where in most cases we see a rapid decay at the beginning of the development followed by a small increase and a slow decay at the end. This tells us that the reflexive behaviour at the very

beginning of the development produces relatively complex paths whereas, when the predictor takes over, the driving trajectories become simpler, which leads to a decrease in path entropy. Usually there exists a transition phase during the development where the robot changes its driving trajectory in order to avoid obstacles producing more complex paths for some time and this is seen as a small increase in the path entropy curve. After that the path entropy slowly decreases since predictive reactions produce rather stereotypical and simple circular paths (see also Fig. 3.2 C). We can also observe that robots with shorter antennas produce more complex paths compared to robots with longer antennas.

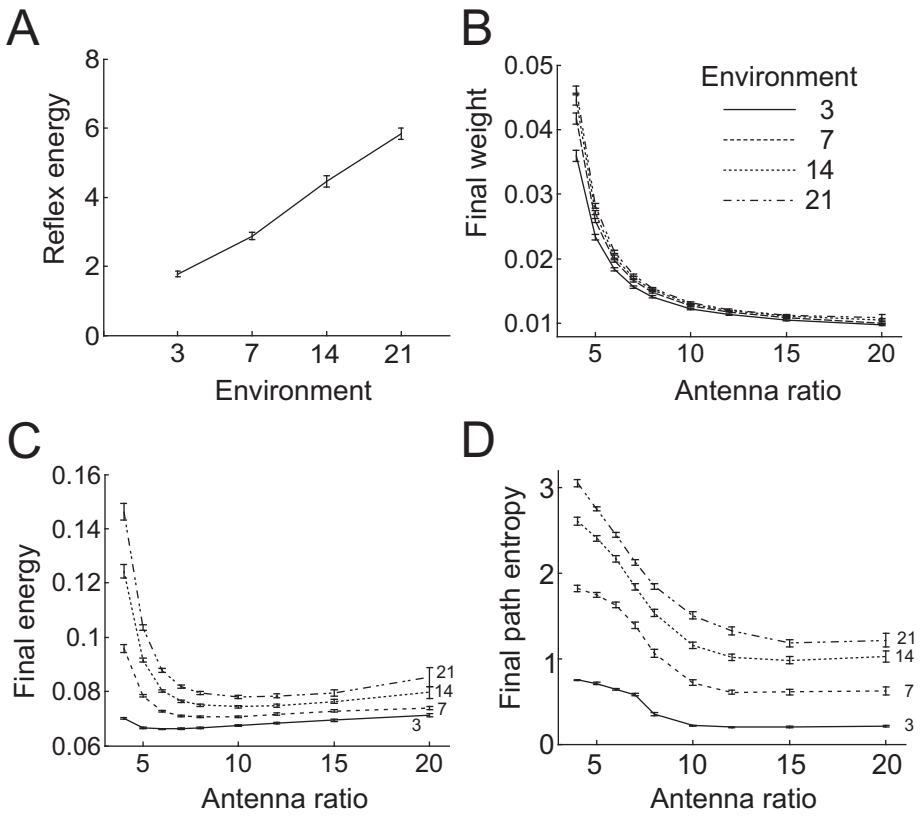

Figure 3.8: **A)** Average reflex energy plotted against environment complexity defined by the number of obstacles. **B)** final weight, **C)** final energy and **D)** final path entropy plotted against the antenna ratio Λ_1/Λ_0. The error-bars represent confidence intervals (95%) of the mean.

Summarised results for all robots and all environments are presented in Fig. 3.8. In panel A we compare energy consumption of the reflexive behaviour ($\omega_1 = 0$) where we see that energy consumption increases significantly with increase of environmental

complexity. This suggests that by measuring reflex energy we can judge complexity of the environment, and that learning is not necessary for such an evaluation. In panel B we compare the final weights (ω_1^f). Results demonstrate that there is no statistically significant difference between different environments except for the robots with short antennas (antenna ratio 4-8). Results for the final energy ($E_z(\omega_1^f)$) are compared in panel C. As expected, robots consume less energy in simple environments and more energy in more complex environments, although those differences are mush less pronounced compared to the pure reflex energy (panel A). We can also observe that robots with very long antennas are energetically slightly worse on average than robots with shorter antennas. In panel D we compare the final path entropies ($H_p(\omega_1^f)$). Here we obtained similar results to those of the final energy where we see that robots produce more complex paths as the environmental complexity increases. Results also demonstrate that in general robots with shorter antennas produce more complex paths compared to robots with long antennas.

3.8 On optimal robots

In the following we are concerned whether there exists an optimal robot for a given environment. We compare the robots' performance with respect to different measures (Fig. 3.9). In panel A we compare side by side energy consumption after learning, i.e. when the reflex x_0 is not triggered anymore. Here we can see that the minimal energy consumption shifts from robots with shorter antennas to robots with longer antennas as the environment's complexity increases, but differences (except for the shortest antennas) are small. As we can see in panel B the most complex paths are produced by shortest antennas ($\Lambda_1/\Lambda_0 = 4$) in all four environments. Concerning the speed of learning (for the speed measure see Eq. 3.17) we observe that robots with long antennas learn much quicker than robots with shorter antennas (panel C). Also the drop in performance, when getting into more complex environments, is less for long antennas as compared to short ones (see lines in panel C). We remind the reader that speed of learning is given by the equilibrium point (peak of input/output ratio, see Fig 3.7) where the reflex signal x_0 and the predictor signal x_1 contribute on average equally to the output.

In general one should think that "a good robot" would be one that produces – after learning – complex paths and learns those quickly. As we can see, however, there is a trade-off between these two constraints, the speed of learning and the path complexity for a given environment. As a consequence, by using the normalised product of these two quantities (see Eq. 3.18), panel D shows that there is an optimal robot existing for any given environment. For instance, the optimal robot for the simplest environment is the robot with antenna ratio five whereas in the most complex environment the robot with antenna ratio eight is the best. Based on the obtained results we can conclude that different robots adapt differently to a specific environment due to their

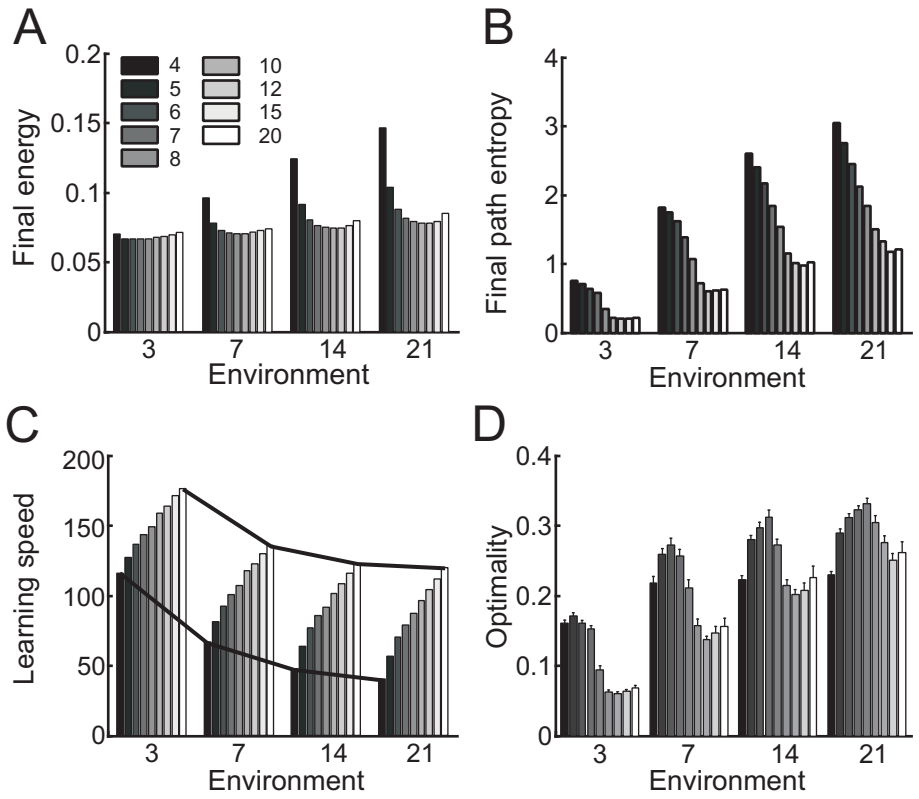

Figure 3.9: Comparison of different robots in specific environments of different complexity. Antenna ratios are given by the gray shading (see panel A). **A)** energy, **B)** path entropy, **C)** learning speed, and **D)** optimality. Average results are shown in each panel obtained from 100 experiments. In panel D error bars represent confidence intervals (95%) of mean.

different physical properties. Note, we did not consider using also the final energy for defining "optimality" because it does not alter the general picture. For most robots the final energy does not vary much (panel A) and, where its high (short antennas, complex worlds) and, thus, not optimal, including it in the measure would only emphasise the effect that robots with short antennas are best in simple worlds and vice versa (as stated above).

3.9 Applying system measures to receptive field analysis

3.9.1 Experimental setup

In the following section we will apply the above presented system measures for receptive field analysis in order to answer the question: What do receptive fields optimise? To do so, we performed simulations on a maze track as shown in Fig. 3.10 which is a slight modification of the maze track presented in Fig. 2.10 C. Here we removed the crossing point in order to make sure that the robot travels along the whole track and does not get stuck on one of the two sub-laps of the track. We also varied the direction angle of robot at its starting position. The value of α_0 was chosen randomly from a Gaussian distribution with mean $\overline{\mu}_{\alpha_0} = 0$ and variance $\sigma^2_{\alpha_0} = 4$. We used the same setup of the robot as presented in Fig. 2.17 and the same system parameters as described in section 2.9.1. In order to include more variance in the data here we place the robot on one of the four starting points as shown in Fig. 3.10 which was chosen randomly from a uniform distribution. Note that the robot was placed at a new starting point (chosen randomly) also after a loss of the track (deviation from the track by more than 20 units). Since we can not set weights of the receptive fields manually due to their unknown structure (no weight freezing possible), differently from the approach presented above, we let the robot learn continuously and evaluated system measures after learning. Learning in this case was treated as finished if there was no reflex triggered during a driving period of 4200 time steps (the robot travels the whole track in $\approx$ 4100 time steps). Due to the reason stated above we excluded speed of learning from our analysis and changed our optimality measure as follows. We used an optimality measure O_{RF} which is a product of the path entropy H_p, the inverse deviation from the track Ψ^{-1} (see Eq. A.3) and the inverse energy E_z^{-1}:

$$O_{RF} = H_p \cdot \Psi^{-1} \cdot E_z^{-1}. \tag{3.27}$$

Note that we normalised values of H_p, Ψ and E_z between zero and one before calculating the product in Eq. 3.27. With this measure we can find the optimal robot which after learning is able to produce different steering actions, performs the task accurately and does not consume a lot of energy.

3.9.2 Statistical evaluation

First of all we looked at the robot's performance with respect to the size (dimension) of the receptive field. We used four robots with RF size of 5×5, 10×10, 15×15 and 20×20 units. In order to compare robots with different RF size we had to place RF fields of all robots closer to the reflex (distance between reflex and RF position was $d = 5$ units), since for small RF size (5×5 units) larger distances d were not possible

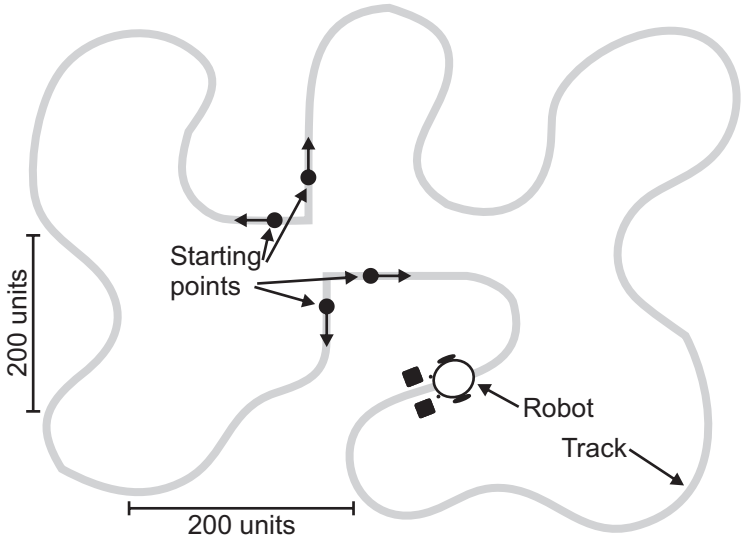

Figure 3.10: Experimental setup of the simulation of RF development on the maze track. Four different starting points (arrows show the direction of driving) were used in the simulations where the starting point was selected randomly from a uniform distribution. The width of the track was 1 unit.

due to poor correlations between RF inputs and the reflex. We also tuned the learning rate for each robot in order to achieve the same learning speed, i.e. number of required learning experiences (see Fig. 3.13 A). Examples of receptive fields of different size are shown in Fig. 3.11 where we show RFs from ten experiments for each case where we can observe that smaller RFs are noisier and less structured compared to larger RFs.

Single examples of motor outputs and corresponding driving trajectories when using different RF size (5×5, 15×15 and 20×20) are shown in Fig. 3.12. Here we can see that the motor output generated by small RFs (panel A) is much larger in amplitude and narrower as compared to motor outputs produced by bigger RFs (panel B and C) which generate weaker and wider responses. As a consequence small RF leads to over-steering which on shallow turns produces bouncing driving behaviour (see panel A), whereas bigger RFs lead to smoother and more accurate driving behaviour. This can be explained by the fact that smaller RFs are less structured as compared to bigger RFs. In general, we can observe that the driving behaviour of the robot with small RFs (5×5 units) is similar to the driving behaviour of the robot with the simple setup (see Fig. 2.10).

The statistical evaluation of different system measures from 100 experiments is presented in Fig. 3.13 B-E. Here we can see that the robot with the smallest RF

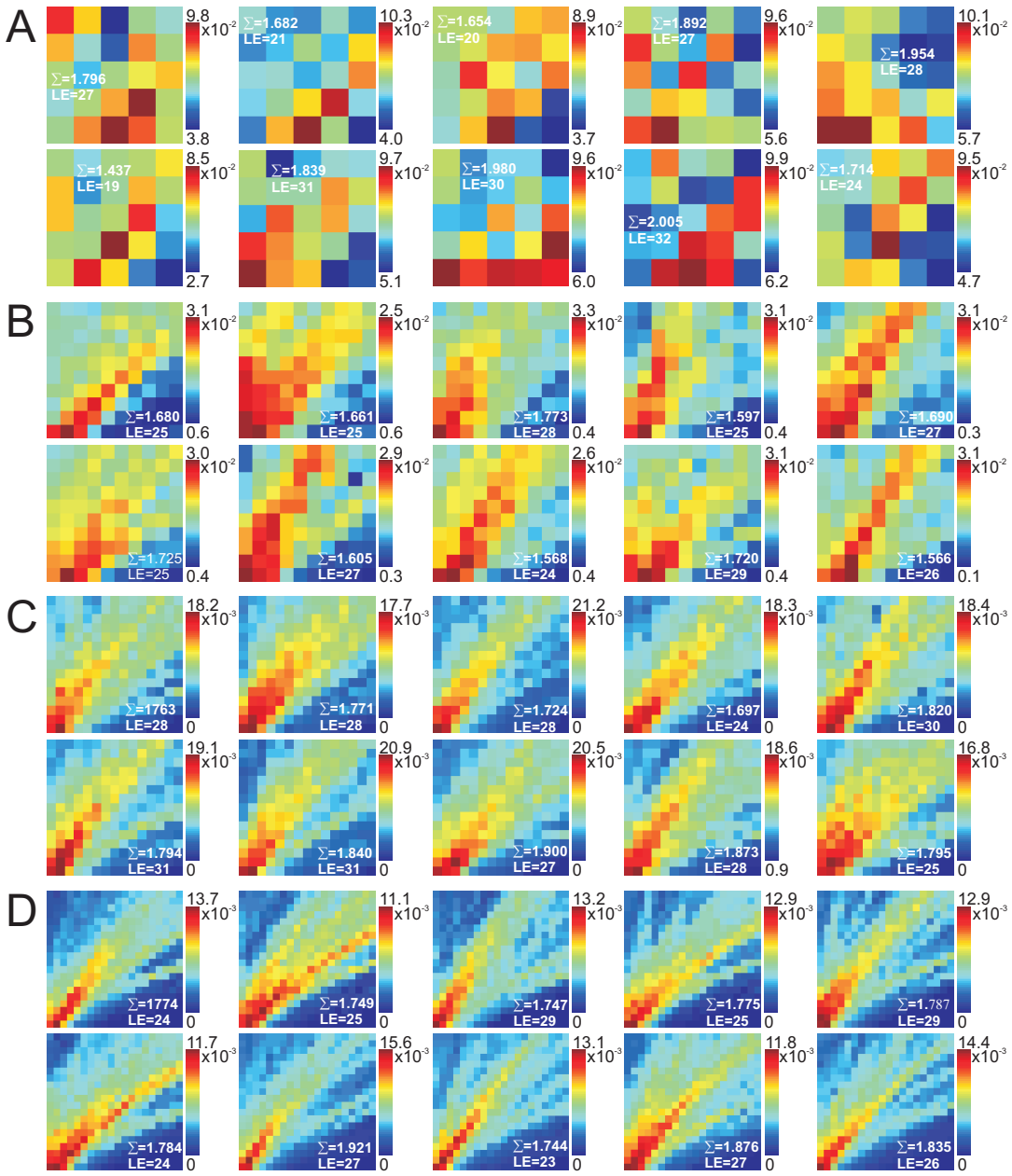

Figure 3.11: Examples of receptive fields obtained on the maze track (see Fig. 3.10) when using different RF size. **A)** 5×5, **B)** 10×10, **C)** 15×15, and **D)** 20×20 units. Values in the receptive field correspond to the total sum of all weights ($\sum$) and the number of required learning experiences (LE), respectively.

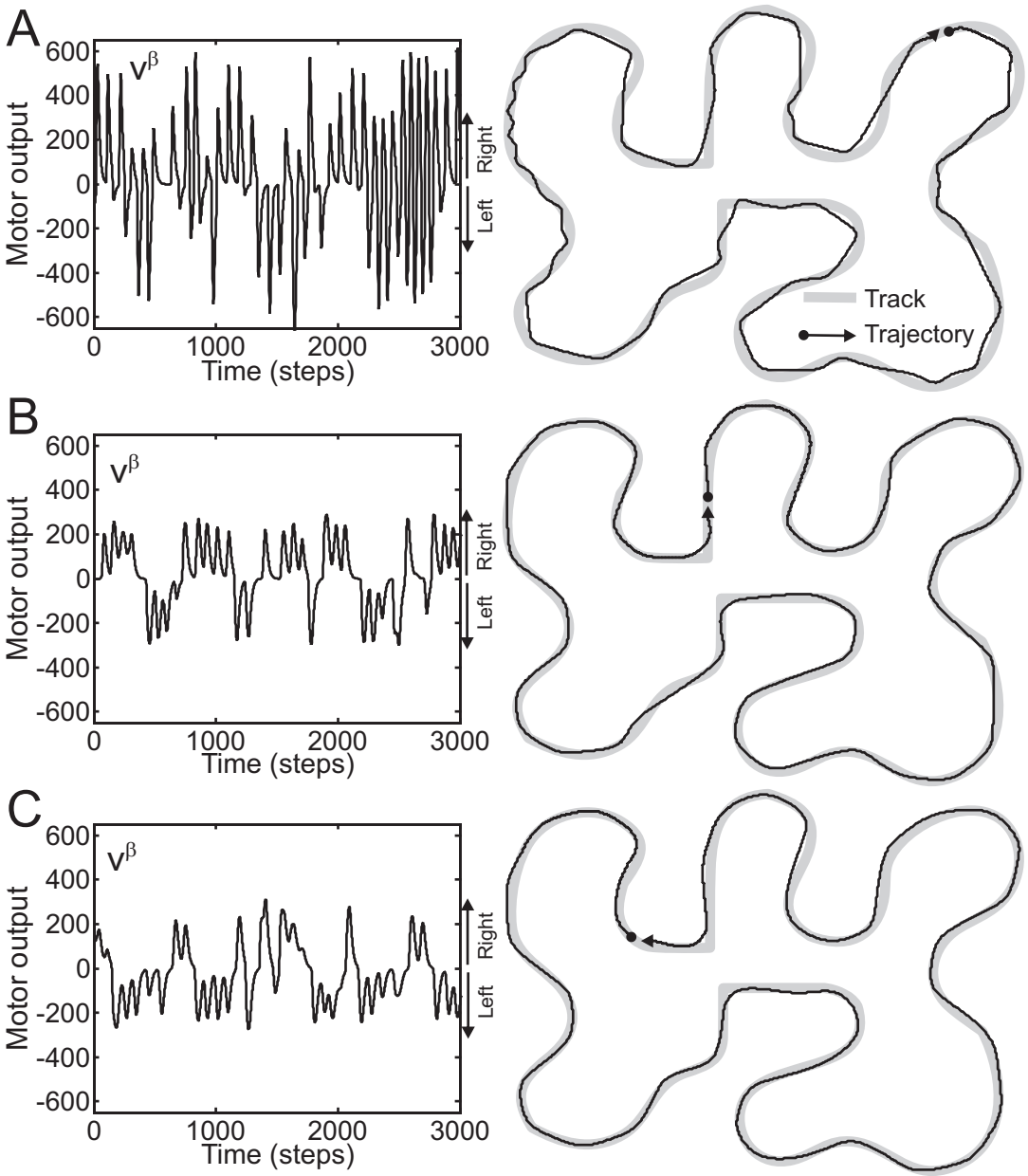

Figure 3.12: Examples of motor outputs v^β (left) and corresponding driving trajectories (right) of the robot obtained on the maze track (see Fig. 3.10) when using different RF sizes. **A)** RF size was 5×5, **B)** 15×15, and **C)** 20×20 units. Note that here motor outputs correspond to the $\approx 3/4$ of the driving trajectory (starting from the black dot).

deviates from the track significantly and more than the robots with larger RFs. This is due to the fact that smaller RFs are less structured and are not capable of producing as many different steering actions as larger RFs, which - as a consequence - leads to over-steering and relatively inaccurate driving behaviour (see Fig 3.12 A). This can also be seen from the path entropy H_p (see Fig. 3.13 D) where we can observe that H_p decreases if we reduce the resolution of the receptive field. The robots with RF size of 10×10 and 15×15 units were driving with the best accuracy, however, the robot with the largest RF was deviating from the track significantly more. This is due to the fact that the robot with very large RF (20×20 units) has it's inputs relatively far away from the reflex and starts to react earlier but with weaker steering reactions compared to the robots with smaller RFs, which leads to the under-steering in most of the cases (see Fig. 3.12 C), however, without triggering the reflex. We can also observe that the robot with the largest RF consumes least energy whereas the robot with the smallest RF is energetically mostly inefficient. This is because the robot with small RF tends to over-steer and the robot with large RF is reacting earlier with weaker steering reactions which as a consequence allows following the track without triggering the reflex with less energy. Concerning optimal robots we obtain that the best robot with respect to the driving behaviour and the energy consumption is the one with RF size of 15×15 units whereas the robot with the smallest RF (5×5 units) shows the worst performance.

We also checked the influence of the learning rate μ on the performance of the robot. Here we used default system parameters: size of the RF was 15×15 units and the distance between reflex and RF position was $d = 8$. Results from 100 experiments are shown in Fig. 3.13 F-H. As expected, the number of required learning experiences decreases if we increase the learning rate (see panel A). We also observe an increase in the deviation from the track and an increase in the energy with an increase of the learning rate. This can be explained by the fact that a smaller learning rate (slower learning process) leads to noise reduction in the RF structure which as a consequence leads to more accurate and less energy demanding driving behaviour.

Finally, we compared the performance of learnt receptive fields (heterogeneous RF) against that obtained with homogeneous RF, random RF and different transformations of the learnt RF. Here we wanted to check whether the structure of the RF plays an important role in agents behaviour. We hypothesised that the learnt RF (heterogeneous RF) will give better behavioural performance compared to the performance of an homogeneous RF or transformed RFs. To test this hypothesis we used the following procedure. First of all we let the robot learn a receptive field using our standard learning procedure. Afterwards we transformed the learnt RF and tested the robot's driving performance (learning rate μ was then set to zero) on the same path. We used the following RF transformations (shown in Fig. 3.14 A): vertical (VF), horizontal (HF) and diagonal (DF) flip of weights, random assignment of weights (R) by shuffling weights of the learnt RF randomly, and homogeneous RF, where all weights are the same and are equal to the average value of the learnt RF.

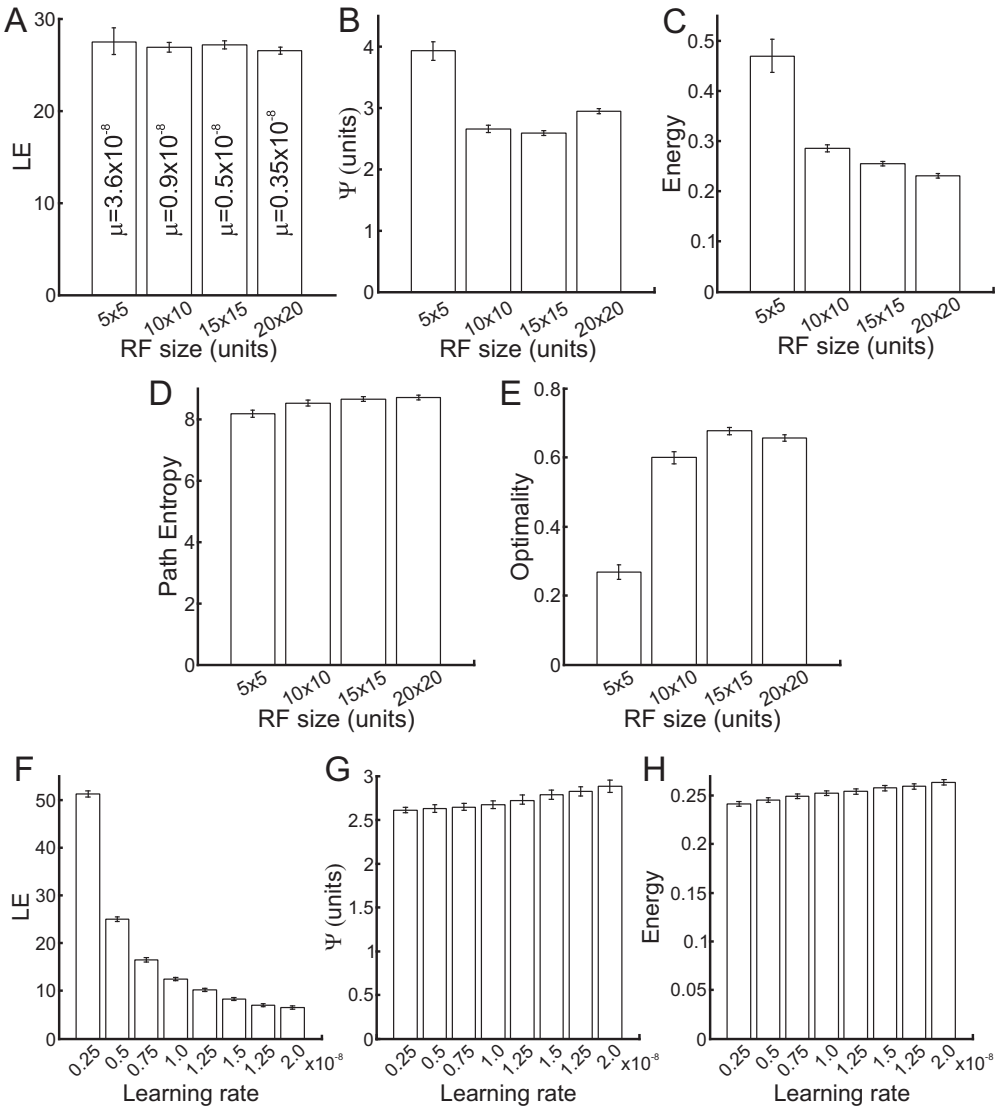

Figure 3.13: Average results from RF simulations on the maze track (see Fig. 3.10) obtained from 100 experiments. **A-E)** Different measures plotted versus RF size. **A)** Number of required learning experiences (LE), **B)** robot's deviation from the track Ψ after learning, **C)** final output energy, **D)** final path entropy, and **E)** optimality as given in Eq. 3.27. **F-H)** Different measures plotted versus learning rate μ. Results are obtained with an RF size of 15×15 units. **F)** Number of required learning experiences (LE), **G)** robot's deviation from the track Ψ after learning, and **H)** final output energy. Error bars represent confidence intervals (95%) of mean.

Note that the vertical and the horizontal flip result in the change of the position and the orientation of the RF pattern whereas the diagonal flip changes only the position of the pattern but leaves the orientation the same. Here we used a relatively low learning rate of $\mu = 0.25 \times 10^{-8}$ (which corresponds to ≈ 50 LEs on average) in order to develop RF structure. The distance between reflex and RF position was $d = 8$. For comparison we also tested the robot's behaviour when driven by the reflex alone (control case). For the control case (C) we performed 100 experiments (4200 time steps each) where we let the robot drive without learning (reflexive behaviour) and computed the system measures. We replaced the robot to one of the four starting points (chosen randomly) in case it lost the track.

Results from experiments with different RF transformations are presented in Fig. 3.14 B-F. First of all we can observe that the heterogeneous receptive fields (learnt RF) increase the robot's driving accuracy (panel B), reduce the energy[1] (panel C), and increase the variance of the motor output (path entropy, see panel D) as compared to the purely reflex-driven behaviour. Secondly, the robot with heterogeneous RF also performs better with respect to the driving behaviour, as shown by the larger path entropy, than the robots with transformed RFs (see panels B and D). We obtain that the robots with transformed RFs use significantly less energy as with the learnt RF. This is due to the fact that transformed receptive fields are not capable of producing appropriate driving behaviour which leads to the triggering of the reflex (see panel E where we plot number of reflexes triggered during test driving). Note that here we use a relatively strong reflex which to be able to bring the robot back to the track. In case the weaker initial reflex was used the robot would not be able to stay on the track and would lead to the loss of the track, which in some cases might be very costly to the agent. Although the robot with learnt RF uses more energy (which is needed to avoid the reflex) than the robots with transformed RFs, performance of heterogeneous receptive field is the best with respect to the driving behaviour and energy consumption (see panel F).

In general, we observed that heterogeneous RFs optimise the agents' behaviour which supports the importance of the receptive field structure and that there are specific system parameters (such as RF size) which lead to the best performance of the agent for the given task.

3.10 Discussion

In this chapter we have started to address the difficult question how to quantify continuous learning processes in behaving systems that change by differential hebbian plasticity. The central problem lies here in the closed loop situation, which leads - even

[1]Note that for VF, HF, DF, R and HM cases final output energy takes both predictive and reflexive energy into account.

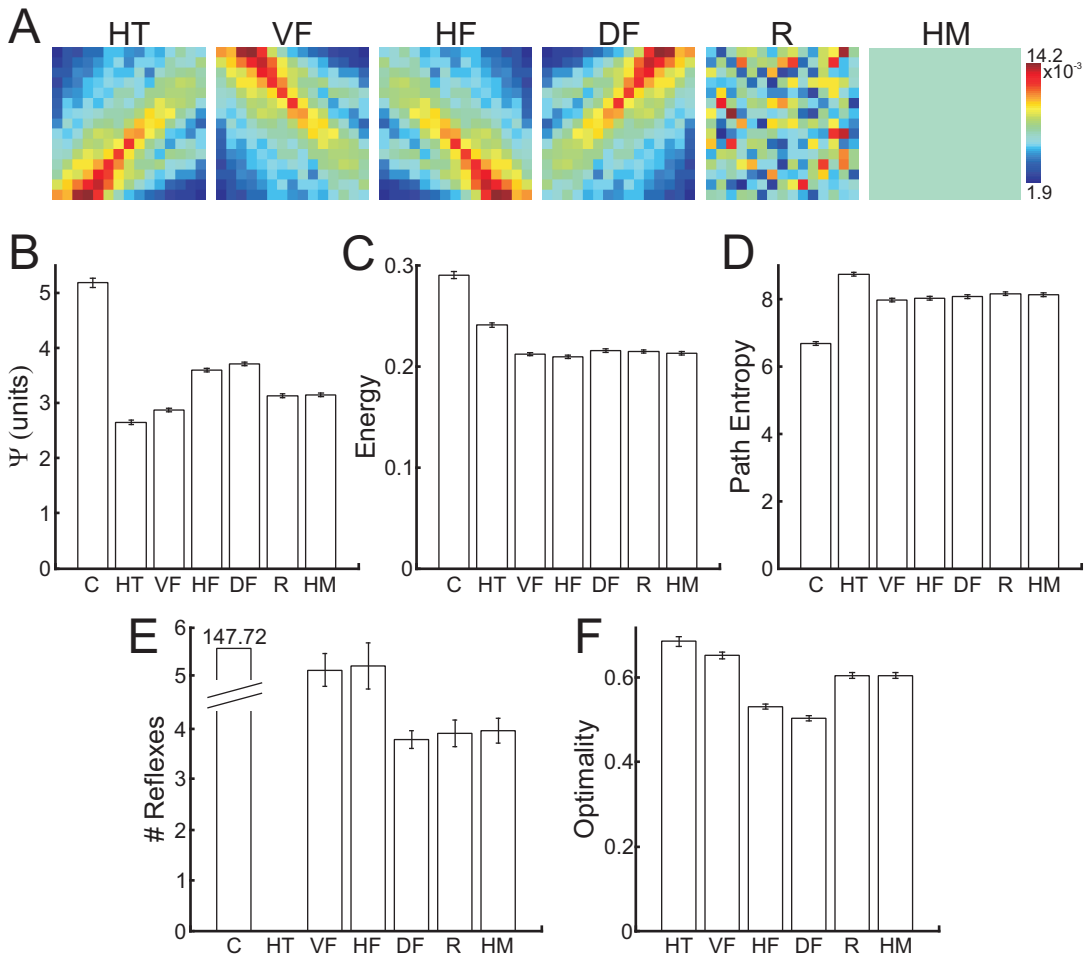

Figure 3.14: **A)** Example of learnt receptive field and it's transformations. HT - heterogeneous receptive field (learnt RF), VF - vertical flip of learnt RF, HF - horizontal flip, DF - diagonal flip, R - randomly shuffled weights of learnt RF, and HM - homogeneous RF where all weights are the same and are equal to the average value of learnt RF. **B-F)** Average results from RF simulation on the maze track (see Fig. 3.10) obtained from 100 experiments. Different measures plotted versus different RF transformations. Note that C is the control case (reflex-driven behaviour). **B)** robot's deviation from the track Ψ after learning, **C)** final output, **D)** final path entropy, **E)** number of reflexes triggered during the test driving, and **F)** optimality as given in Eq. 3.27. Error bars represent confidence intervals (95%) of mean.

in very simple linear cases - to an intricate interplay between behaviour and plasticity. Signal shapes and timings change in a difficult way influencing the learning. As a

consequence, it is not easy to find an appropriate description and the right measures for capturing such non-stationary situations. Fig. 3.1 A shows the structure of our closed loops system and this diagram has been used in earlier studies for convergence analyses (Porr and Wörgötter, 2003a,b, 2006; Kulvicius et al., 2007). From this diagram it becomes clear that τ, z as well as $x_{0,1}$ are the relevant variables in our system. While learning is defined by the relation between inputs $x_{0,1}$ and, hence, τ; behaviour is defined by output z.

3.10.1 Aspects of model identification

In the first part of this chapter we have concentrated on the inputs and we could show how τ develops over time for different robots and in different worlds. The peaked characteristics of the development of τ during learning (Fig. 3.7) is a nice example of the mutual interaction between behaviour and plasticity. Touching a wall with a shallow angle just does not occur anymore after some learning and the system finds itself in the domain of large approach angles where τ shrinks again (Fig. 3.4), contrary to our naive first intuition, which had argued for a continuous growth of τ. This also leads to a biphasic weight development and it was possible to use the measured τ-characteristics, together with some assumptions on the amplitude change of x_0 and x_1, to quite accurately calculate such a weight development in an analytical way.

In the introduction we had asked (question 1) to what degree the temporal development of such systems could be described by knowing just the initial conditions of robot and world. The first part of this paper showed that one needs a bit more than just the initial conditions. Only together with some information on the general structure of the development of x and τ, we can reproduce the biphasic characteristics of the weight development by which the essence of such systems is captured. Essentially this part of the study was, thus, concerned with aspects of model identification asking by which parametrisation the behaviour of a simulated robot would be captured.

Several methods are known from the literature to address the model identification issue in a broader context. For example one can use a [Non-linear] Auto-Regressive Moving Average approach with or without exogeneous inputs ([N]ARMA[X], Box et al., 1994) to arrive at a general model of behaving robot systems (Iglesias et al., 2008; Kyriacou et al., 2008), but these models contain many parameters for fitting and parameters do not have any direct physical meaning. Our attempts stop short of a complete model identification approach, which does not seem to be required for our system. Instead, here we could use a rather limited model with quite a reductionist set of equations (see section 3.6), which was to some degree unexpected given the complexity of the closed loop behaviour of our robots (Fig. 3.2).

3.10.2 Comparison to other work on information flow in closed-loop systems

In the second part of this paper, we have started to quantify the behaviour of our little beetles by considering their output z. We have defined measures for energy, input/output ratio and entropy focusing on the question whether there is an optimal robot for a given environment (question 2 in the introduction). Interestingly one finds in the first place that learning acts "equalising". Robots with different initial (reflex) energy (Fig. 3.8 A) become very similar after learning (Fig 3.8 B, note the different scales in panel A and B). This finding can be understood from some older studies on closed-loop differential hebbian (ISO, ICO) learning. Fig. 3.1 A shows that these systems will learn avoiding the reflex and that learning will stop once this goal has "just" been reached leading to an asymptotic equilibrium situation (Porr and Wörgötter, 2003b). Furthermore the systems investigated here are linear, hence all of them will in the end essentially require the same total effort for performing the avoidance reaction. These two facts explain why their energy is very similar in the end. The fact that robots are different, however, does surface when looking at the paths they choose after learning. Robots with long predictive antennas can never make sharp turns anymore and their paths are dominated by performing the same shallow turns again and again leading to little path variability and hence to a small final path entropy (Fig. 3.8 D). On the other hand, these same long-antenna robots learn their task much faster than their short-antenna fellows: for the former, the equilibrium point between reflex and predictor (peak in the input/output ratio) is reached faster than for the latter (Fig. 3.7 E-H).

This leads to a trade-off and by using the normalised product of learning speed times path entropy we found that for different environments different robots are optimal (Fig. 3.9 D). Clearly, this type of optimality is to some degree just in the eyes of the beholder and one might choose to weigh the two aspects (learning speed and path complexity) differently by which other robots would be valued more than those currently called 'optimal'. Nonetheless, also with a different weighing one will observe that some robots would be better than others in the different worlds.

In general this part of the study relates to work focusing on information flow in closed-loop systems. There have been a few contributions to this topic. Tishby et al. (1999) introduced an Information-Bottleneck (IB) framework that finds concise representations for a system's input that are as relevant as possible for it's output, i.e. a concise description that preserves the relevant essence of the data. The relevant information in one signal with respect to the other is defined as the mutual information that the one signal provides about the other. Although, the Information-Bottleneck framework was successfully applied in various applications, like data clustering (Slonim and Tishby, 2000; Slonim et al., 2001), feature selection (Slonim and

Tishby, 2001), POMDPs[2] (Poupart and Boutilier, 2002), it conceptually differs from our study, since we are interested in the dynamics of sensory-motor systems during the learning process.

In the work of Klyubin et al. (2004, 2005, 2007, 2008) the authors used a Bayesian network to model perception-action loops. In their approach a perception-action loop is interpreted in terms of a communication channel-like model. They show that maximisation of information flow can evolve into a meaningful sensorimotor structure (Klyubin et al., 2004, 2007). In Klyubin et al. (2005, 2008) the authors present a universal agent-centric measure, called "empowerment", which is defined as the information-theoretic capacity of an agent's actuation channel (the maximum mutual information for the channel over all possible distributions of the transmitted signal). The empowerment is zero when the agent has no control over its sensory input, and it is higher when the agent can control what it is sensing. In these studies it could be demonstrated that maximisation of empowerment can be used for control tasks (such as pole balancing) as well as for an evolution of the sensorimotor system or even to construct contexts which can assign semantic "meaning" to the robot's actions (Klyubin et al., 2005, 2008). Similar to the work of Klyubin et al. (2004, 2005, 2007, 2008) in the study of Prokopenko et al. (2006) the authors used two measures called generalised *correlation entropy* and generalised *excess entropy* to alter the locomotion of a simulated modular robotic system (snake-like robot) by an evolution process. The mentioned studies differ from our approach, since in these works information measures had been used to drive a sensorimotor adaptation on a relatively large time scales (simulating evolution by using genetic algorithms) whereas in our approach we use information measures to investigate the behaviour of closed-loop system during on-line learning on relatively short time scales.

Lungarella et al. (2005) have shown that coordinated and coupled sensorimotor activity decreases the entropy and increases the mutual information within specific regions of the sensory space. In contrast to our study they analysed information flow only on the sensory inputs whereas we consider inputs as well as outputs (input/output ratio, path entropy, energy). Also, different from our attempt, these authors analysed the system in a reflex-based closed-loop scenario where no learning had been applied. Ay et al. (2008) and Der et al. (2008) used a predictive information measure (PI, mutual information between past and future sensor values) to evaluate behavioural complexity of agents and to use PI as an objective function for the agents' adaptation, however, similar to Lungarella et al. (2005), only looking at the input space.

An earlier study of Lungarella and Sporns (2006) has demonstrated that learning can affect information flow (transfer entropy) of the sensorimotor network of a behaving agent. In this study transfer entropy was used to analyze the causal structure of the loop, i.e. causal effects of sensory inputs on motor states and vice versa, whereas

[2]POMDP - Partially observable Markov decision process

in our study we use system measures to analyze the system dynamics during learning with respect to the speed of learning and behavioral performance of an agent. Also differently from our approach Lungarella and Sporns (2006) used incremental reward based learning, which belongs to a different class of learning algorithms.

Our approach more closely relates to the study of Porr et al. (2006). They define the information value (called predictive information) only by the weights of the ISO learning rule (Porr and Wörgötter, 2003b), where, different from our approach (see Eq. 3.16), sensory inputs and outputs are not included in this measure. In Porr et al. (2006) weights reflect the predictive power of their corresponding inputs: the larger the weights the higher the predictive information. Essentially this measure shows which inputs are more predictive in relation to the signal at x_0, whereas in our approach the measures of input/output ratio, path entropy and energy reflect the general behaviour of the system, for example the contribution of reflex and predictor to the system's output.

Our measures, similarly to Porr et al. (2006), are developed within the framework of predictive correlation based learning (specifically using the ICO-rule here). These measures can be also used for other learning rules as long as the reflex and the predictive inputs can be identified. The previously discussed empowerment measure (Klyubin et al., 2005, 2008) is independent of the specific learning rule and can treat the system as a black box. As mentioned before empowerment is defined as channel capacity, which is the maximum mutual information over all possible distributions of the transmitted signal. This quantity is difficult to calculate and may require using a "detachable" world model that allows exact repetitions of certain behaviours in a particular situation (Klyubin et al., 2008). This means that it is not straightforward to use empowerment for analysing on-line behavioural systems. Note that all our output-signal based measures used by us, for example our path entropy measure, which measures the variability of different actions, can also be applied independently of the learning rule and the actual behavioural pattern and could, thus, be used also in other systems quantifying their (possibly entirely different) behaviour and its variability.

4

Place Cell Model and Goal Navigation

4.1 Introduction

In the previous chapters 2 and 3 we showed how receptive fields can be developed from visual stimuli by using temporal sequence learning and used for a driving task in a closed loop scenario. As in the previous chapters we will also here stay in the closed-loop context, however, we will develop another kind of receptive fields, called place fields, and use them for gaol directed navigation. In contrast to chapter 2, here we will use multi-modal sensory cues (visual and olfactory) to develop place fields instead of uni-modal cues (visual cues alone). In the current chapter, different from chapters 2 and 3, we will apply different learning mechanisms: unsupervised learning (vector quantization) for place field development and reinforcement learning (Q-learning) for path learning.

Place cells are principal neurons in hippocampus which respond maximally when the animal is in a specific location in an environment. They were discovered in rat hippocampus by O'Keefe&Dostrovsky in 1971 (O'Keefe and Dostrovsky, 1971; O'Keefe and Nadel, 1978) and investigated in numerous studies (for reviews see Eichenbaum et al., 1999; Hölscher, 2003). Place fields form from environmental cues and play an important role in spatial navigation. Cells having similar properties to rat place cells had also been found in humans using extracellular recordings from epileptic children (Ekstrom et al., 2003). Thus, the formation of place fields, and their influence on navigation remains an important experimental and theoretical question. In particular, little is known on how different sensory cues contribute to place field formation and spatial navigation. Thus, the goal of the first part of this chapter is to investigate how place fields are formed under visual as well as olfactory influences extending the uni-modal view of place field representation.

In the second part of this chapter, we address the question of how place fields can be used in navigation, and compare this to olfactory based navigation based on self-laid scent marks. Here we would like to stress that we are dealing with a closed loop system (see Fig. 4.1) where place field development and path learning progresses simultaneously, i.e. the trajectory of the rat influences the development of place fields, whereas, at the same time, place fields influence the weight development of

motor neurons. We create place cells from allothetic visual and olfactory cues. Place cells are connected to motor neurons, which produce certain motor actions. The rat has to learn appropriate motor actions, which eventually lead to the food source. As a consequence, sensory inputs as well as formation of place fields are affected whenever the rat navigates in the environment, thus closing the loop as shown in Fig. 4.1.

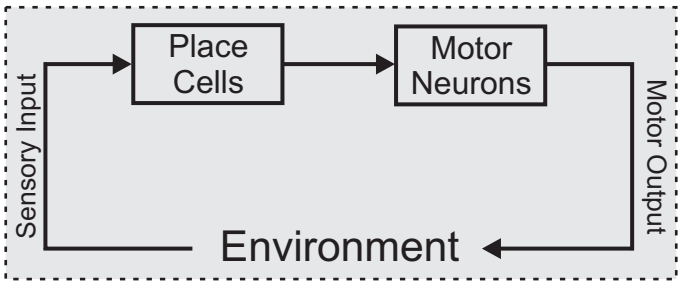

Figure 4.1: Schematic diagram of the closed loop system.

Different models have been proposed for hippocampal place cell formation including Gaussian functions (O'Keefe and Burgess, 1996; Touretzky and Redish, 1996; Hartley et al., 2000; Foster et al., 2000), back-propagation algorithm (Shapiro and Hetherington, 1993), auto-associative memory (Recce and Harris, 1996), competitive learning (Sharp, 1991; Brown and Sharp, 1995), neural architecture based on landmark recognition (Gaussier et al., 2002), neuronal plasticity (Arleo and Gerstner, 2000; Arleo et al., 2004; Strösslin et al., 2005; Sheynikhovich et al., 2005; Krichmar et al., 2005), independent component analysis (Takács and Lőrincz, 2006; Franzius et al., 2007), self organizing map (Chokshi et al., 2003; Ollington and Vamplew, 2004) or Kalman filter (Bousquet et al., 1998; Balakrishnan et al., 1999). None of these, however, addresses the question of how multiple sensory inputs might affect place field formation. Experiments with rodents demonstrate that visual cues play an important role for the control of place cells (Muller and Kubie, 1987; Knierim et al., 1995; Collett et al., 1986; O'Keefe and Speakman, 1987; Maaswinkel and Whishaw, 1999; Dudchenko, 2001). On the other hand, in the absence of visual cues rats can rely on other cues such as olfactory, auditory or somatosensory stimuli (Hill and Best, 1981; Carvell and Simons, 1990; Maaswinkel and Whishaw, 1999; Wallace et al., 2002a). Thus, it seems reasonable to consider the influence of such cues also on the formation of place fields. This view is supported by the observation that place fields become unstable when olfactory cues are removed, suggesting that olfactory cues are important in the formation and stability of place fields (Markus et al., 1994; Save et al., 2000).

Other types of cells related to hippocampal place cells and spatial navigation are head direction cells and grid cells. Head direction cells are found in found in

many brain areas including postsubiculum, the thalamus, lateral mammillary nucleus, dorsal tegmental nucleus, and striatum (Taube et al., 1990a,b; Muller et al., 1996; Knierim et al., 1998). Head direction cells respond maximally when the animal's head is oriented in preferred direction in the horizontal plane. Like place cells, head direction cells are under control of distal stimuli, and have different preferred directions in different environments. Experimental data suggests that the head direction cell system may orient the place cell system (Jeffery and O'Keefe, 1999; Calton et al., 2003; Yoganarasimha and Knierim, 2005).

Grid cells are found in entorhinal cortex (Hafting et al., 2005; Sargolini et al., 2006; Barry et al., 2007). Grid cells, like place cells, also fire strongly when an animal is in specific locations in an environment, but differ from place cells in that they have multi-peak firing fields which are organised into a hexagonal grid. It has been suggested that grid cells may make associations between places and events which is needed for the formation of memories (Hafting et al., 2005).

Many experimental studies have been performed on goal directed learning in rodents (Barnes et al., 1980; Morris, 1984; Prados and Trobalon, 1998; Lavenex and Schenk, 1998; Maaswinkel and Whishaw, 1999; Wallace et al., 2002a; Etienne and Jeffery, 2004; Jeffery et al., 2003; Hines and Whishaw, 2005). Navigation models based on place cells usually address goal learning by using reinforcement learning algorithms (Arleo and Gerstner, 2000; Arleo et al., 2004; Strösslin et al., 2005; Sheynikhovich et al., 2005; Krichmar et al., 2005) where place cell representation is based on the combination of visual information and information provided by head direction cells or path integration.

Path integration was considered by many researchers as evidence for an additional mechanism when navigating in the absence of visual cues (for a review see Etienne and Jeffery, 2004). Experimental data suggests that grid cells may be related to the path integration system (Hafting et al., 2005; Sargolini et al., 2006; McNaughton et al., 2006). However, Save et al. (2000) have shown that path integration alone is not sufficient to maintain stable receptive fields of place cells when rats navigate in the dark. Without additional cues, path integration leads to an accumulation of errors in direction and distance, and it thus needs to be reset through position information from stable cues (Etienne et al., 1996, 2004). In the study of Strösslin et al. (2005) the authors claim that their model is able to work in the dark based on self-motion cues (visual cues together with path integration were used), yet it is unclear how the model can succeed if visual cues used for recalibration are not available while navigating for a longer time in the dark.

Thus, for navigation in natural environments it seems reasonable to consider other sensory inputs, and it is known from the literature that rodents can form spatial representations based on olfactory cues and use this information for spatial orientation and navigation (Tomlinson and Johnston, 1991; Lavenex and Schenk, 1995, 1996, 1998). Experiments show that rats can track odours or self-generated scent marks to find a food source (Wallace et al., 2002a, 2003). To accommodate these findings, we

propose a novel navigation mechanism based on self-marking by odour patches combined with a reinforcement learning (Q-learning) algorithm based on multi-sensory formed place fields in order to improve spatial navigation.

Studies show that rats use visual and/or olfactory cues when available, and that such allothetic cues dominate over path integration (ideothetic component) information (Maaswinkel and Whishaw, 1999; Whishaw et al., 2001). Therefore, the focus of the current chapter is on place cell formation and spatial navigation in cue-rich, illuminated environments, where path integration would be extraneous.

Another interesting consideration concerns the question how navigation is affected by remapping. It is known that place fields change very quickly when the rat is confronted with a new environment and that many place fields will re-obtain their former properties as soon as the animal returns to the initial environment (Muller and Kubie, 1987; Wilson and McNaughton, 1993; Shapiro et al., 1997; Tanila et al., 1997; Knierim et al., 1995, 1998). It is, however, an unresolved question how remapping affects navigation and navigation (re-)learning (Jeffery et al., 2003).

In this chapter, we concentrate on the impact of olfactory cues on place cell formation and on goal navigation learning in different environments. We focus on the following three questions: 1) What is the contribution of olfactory cues to the formation of place cells and goal navigation? 2) Can goal navigation based on place cells be improved by additional navigation mechanisms? 3) How does the remapping of place fields influence goal navigation when switching between different environments?

The chapter is organised as follows. First we describe the sensory inputs and the model of the system. Then we present different goal navigation strategies and thereafter we show the results of place cell analysis, and a comparison of the presented navigation algorithms. Finally, we discuss our results and relate them to other studies and biological data.

4.2 Sensory input

We use a square box with dimensions of 10000×10000 units (discrete environment) where the walls of the arena are marked by different landmarks (see Fig. 4.2 A). Visual and olfactory cues are used as allothetic inputs to form place cells in our model.

4.2.1 Visual input

As visual input, we use the perpendicular distances from the rat's position to all four walls (Fig. 4.2 A), similar to many other models which use distances to walls or landmarks (Sharp, 1991; Recce and Harris, 1996; O'Keefe and Burgess, 1996; Touretzky and Redish, 1996; Hartley et al., 2000; Ollington and Vamplew, 2004). Let us define the visual input by $v^k_{x,y}$, where x and y denote the position in the environment and $k = 1 \ldots 4$ is the number of possible visual inputs related to the four walls of the

arena. In our model the rat has a view-field of 180 *degrees* (real rats have a wider field of view), which means that the rat can see only the walls which are ahead, but can not see what is behind. Prediction of the distance to a non-visible wall is made by taking the last estimate of distance to the wall when it was visible. This can be described by the following recurrent equation:

$$v^j_{x,y}(t) = v^j_{x,y}(t-1), \tag{4.1}$$

where j denotes the index of the non-visible wall, and t denotes the time in steps. Note that if the rat is moving along a linear trajectory away from a non-visible wall then the error of the estimate of this wall accumulates over time. The estimate is re-calibrated as soon as the wall becomes visible again.

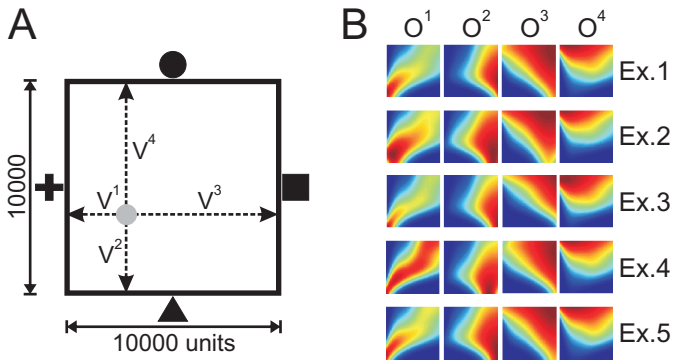

Figure 4.2: **A)** Image of square arena with landmarks. Perpendicular distances from rat's position (grey dot) to all four walls of square arena are used as visual stimuli. **B)** Examples of odours used as olfactory stimuli to the rat. Blue colour corresponds to low and red to high intensity of the odour. Five examples (Ex.1-Ex.5) are shown where each box represents a different odour coming from a different location in the environment.

4.2.2 Olfactory input

We also use four different odours as an additional input to the place cells. Five examples of odours are shown in Fig. 4.2 B, where each box represents a different odour with a different source location in the environment. We model odours at the ground level (2D space) by the following Gaussian functions:

$$o^k_{x,y} = e^{-\left(\frac{\left[a\,(x-s^k_x+\xi^k_x)\right]^2}{2\sigma^2_y}+\frac{\left[a\,(y-s^k_y+\xi^k_y)\right]^2}{2\sigma^2_x}\right)}, \tag{4.2}$$

with $\sigma_x = 15 + a \cdot x + 5\,sin(0.1\,a \cdot x)$, and $\sigma_y = 15 + a \cdot y - 5\,sin(0.1\,a \cdot y)$, where x and y denote the position in the environment, $k = 1 \ldots 4$ is the number of the odour sources, and $a = 0.01$ is the scaling factor. The variables $s^k_{x,y}$ denote the coordinates of the centre (maximum intensity) of the odour source and are given as follows: $s^1_{x,y} = (100, 100)$, $s^2_{x,y} = (9900, 100)$, $s^3_{x,y} = (9900, 9900)$ and $s^4_{x,y} = (100, 9900)$. Values ξ^k_x and ξ^k_y are randomly drawn from a Gaussian distribution with zero mean and a standard deviation of 100. Note, that here we model static odours that do not change during different runs of the same experiment but differ across experiments. The rat can smell the odours locally, and it does not sense the direction of the odour source.

Noise is also added to the visual sensory inputs, assuming that the rat makes larger errors in the estimation of long distances. Similarly, the rat makes larger errors in estimating odours with low intensity and smaller errors for odours with high intensity. This is given by the following equations:

$$V^k_{x,y} = \frac{v^k_{x,y} + 0.03\,v^k_{x,y}\eta^k_v}{L}, \tag{4.3}$$

$$O^k_{x,y} = \frac{o^k_{x,y} + 0.03\,(1 - v^k_{x,y})\eta^k_o}{\max_{x,y} o^k_{x,y}}, \tag{4.4}$$

where η^k_v and η^k_o are random values from a uniform distribution within the interval [-1;1]. Note, that both visual and olfactory inputs are normalised and bounded within the interval [0;1], where $L = 10000$ units is the size of the environment.

4.3 Place cell model

We model place cells by using a simple feed-forward network with an input and an output layer as shown in Fig. 4.3 A. At the input layer we have sensory inputs $X : [V^k_{x,y}, O^k_{x,y}]$ received from visual and olfactory stimuli. Here we have a fully-connected network where every neuron in the input layer is connected to every neuron in the output layer via connection weights $W^i = [w^{i,1} \ldots w^{i,n}]$, where $i = 1 \ldots N, N = 500$ is the total number of place cells and n is the number of sensory inputs ($n = 4$ if only visual cues are used and $n = 8$ if both visual and olfactory cues are used). Weights are initialised randomly by a function f_z:

$$f_z = \left(1 + e^{\frac{z-m}{2\sigma^2}}\right)^{-1}, \tag{4.5}$$

where z is a random number from a uniform distribution within the interval [0;1], $m = 0.5$ and $\sigma = 0.2$. The distribution of initial weights is plotted in Fig. 4.3 C. We have chosen such a distribution for the reason that if the weights are initialised according to a uniform distribution then all place field centres are located around the

centre of the environment and we do not obtain place fields close to the walls of the environment. In our model weights are basis vectors, which are used to compute firing rates of place cells (see equation below) where we start with a random initialisation of basis vectors. By employing competitive learning, cells become tuned to a specific input, which leads to the spatial selectivity of the place cells.

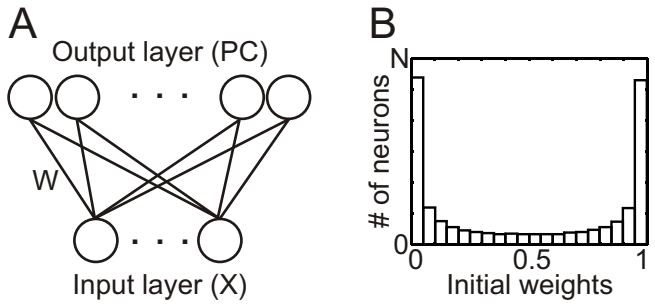

Figure 4.3: **A)** A simple feed-forward network with sensory inputs x at the input layer, connection weights w and place cells (PC) at the output layer. **B)** Distribution of initial weights of the neural network (A).

The firing rate of place cell i is expressed by a Gaussian function (similar to O'Keefe and Burgess, 1996; Hartley et al., 2000) and is computed as follows:

$$r_t^i = e^{-\frac{\left[\frac{1}{n}||X_t - W_t^i||\right]^2}{2\sigma_f^2}}, \tag{4.6}$$

where $\sigma_f = 0.07$ defines the width of the place field, n is the dimension of the input space, and the norm is the Euclidean distance. Weights of our neural network are modified according to a winner-takes-all mechanism where we change only the weights of the best matching unit β_t:

$$\beta_t = \underset{i}{\operatorname{argmin}} ||X_t - W_t^i||. \tag{4.7}$$

Weights of the winner neuron β_t are changed according to the following equation:

$$W_{t+1}^{\beta_t} = W_t^{\beta_t} + \mu(X_t - W_t^{\beta_t}), \tag{4.8}$$

where $0 < \mu \ll 1$ is the rate factor.

4.4 Formation of place fields

In the following we are going to present results on place fields obtained by our place cell model and provide an analysis on place cell directionality.

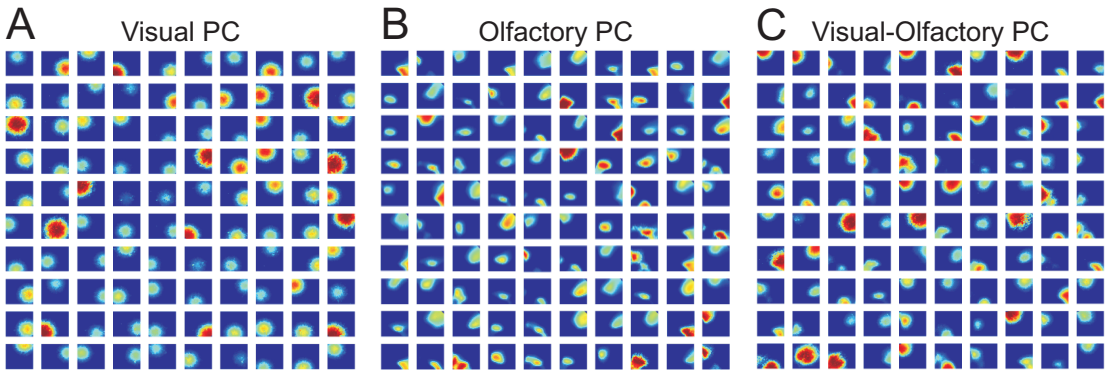

Figure 4.4: Examples of place fields (100 out of a total of 500 cells). **A)** PFs obtained when using visual cues alone. **B)** PFs obtained when using olfactory cues alone. **C)** PFs obtained when using both, visual and olfactory, cues. Selected place fields with maximum firing rate $r > 0.5$ are shown for each case.

4.4.1 Place fields

Examples of place fields (PF) after random exploration over 5000 time steps are presented in Fig. 4.4. PFs obtained when using visual or olfactory cues alone are shown in panel A and B. PFs obtained from both visual and olfactory cues are shown in panel C. Here we show only selected PFs which have a maximum firing rate of $r > 0.5$. Resulting PFs are localised, differ in size and firing rate, and are similar to real PFs. For examples of place fields obtained from the rodent hippocampus see Wilson and McNaughton (1993); O'Keefe (1999).

For the formation of place fields we used a relatively low rate factor ($\mu = 0.01$) to develop connection weights between input and output layer (see Fig. 4.3 A), because weights oscillate and do not converge when a high rate factor ($\mu = 0.1$) is used, and this does not lead to the final stabilisation of place cells. For comparison of weight development for different rate factors see Fig. 4.6 A. The distribution of firing rates is shown in Fig. 4.5 C, where we have fewer cells with a high firing rate than cells with a low firing rate, which resembles experimental data (Hartley et al., 2000). Some of the cells which are silent in a specific environment become active when moved to the other environment (see Fig. 4.16). PF centres from a single experiment (location of maximum firing rate within the field) are shown in Fig. 4.5 B, where circles represent centres of PFs with a low firing rate ($r \leq 0.5$) and dots those with a high firing rate ($r > 0.5$). We observed that cells with low firing rate are distributed around the centre of the environment (similar to Gaussian distribution, panel D) whereas cells with high firing rate are evenly distributed within the whole environment (see panel E). The latter cells will drive the learning in the goal navigation task (see section 4.5.2).

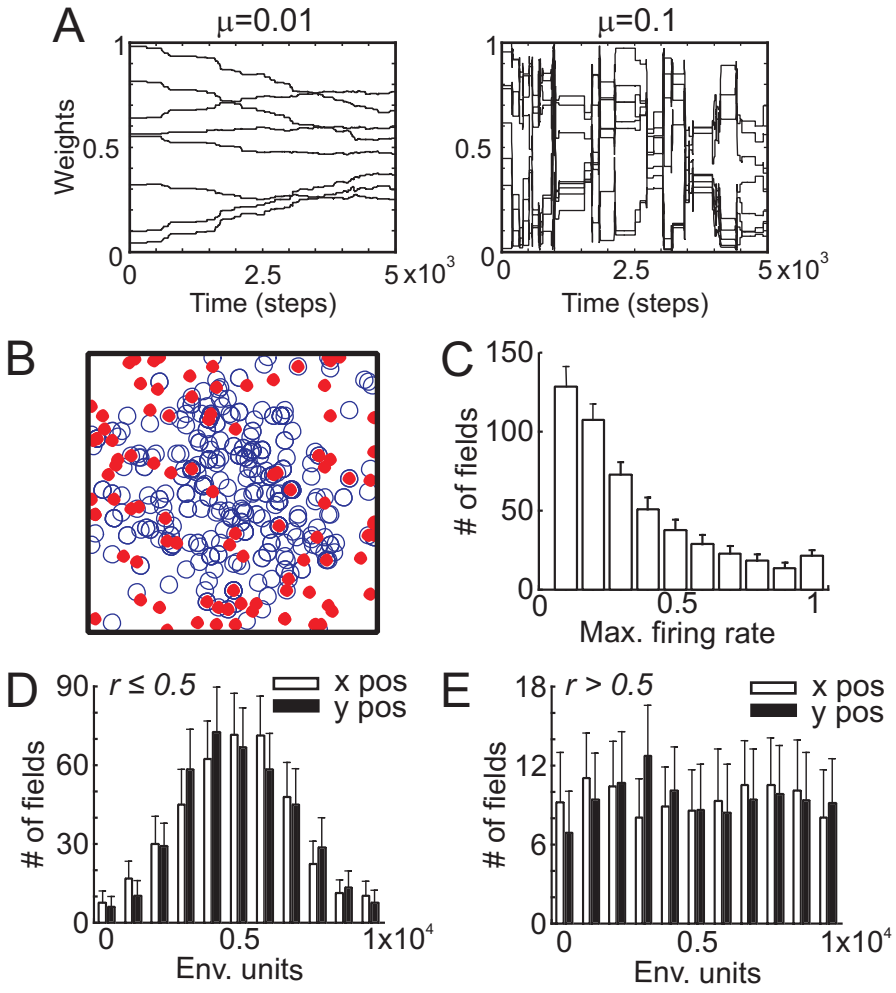

Figure 4.5: **A)** Connection weights between input neurons and place cells (see Fig. 4.3 A) depending on the rate factor μ. **B)** Distribution of place field centres within the environment from single experiment. Dots denote centres of place fields with maximum firing rate $r > 0.5$ whereas circles denote centres of fields with maximum firing rate $r \leq 0.5$. **C)** Distribution of maximum firing rates r of 500 cells; average and standard deviation (SD) for 100 experiments. **D, E** Distribution of x and y position of place cell centres with maximum firing rate $r \leq 0.5$ (C) and $r > 0.5$ (D); average and standard deviation (SD) for 100 experiments.

4.4.2 Place cell analysis

Before looking at the comparison of goal navigation strategies we would like to investigate the contribution of the olfactory input to place cell formation. This influence

can be assessed by measuring the directionality of place cells. Here we let the rat to explore the environment randomly as shown in Fig. 4.6 A for 5000 time steps (development phase). After the development phase we let the rat move in the environment for another 5000 time steps to create test data. To evaluate the directionality of place cells we looked at the locations which had been passed by the rat in different directions. We say that a cell is omnidirectional, i.e. independent of the movement direction, if at a given location the cell fires with its highest firing rate regardless of crossing the location in different directions.

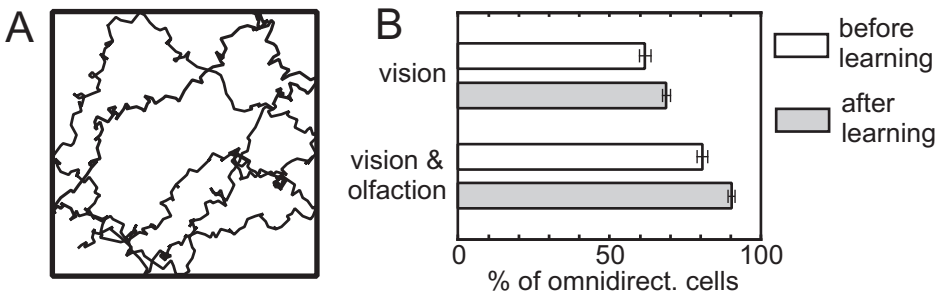

Figure 4.6: **A)** Example of the rat's trajectory when the rat explores the environment randomly. **B)** Percentage of omnidirectional cells before and after learning (rate factor $\mu = 0.01$). The average together with confidence intervals (95%) is shown in 20 experiments.

Averaged results of 20 experiments are presented in Fig. 4.6 B where we compare the directionality of place cells obtained from visual cues alone with that obtained from both visual and olfactory stimuli. The white bars show the control case, with place cell directionality before the development phase (i.e. before learning). We can see that we obtain more omnidirectional cells when we use combined stimuli compared to visual stimuli alone and more omnidirectional cells develop during the development phase compared to the control case. The improvement in omni-directionality when using olfactory cues can be explained by the fact that perception of olfactory cues is direction independent whereas perception of visual cues depends on local views. Note that the view-field influences the directionality of place fields. The larger the view-field, the fewer directional cells are obtained. Since the rats do not have the omnidirectional view we still would get more directional cells obtained from visual information alone compared to combined stimuli (visual and olfactory cues) or olfactory cues alone. Our results on place cell directionality are qualitatively similar to experimental data of Battaglia et al. (2004). For further discussion on place cell directionality see section 4.9.1.

4.5 Navigation strategies

4.5.1 Goal navigation task

The rat has to learn to navigate from its home location to the goal, i.e the food source. The rat can use allothetic visual and olfactory cues described above but it can not see or smell the food source (similar to the Morris water maze task[1], Morris, 1984). The rat gets a reward only when it approaches the goal location. The setup for such a spatial task is shown in Fig. 4.7 A. We use the same discrete environment (square box) as described above, where we have different landmarks on all four walls (see Fig. 4.2 A). The home location of the rat is in the bottom-left corner, 1000 units from both walls and is marked by a grey dot. The dimensions of the food source, marked by a square, are 2000×2000 units and it is located 3000 units from the left wall and 2000 units from the upper wall. At the beginning, the rat explores the environment randomly and finds the goal just by chance as shown by trajectory (1) in panel A, whereas after a few learning runs the rat finds a more or less direct path (trajectory (2)) to the food source. Whenever the rat finds the food location we start a new run from the start position (home location). A maximum number of 200 steps is allowed for one run with a step size in the range of 400-600 units. In our model during the first run in most of the cases (80%) the rat finds the goal within less than 200 steps, so the rat has enough time to find the goal even when navigating randomly. Another reason for the 200 step limit is related to the frustration phenomenon observed in animals where creatures return to "home-base" if the goal is not found within an expected time (Eilam and Golani, 1989; Whishaw et al., 2001; Wallace et al., 2002b; Hines and Whishaw, 2005; Nemati and Whishaw, 2007).

4.5.2 Q-learning with function approximation

As a first approach we apply reinforcement learning (Sutton and Barto, 1998) as used by other studies on hippocampus-based navigation (Arleo and Gerstner, 2000; Arleo et al., 2004; Foster et al., 2000; Strösslin et al., 2005; Krichmar et al., 2005). Here we employ a version of Q-learning with function approximation similar to Reynolds (2002). The algorithm is implemented by a three layer neural network (see Fig. 4.7 B) where place cells are formed from sensory input as described in section 4.3. The place cells are connected to motor neurons representing eight directional cells: north (N), north-east (NE), east (E), south-east (SE), south (S), south-west (SW), west (W) and north-west (NW). The actual direction of movement is determined by

[1]Morris water maze task is a task where a rat or mouse is placed into a small pool of water (usually 1.2 to 1.8 meter) which contains an escape platform hidden a few millimeters below the water surface. Escape from the water reinforces the rat to quickly find the platform, and on subsequent trials the rat is able to locate the platform faster.

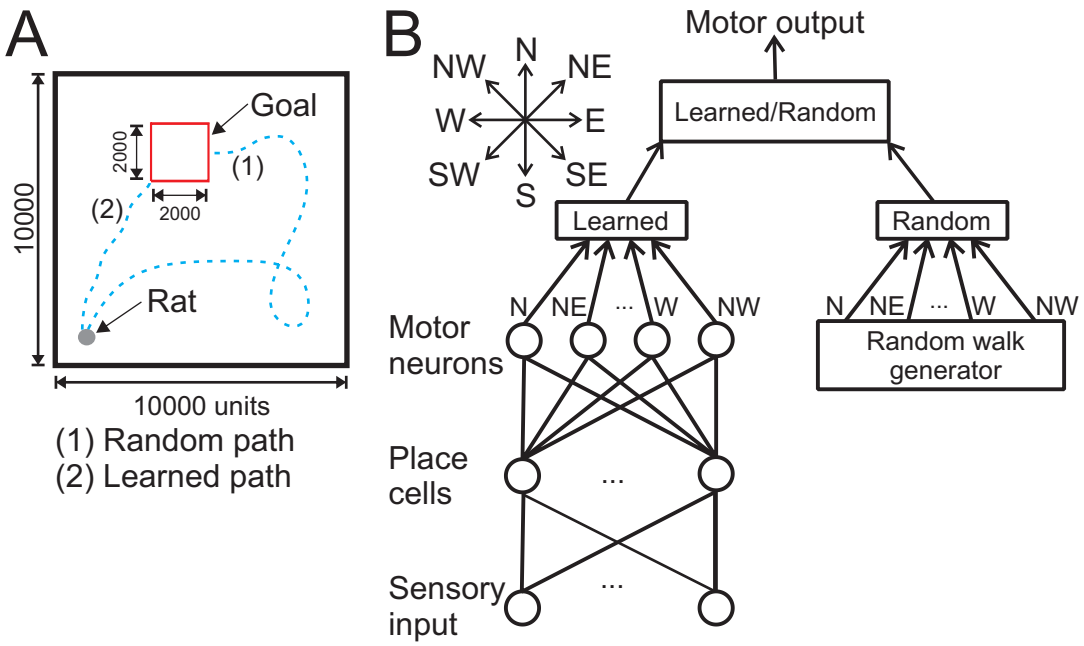

Figure 4.7: **A)** Environmental setup of the goal navigation task. We used a discrete square arena with dimensions of 10000 × 10000 units and a goal (food source) with dimensions of 2000 × 2000 units. The starting position of the rat was 1000 units from both left and bottom walls, whereas the location of the food source was 3000 units from the left wall and 2000 units from the upper wall. **B)** Neuronal setup of the model rat's navigation system. Place cells are formed from sensory input as described in section 4.3. Each place cell in the network is connected to eight motor neurons (eight directions). The rat makes a movement to the direction which has the strongest connection between place cells and motor neurons for eight directions averaged over all cells, which are firing at the present location. The rat makes a random movement whenever the connection weights are zero at the present location.

the maximum Q-value of the eight possible directions averaged over all cells, which are firing at the present location, with additional noise. For example the horizontal movements W or E are given by the following simple equations:

$$\begin{aligned} \Delta x &= \pm(\Delta s + b \cdot \eta_x), \\ \Delta y &= b \cdot \eta_y, \end{aligned} \tag{4.9}$$

where $\Delta s = 500$ is the step size, η_x and η_y are random values from a uniform distribution within the interval [-1;1], and $b = 100$ is the amplitude of the noise. Here we use the minus sign for the W direction and the plus sign for the E direction. Similarly,

for SW or NE we have:

$$\begin{aligned} \Delta x &= \pm(\tfrac{\Delta s}{\sqrt{2}} + b \cdot \eta_x), \\ \Delta y &= \pm(\tfrac{\Delta s}{\sqrt{2}} + b \cdot \eta_x), \end{aligned} \tag{4.10}$$

and the equivalent for the other directions. The rat makes a random movement whenever Q-values are zero at the present location. In this case, the rat keeps the direction of the movement with a probability of $1 - p_r$, whereas with $p_r = 0.25$ it will randomly take a new direction. When Q values are non-zero we use a usual RL strategy, with exploration and exploitation, where the direction of the movement is chosen according to the learned Q-values most times, (exploitation probability $1-p_e$), and a random move is made with exploration probability $p_e = 0.1$.

As mentioned before, the learning mechanism from place cells to motor cells is a version of Q-learning with function approximation (QLFA). Let us define our basis functions Φ_i as a function of the firing rate r_t of the place cell i at the time step t:

$$\Phi_i(r_t) = \begin{cases} 1 & \text{if } r_t^i > 0.5, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

Here, $i = 1 \ldots N, N = 500$ is the total number of place cells. Note, we discretise the space representation provided by place cell prior to the goal-navigation learning in order to reduce the amount of noise in the place field system since low firing rates give larger errors in position estimation compared to the real position of the rat in the environment. By using binary cells we still get different place field sizes and we preserve the directionality of place cells.

We define the action-value function by the following equation:

$$Q(r_t, a_t) = \frac{\sum_i \Theta_{i,a_t} \Phi_i(r_t)}{\sum_i \Phi_i(r_t)}, \tag{4.12}$$

where $\Theta_{i,a}$ is the weight from the $i-th$ place cell to the motor action a. In the given equation we sum over all basis functions, but at a specific location within the environment only a specific subset of basis functions will be non-zero. We use an averaging Q-learning rule according to Reynolds (2002) where we update weights Θ_{i,a_t} of the actually taken action a_t at the time step t according to the following learning rule:

$$\Theta_{i,a_t} = \Theta_{i,a_t} + \alpha(R_{t+1} + \gamma \max_a Q(r_{t+1}, a_{t+1}) - \Theta_{i,a_t})\Phi_i(r_t), \tag{4.13}$$

where $\alpha = 0.7$ is the learning rate, $\gamma = 0.7$ is the discount factor and R is a reward. We define our reward function R_t by

$$R_t = \begin{cases} 1 & \text{if the rat has found the goal,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.14}$$

4.5.3 Self-marking navigation

The second approach in our study is to use navigation based on self-generated odour marks, where the rat follows the self-laid scent marks to find the food source. The rat always explores the environment randomly by keeping the direction of the movement whenever it does not smell anything locally. Note that the rat can smell only within a given radius of 600 points, which corresponds to the maximum step size. At the beginning, the rat finds the food source by moving randomly and marks it by a small amount of scent. In the next runs, when the rat approaches the previously laid scent mark within a distance at which the rat can smell it, the rat will mark its location and then will go directly to the perceived scent mark and remark it again by another small amount of scent. The whole navigational process can be defined as follows. The rat marks the location of the food source or remarks the current location if it smells another scent mark/marks ahead by

$$u_{t+1}^{x,y} = u_t^{x,y} + \Delta u, \tag{4.15}$$

where u defines the self-laid odour patches in the environment, x, y define coordinates of the position within the environment and $\Delta u = 0.005$. The locations which have strong smell, i.e. $u_t^{x,y} = 1$, are not remarked anymore. The rat goes directly to the location $l_t^{x,y}$ marked by scent mark which has the strongest smell according to

$$l_t^{x,y} = \underset{x,y}{\operatorname{argmax}}\, u_t^{x,y}, \tag{4.16}$$

otherwise it makes a random movement as explained above. It is worth noting that the given method propagates scent-marks backwards from the location of the reward as in reinforcement learning, but here we do not have predefined features. Instead, we create them "on the fly", and we do not directly memorise action values associated to states, where a state is defined by the rat's position in the environment x, y. In our model self-laid scent marks are modelled by little "drops" which are less intense relative to the environmental odours which may have very strong odour sources and diffuse within the environment. Self-generated odour marks can be smelled and distinguished by the rat only locally within a relatively small radius (in our case within one step size).

4.5.4 Combining Q-learning with self-marking navigation

The third and the last approach is a combination of the two previously described methods. In this case the rat marks the location only if it smells another scent mark/marks *and* the normalised maximum Q-value at this location obtained by using the first method has reached a given threshold of $\lambda = 1.5$:

$$u_{t+1}^{x,y} = \begin{cases} u_t^{x,y} + \Delta u & \text{if } \frac{\max_a Q(r_t,a_t)}{\frac{1}{8}\sum_a Q(r_t,a_t)} > \lambda, \\ u_t^{x,y} & \text{otherwise.} \end{cases} \tag{4.17}$$

The action in the combined strategy is taken by the following rule. If the rat does not smell any scent mark within given radius then it takes an action according to the Q-values, otherwise the rat follows the scent gradient. By using this type of navigation the rat develops Q values and lays scent marks at the same time.

4.6 Goal navigation

In the following we will present navigation results on different navigation strategies from single experiments and later on we will compare navigation performance of different strategies by providing statistical analysis.

4.6.1 Navigation using Q-learning based on place cells

An example of navigation by using Q-learning based on place cells formed from combined visual and olfactory cues is presented in Fig. 4.8. Trajectories of the rat's paths obtained from 30 runs are shown in panel A, and the number of steps needed to reach the goal versus number of runs are plotted in panel B. The rat found a more or less straight path to the goal after seven trials.

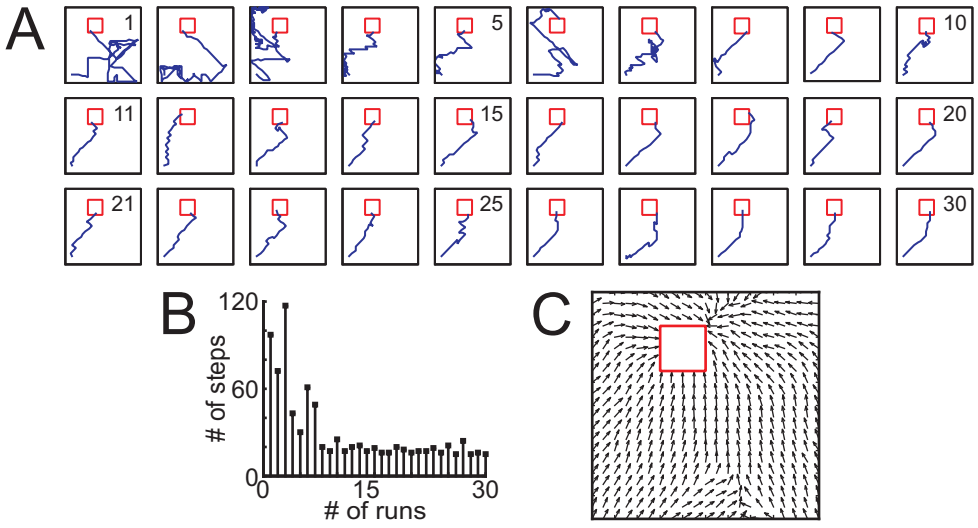

Figure 4.8: Results from single experiments using Q-learning navigation algorithm based on place cells obtained from visual and olfactory cues. **A, B)** Goal navigation from home location. **A)** Trajectories of rat's paths from 30 runs. **B)** Number of steps needed to reach the goal versus number of runs (B). **C)** Goal navigation from random start position. Diagram shows vector field representation of learnt actions.

Results obtained from a single experiment when looking for a goal from a *random* start position are presented in Fig. 4.8 C, where in every trial the rat was placed randomly within the environment. A vector field representation of learnt Q-values after 100 runs is shown where each vector represents the cumulative direction of movement from corresponding location. The vector field was calculated according to the following procedure. A 20×20 grid was used to define specific points in the environment. Corresponding subsets of place cells were found, which fire at each intersection point of the grid. Average Q-values for eight directions were calculated for the corresponding subset of place cells. The resulting movement direction vector was computed from the obtained average Q-values for each intersection point of the grid.

4.6.2 Self-marking navigation

Results for self-marking navigation are shown in Fig. 4.9. Trajectories of the rat's paths obtained from 60 runs are presented in panel A. The environment with self-laid scent marks (marked as dots) is shown in panel B, where the dot's size is proportional to the strength of the scent mark. The rat follows the scent gradient to find the food source. The number of steps needed to reach the goal versus number of runs is plotted in panel C, where the rat had generated the trail of scent marks, which leads from the home location to the food source after 56 runs (see the last four trajectories in panel A).

In Fig. 4.9 D we show the resulting map of self-laid scent marks (marked by dots) from self-marking navigation after 200 runs when looking for a goal from a random start position. Here we use more runs since self-marking navigation converges slower than Q-learning (see Fig. 4.8 B). When starting from random positions, the rat creates a map of a tree-like structure of scent marks, where it chooses the closest branch and then follows the gradient of scent marks, leading to the goal.

4.6.3 Combined navigation

One example of navigation with combined strategies is shown in Fig. 4.10 where trajectories of the rat's paths obtained from 30 runs are presented in panel A and the number of steps needed to reach the goal versus number of runs in panel C. In this experiment the rat found a more or less straight path to the food source already after five runs. From the given example we can see that scent marks (panel B) are laid only along the way to the food source whereas in the previous example of self-marking navigation scent marks (see Fig. 4.9 D) are spread out widely throughout the environment.

Results of combined navigation when starting from random positions are presented in panel Fig. 4.10 D, where we show the vector field of learnt actions (left) and the corresponding map of scent marks (right) after 100 runs. As expected we obtained

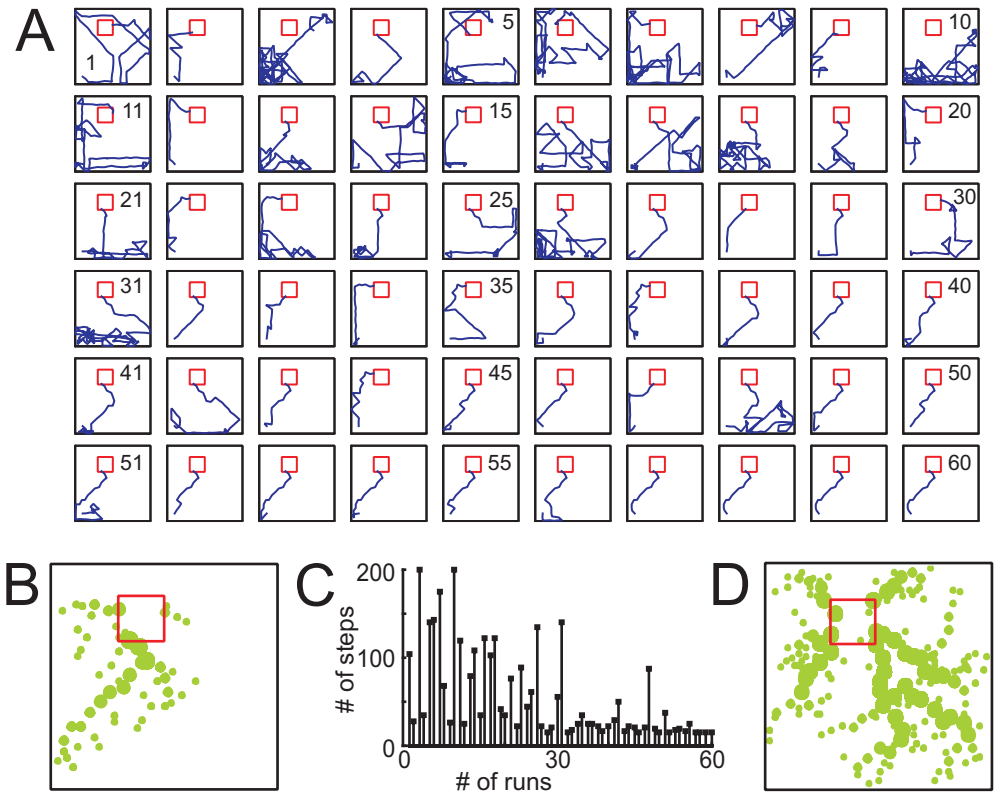

Figure 4.9: Results from single experiments using self-marking navigation (no place cells). **A-C)** Goal navigation from home location. **A)** Trajectories of rat's paths from 60 runs. **B)** Environment with self-laid scent marks (marked by dots) is shown where larger dots represent stronger scent. The rat follows the trail of scent marks to find the goal. **C)** Number of steps needed to reach the goal versus number of runs. **D)** Goal navigation from random start position. Diagram shows vector field representation of learnt actions.

similar results to those of self-marking and Q-learning navigation (see Fig. 4.8 C and Fig. 4.9 C).

In general, we observed that when starting from the same position (home location) the rat creates one main trail of scent marks, whereas when starting from a random location the rat creates tree-like structures of scent marks with several main branches. Also, the rat creates more scent marks when using pure self-marking navigation compared to the combined strategy.

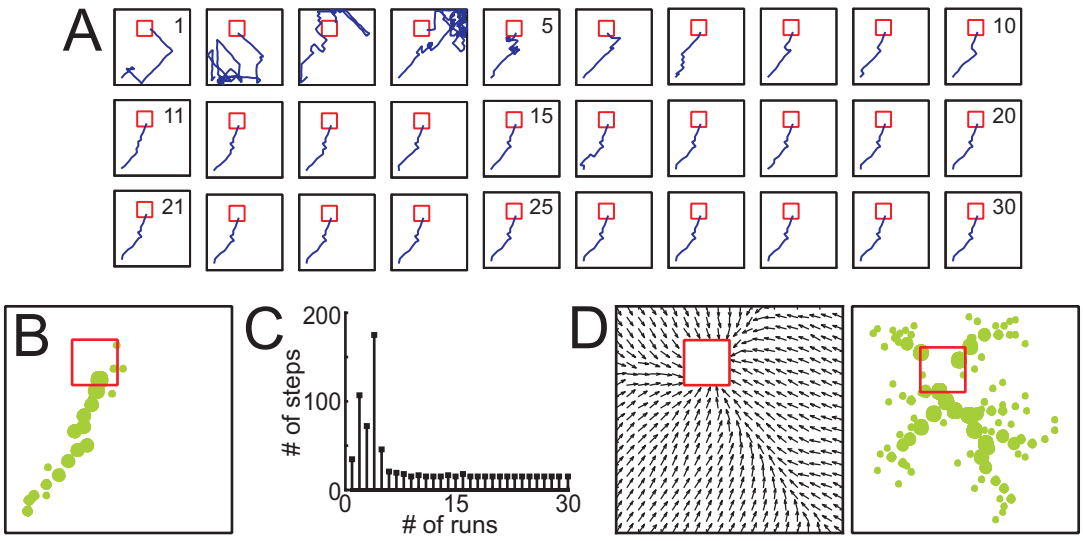

Figure 4.10: Results from single experiments using combined navigation strategy where Q-learning based on place cells obtained from visual and olfactory cues is combined with the self-marking navigation. **A-C)** Goal navigation from home location. **A)** Trajectories of rat's paths from 60 runs. **B)** Environment with self-laid scent marks (marked by dots). **C)** Number of steps needed to reach the goal versus number of runs. **D)** Goal navigation from random start position. Diagrams show vector field representation of learnt actions (left) and corresponding self-laid scent marks (right).

4.6.4 Navigation in environments with multiple targets

We also investigated the performance of self-marking navigation and the combined navigation strategy in the environment with multiple targets. For this experiment we used an environment with two food sources as shown in Fig. 4.11 A, where in one case the rat always started to search for food from the same start position (home location) and in the other case the rat was placed at a random position.

Results of a single experiment for self-marking navigation when always starting from the home location are shown in Fig. 4.11 B, where we show a map of self-laid scent marks after 200 runs. In the beginning the rat back-propagates scent marks from both goal locations, where at the end it creates a stronger trail of scent marks, which leads to only one of two food sources (see left and right sub-panels). When starting from a random location (panel C), the rat creates a map of scent marks with a tree-like structure similar to the case with one food source (see Fig. 4.9 D). Here we obtain two trees of scent marks where each leads to one corresponding food source.

Results of combined navigation are presented in Fig. 4.11 D, E. As expected, when

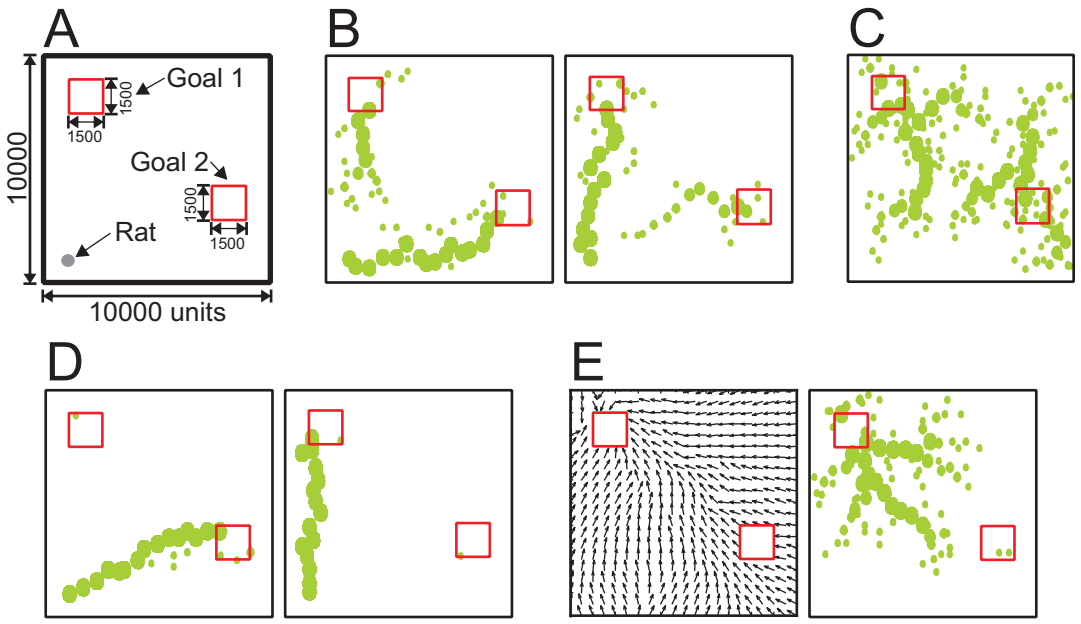

Figure 4.11: Results from single experiments using self-marking navigation and combined navigation in the environment with two targets. **A)** Environmental setup of the goal navigation task. We use a discrete square arena with dimensions of 10000×10000 units and two food sources with dimensions of 1500×1500 units (small squares). The starting position of the rat is 1000 units from both left and bottom wall. The location of the first food source is 1000 units from both left and upper wall, whereas the location of the second food source is 1000 units from the right wall and 2500 units from the bottom wall. **B, C)** Self-marking navigation: self-laid scent marks obtained for the same start position (B) and for random start position (C). **D, E)** Combined navigation: self-laid scent marks obtained for the same start position (D); E - vector field of learnt Q-actions (left) and self-laid scent marks (right) obtained for random start position.

starting from the home location (panel D), the rat marks only one route. Note, as opposed to self-marking navigation (panel B), the rat back-propagates scent marks only from one of the two food sources. Results for combined navigation when starting from a random location are shown in panel E where we show the vector field of learnt actions (left sub-panel) and the corresponding map of scent marks (right sub-panel). As opposed to self-marking navigation (panel C), the rat creates only one tree of scent marks, where all direction vectors point to the marked food source. This is due to the fact that in combined navigation the rat marks only the locations where Q-values are relatively high. As soon as the rat finds one of the two goals, it goes to

that goal location more often and propagates scent marks backwards (similar to the results presented in panel D).

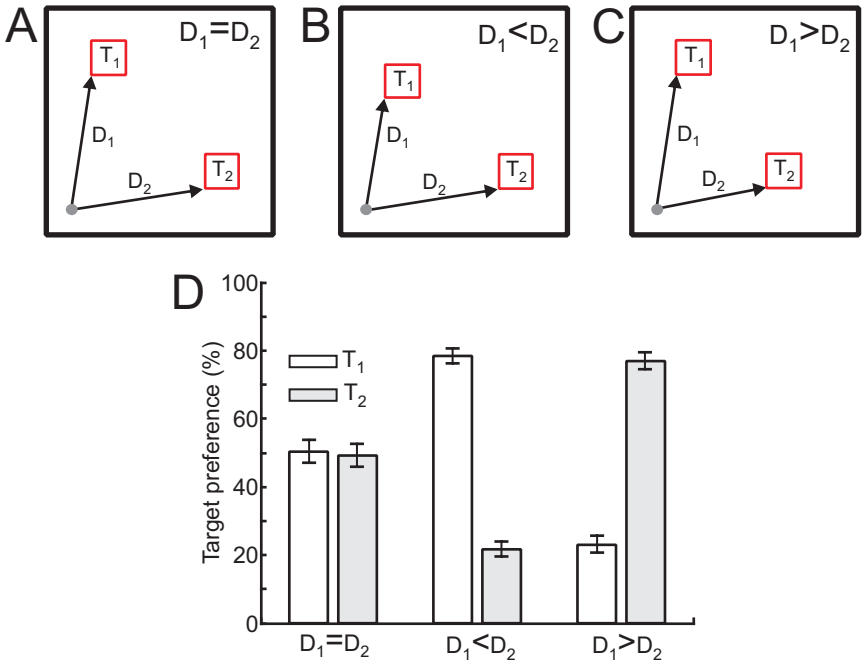

Figure 4.12: **A-C)** Environmental setups for goal navigation task with two targets. Size of the environment is 10000×10000 units, size of the targets is 1500×1500 units, and the rat's home position is 1000 units from both the left and the bottom wall. **A)** Distance $D_1$to the target T_1 is the same as to the target T_2 ($D_1 = D_2$). The location of the first target is 2000 units from the left wall and 1500 units from the upper wall, whereas the location of the second target is 1500 units from the right wall and 2000 units from the bottom wall. **B)** Target T_1 is located closer to the rat's home position as compared to T_2 ($D_1 < D_2$, $D_2 - D_1 = 940$ units). The location of the first target is 2000 units from the left wall and 2500 units from the upper wall, whereas the location of the second target is 1500 units from the right wall and 2000 units from the bottom wall. **C)** Target T_2 is located closer to the rat's home position as compared to T_1 ($D_1 > D_2$, $D_1 - D_2 = 940$ units). The location of the first target is 2000 units from the left wall and 1500 units from the upper wall, whereas the location of the second target is 2500 units from the right wall and 2000 units from the bottom wall. **D)** Results from the goal navigation task with two targets for the cases A, B and C. Average results together with confidence intervals (95%) from 100 experiments are shown for each case.

We also observed that if one of two food sources is located significantly closer to

the home location than the other, the rat in most of the cases finds the closer food source when using the self-marking navigation strategy. This is due to the fact that the rat propagates scent marks from the food sources to home location backwards, and scent-marks from the closer food source reach home location earlier than those of food source which is further away. To demonstrate this we performed the following experiments. Three environmental setups have been used. In the first case we placed two targets, T_1 and T_2 at equal distances away from the rat's home position ($D_1 = D_2$, see Fig. 4.12 A). This is a control case where we wanted to demonstrate that the rat has no prior bias to any of two targets. In the second case we placed the first target T_1 940 units closer than the second target T_2 to the rat's home position ($D_1 < D_2$, see panel B), and in the third case the second target T_2 was 940 units closer than the first target T_1 ($D_1 > D_2$, panel C). We tested the rat's target preference in all three cases by letting the rat search for the goal for ten times (one experiment consisted of 75 runs) and checked how many times out of ten the rat selected goal T_1 and T_2. We repeated this procedure 100 times for each case.

Results for this experiment are shown in Fig. 4.12 D where we plot the average target preference for each case obtained from 100 experiments. As expected we can see that for the control case, where distances to the both targets are the same, the rat has no preference to any target whereas if one of the targets is close to home location than the other then the rat in most of the cases ($\approx 80\%$) will select the target which is closer to the home location.

In summary, we observed that the rat learns a unique route which leads to one of the two targets and only in the case of pure self-marking navigation when starting from a random locations does the rat create routes to both targets.

4.6.5 Statistical evaluation of different navigation strategies

In the following we statistically determine the effectiveness of different stimuli for the goal navigation task and compare the previously described navigation strategies. The task for the rat was to find a route from home location to the food source as shown in Fig. 4.7 A. Four different navigation strategies were used for comparison as shown in 4.13 A. For a more detailed description of different navigation strategies see section 4.6).

Comparison of different navigation strategies is presented in 4.13 B, C. The average number of steps needed to find the goal versus number of runs obtained from 200 experiments is shown for each case in panel B. We obtained faster convergence when both visual and olfactory cues are used as compared to visual stimuli alone (see VQ, VOQ). This can be explained by the observation that cells formed from combined stimuli are less directional than those formed from visual cues alone. Note, that if we have a place cell system where all place cells are directional then it will require learning of actions for every movement direction of an animal for every specific location in the environment. For instance, if the rat learns the direction to the goal from a specific

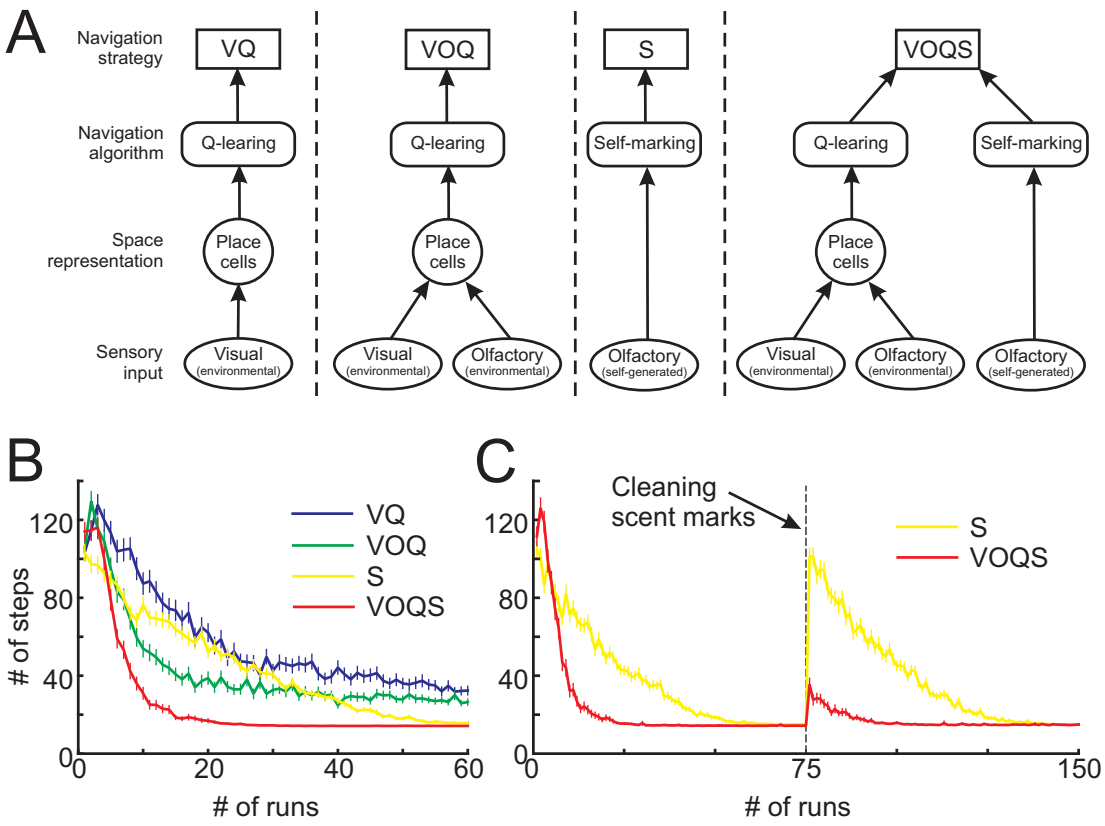

Figure 4.13: **A)** Four cases of different navigation strategies. VQ : place cells obtained from visual cues alone are used for goal navigation by using Q-learning. VOQ : similar to the case VQ, but here place cells are obtained from both visual and olfactory cues. S : Self-marking navigation (no place cells) where the rat follows self-generated marks to find a goal. $VOQS$: Combined navigation where the rat marks the location only if the Q-value (obtained from the VOQ) has reached a given threshold. **B, C)** Comparison of different goal navigation strategies. The average number of steps needed to find the goal is plotted versus the number of runs in 200 experiments. The vertical bars show the standard error mean (SEM). **C)** Comparison between the case S and $VOQS$ (see panel A) where the self-generated marks were "cleaned" after run 75.

location with a certain movement direction (e.g. north) then the rat will not know the direction to the goal from the same location when crossing this location with a different movement direction (east), since place cells will not fire when moving along this different direction. If we have an omni-directional place cell system then we learn actions for a specific location independently of the movement direction of the

animal (the same actions for all movement directions for a specific location) which as a consequence makes the learning faster. Self-marking navigation alone (S) converges much slower than Q-learning based on PCs obtained from combined stimuli (VOQ), whereas the combination of self-marking navigation with Q-learning ($VOQS$) is faster than Q-learning alone (VOQ). Note that the number of steps needed to reach the goal when using Q-learning (VQ/VOQ) is larger on average than that for self-marking navigation (S) or combined method ($VOQS$). This is due to the fact that we use a RL strategy with exploration and exploitation, where the rat tries random directions hoping to find a better path. This sometimes leads to a loss of track and long trajectories, which on average shifts the curve up. In self-marking navigation or with the combined method the rat does not explore the environment anymore as it now follows self-laid scent marks.

We also compared self-marking navigation (S) with the combined method ($VOQS$) in a task where after learning of the spatial task the self-generated marks were "cleaned" (i.e. $u(x, y) = 0$). Results are presented in Fig. 4.13 C. As expected, the rat has to relearn the path to the goal from scratch when using self-marking navigation alone, whereas the combined strategy allows the rat to use learnt Q-values (or in the other words, to navigate using allothetic visual and olfactory cues) whenever self-generated scent marks are not available anymore and it remarks the path again. The small peak with a decay after "cleaning" (see case $VOQS$) is a result of the previously discussed exploratory behaviour.

4.7 Hierarchical input preference in spatial navigation

In the presented combined strategy scent trails are used by the rat to find a goal after learning. However, this kind of strategy is inconsistent with biological findings. Maaswinkel and Whishaw (1999) showed that rats use visual cues for spatial navigation if they are available. If visual cues are not available, the rats rely on self generated odour cues. To address this problem we modified our combined navigation strategy by adding hierarchical input preference to the model.

4.7.1 Modifying the combined navigation strategy

We modified the combined navigation algorithm in the following way. At the beginning the rat uses both environmental cues and self-marking cues (combined strategy) in order to speed up learning as described above. This differs from the previous version in that the rat stops laying and following scent marks as soon as the trail of scent marks reaches the home location, whereas Q-values are still left modifiable. Furthermore, the rat prefers environmental cues (i.e. navigation based on Q-values) if they are available; if not, the rat follows previously generated scent marks. Here

we use a combined strategy (Q-learning with self-marking navigation) for learning as it makes learning faster and only later on we use the hierarchical input preference for navigation. During learning, Q-values as well as odour marks are generated where initially the Q-value development dominates in the learning and guides the placing of the odour patches since the rat lays a scent mark only if the normalised maximum Q-value at this location has reached a given threshold. As we would associate the Q-system with landmarks we find that during learning we are, due to Q-dominance, compatible with Maaswinkel and Whishaw (1999). Note, that if we were starting with the hierarchical input preference from the beginning then this would lead to a slower convergence since the rat would learn the route based on landmarks alone (without self-generated odour marks) and this would lead to the results obtained by using Q-learning algorithm alone. After learning the model allows distinguishing between different input preferences.

4.7.2 Goal navigation using hierarchical input preference

To demonstrate hierarchical input preference in spatial navigation we have performed two different experiments. In the first experiment we flipped the self-generated scent marks after learning along the diagonal of the box in a way that the scent trail does not lead to the goal anymore (see left and right panels in Fig. 4.14 A, B), where environmental cues were left unaffected. In the second experiment we removed all environmental cues (visual and olfactory) after learning and left the scent trail unaffected.

Two examples of single results from the first experiment are shown in Fig. 4.14 A and B, where in the left sub-panel we show the scent trail and the rat's trajectory at the end of learning and in the right sub-panel we show three trajectories of consecutive runs after scent marks were flipped. We found that the rat takes a correct route to the goal using environmental cues. We also noticed that the route is along the trail of scent marks that were produced during learning, which means that the rat has created two similar representations of route to the goal, where one is based on environmental cues and the other based on self-laid scent marks. After learning, the rat prefers environmental cues, so the rat's performance remains unaffected when we flip the scent marks. Statistics for 200 experiments are presented in panel C. We show the average number of steps needed to find a goal versus number of runs, where after 49 runs we flipped the scent trail. This analysis shows that the rat finds a path using combined navigation after approximately 20 runs, on average. After learning, the rat switches to the navigation based on environmental cues, and we observe an upwards curve shift due to the exploration and exploitation strategy of the Q-learning. As expected, the rat's performance is not affected after scent marks were flipped since the rat prefers environmental cues after learning. Statistics for the second experiment are presented in Fig. 4.14 D where we can see that as soon as environmental cues are unavailable (i.e. removed) the rat follows the trail of scent marks which leads to the

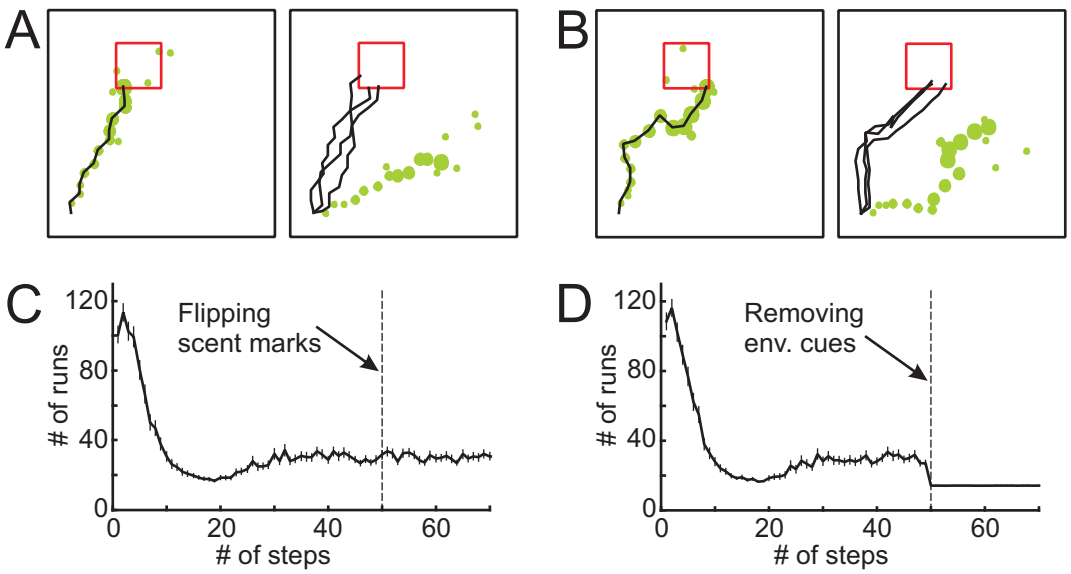

Figure 4.14: Navigation results when using hierarchical input preference. **A-C)** Navigation results when self-generated marks were flipped after run 49. **A, B)** Results of single experiments: self-generated marks and rat's trajectory at the end of learning (left) and flipped self-generated marks and rat's trajectories of three consecutive runs after scent marks were flipped. **C)** The average number of steps needed to find the goal is plotted versus the number of runs in 200 experiments. The vertical bars show the standard error mean (SEM). **D)** Navigation results when environmental cues were removed after run 49. The average number of steps together with SEM is plotted versus the number of runs in 200 experiments.

food source. Lack of exploration in this case leads to the noise free flat line after run 49. Our modified model captures similar properties of hierarchical input preference observed in animals (Maaswinkel and Whishaw, 1999). For further discussion and relation to biological data see section 4.9.2.

4.8 Remapping of place fields and goal navigation

It is known from the literature that place fields can change in firing rate, position, shape, or turn on/off when the animal is exposed to different environments, a phenomenon which is called remapping (Muller and Kubie, 1987; Wilson and McNaughton, 1993; Shapiro et al., 1997; Tanila et al., 1997; Knierim et al., 1995, 1998). Fundamental changes occur within 5-10 minutes of exploration in a new environment, whereas the firing rate can change even within the first second (Wilson and

McNaughton, 1993). Here we also investigate how remapping of place cells affects goal navigation task when the rat switches between different environments. We compare different navigation strategies with respect to change of environmental cues, as well as to a change of the goal location.

4.8.1 Experimental setup

To look at the remapping of place cells, we first let the rat explore randomly the whole environment "A" for 5000 time steps. Environment "A" contains visual and olfactory cues as shown in Fig. 4.15, as already used in the previously described experiments. Afterwards the rat is exposed to another environment, "B", for 5000 time steps (see panel A and B). In our model we use the same visual landmarks and the same odours for both environments "A" and "B". In order to change the environment we switch the landmarks and change the locations of odour sources. Landmarks are used by the rat in order to distinguish between the four walls and to estimate the distance to them. When we switch landmarks the rat gets different estimates of distances to the walls marked by the same landmark when being at the same position in the environment "A" and "B". The rat also gets different odour intensity at the same position in the environment "A" compared to the environment "B". After exploration in the environment "B" the rat was moved back to the familiar environment "A".

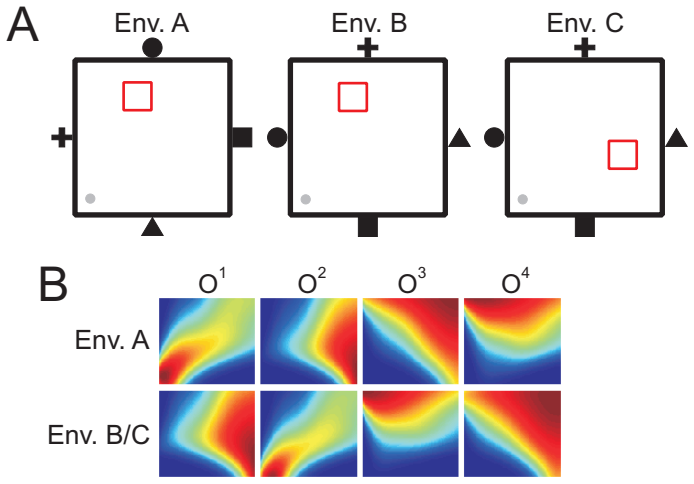

Figure 4.15: **A)** Images of different environmental setups. Landmarks are switched in the environment "B" as compared to the original environment "A" whereas in the environment "C" allothetic cues as well as the location of the goal are changed. **B)** Change of olfactory cues. The locations of odour sources are changed in the environment "B/C" as compared to the environment "A".

To compare Q-learning based on place fields obtained from combined visual and olfactory stimuli with the combination of Q-learning with the navigation based on self-generated odour marks we performed two different sets of experiments. In the first set of experiments, we switched between two environments "A" and "B", changing only environmental cues and keeping the location of the goal unchanged (see Fig. 4.15 A). In the second set of experiments, we switched between environment "A" and "C", and in "C" the environmental cues as well as the location of the food source were changed.

4.8.2 Place field remapping

The resulting place fields of a remapping experiment when switching between environments "A" and "B" are shown in Fig. 4.16, with the same selected 100 of total 500 place cells shown for each case. As expected, we can see that place fields of cells can change their firing rate, position, shape, or turn on/off. Note that there are also cells which do not change their properties in both environments. Place cells, as expected, display their original fields when returned to "A" (from environment "B" to "A").

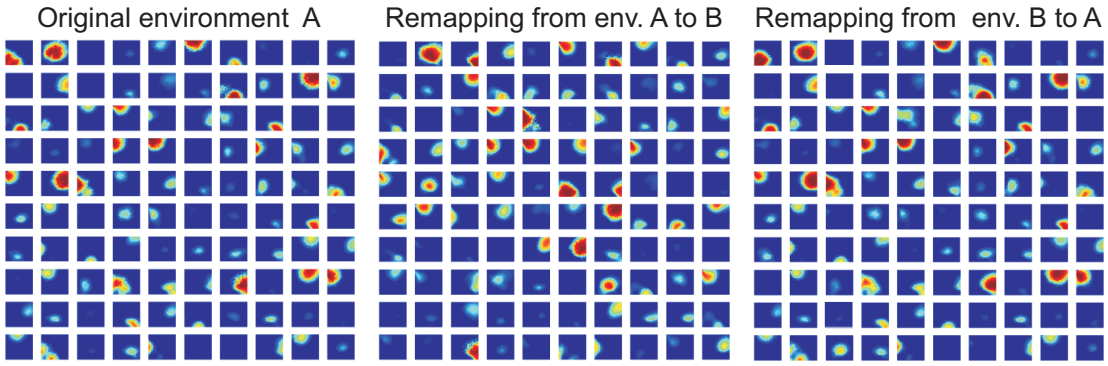

Figure 4.16: Remapping of place fields from environment "A" to "B" and from the environment "B" back to "A". The same selected cells (100 of total 500) are presented in all three cases.

The average distribution of change in maximal firing rates of PFs between environments "A" and "B" in 100 experiments is shown in Fig. 4.17 A. Note that we show change in firing rates of PFs only for cells with maximum firing rate $r > 0.5$, which are the cells that actually drive Q-learning. Positive values mean that cells increased firing rate or turned on when moving the rat from the environment "A" to "B" and vice versa. The distribution of changes in the positions of place fields (only with maximum firing rate $r > 0.5$) is presented in panel B, where we plot the average

distance between PFs centres (given by the location of the maximal firing in the PF) in environment "A" versus "B".

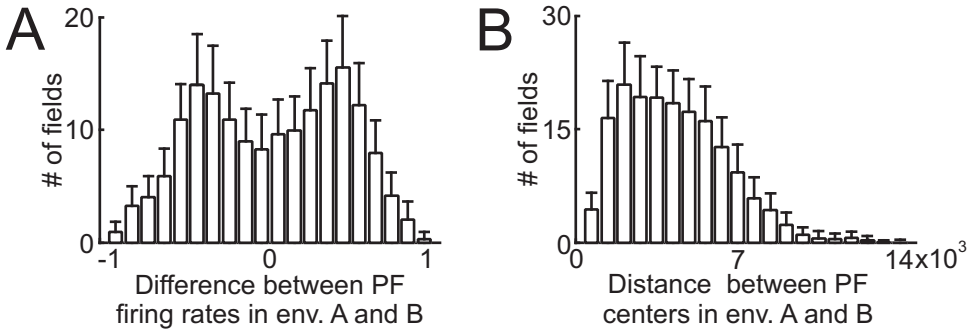

Figure 4.17: **A)** Average difference between maximum firing rate of place fields in environment "A" and "B" together with standard deviation (SD) are plotted for 100 experiments. -1 means that the cell stopped firing when switched to the other environment and +1 means that the cell was off in environment "A" but turned on when moved to environment "B". **B)** Average distance between centres of place fields in environment "A" and "B" together with SD are plotted for 100 experiments.

4.8.3 Influence of remapping on goal navigation

In the following subsection we present results on spatial navigation with respect to the remapping of place fields when switching between to different environments. For environmental setup see Fig. 4.15. The results of goal navigation while switching between environments "A" and "B" are shown in Fig. 4.18, where the average number of steps needed to find the food source is plotted versus number of runs for 200 experiments. Navigation results obtained by using Q-learning based on PCs obtained from visual and olfactory stimuli (VOQ) are presented in panel A, and results of the combined method ($VOQS$) are shown in panel B. Note that here we used a combined strategy without hierarchical input preference, i.e. the rat would still follow a scent trail after learning. We can see that by using both navigation strategies the rat can learn to find the goal in two environments "A" and "B", whenever the location of the food source is the same in both environments, and it goes directly to the goal after returning to the previous environment. It is worthwhile to note that in our model we do not introduce unfamiliar cues to the rat in the new environment, but we just "fool" the rat by switching visual cues and changing the position and shape of olfactory cues. That is why we also observe that the rat uses some information (i.e. learnt Q-values) from the previous environment, and it does not have to relearn from scratch when moved to the new environment. In panel A, for comparison, we show

the control case where in environment "A" and "B" we initialise Q-values randomly from a uniform distribution within the interval [0;1].

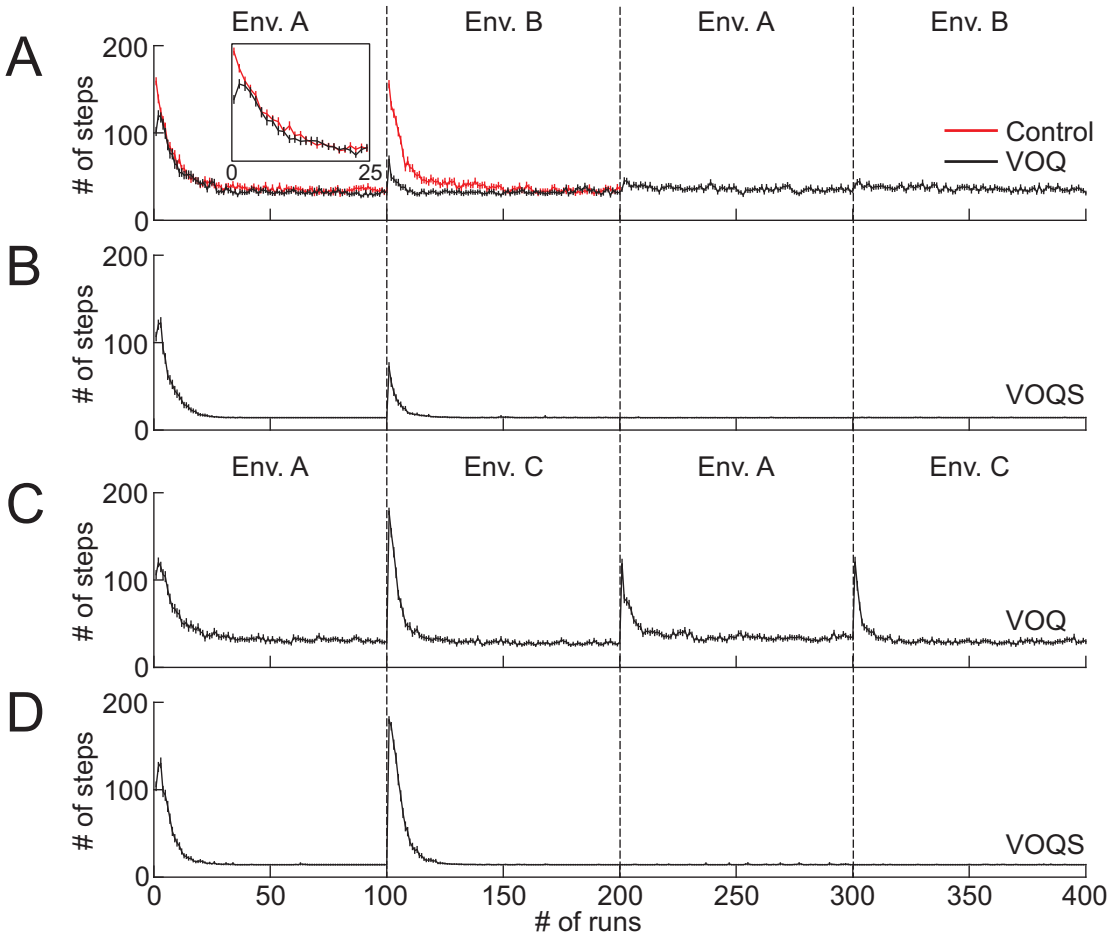

Figure 4.18: Comparison of goal navigation strategies with respect to different environmental setups: **A, B** - only environmental change, **C, D** - the environment an the location of the goal changed (see Fig. 4.15). The average number of steps needed to find the goal are plotted versus the number of runs in 200 experiments. The vertical bars show the standard error mean (SEM). Cases VOQ and $VOQS$ are as explained in Fig. 4.13 A. Control: the same as in case VOQ, but we start learning with random Q-values at the beginning in the environment "A" and "B" whereas in case VOQ we initialise weights with zero Q-values only at the very beginning and do not reset values while switching between the environments.

The results for goal navigation while switching between environments "A" and "C" (the location of the goal is also changed) for the cases VOQ and VOQS are

presented in Fig. 4.18 C and D respectively. Here we found that the rat has to relearn the food location all the time (panel A), even if returned to the previously visited environment. However, by employing the combined strategy (see panel D), the rat can easily find the food source in both environments even if the location of the goal is changed, because the rat can follow the trail of scent marks. Note that if we used the combined strategy with hierarchical input preference we would have obtained results similar to the case VOQ (see panel F), since after learning the rat would prefer environmental cues and navigate according Q-values.

In general, we observed that the rat can learn both environments when the location of the goal is unchanged but has to relearn the route in case of changes in both environmental cues and location of the goal. For further discussion on remapping results see section 4.9.3.

4.9 Discussion

In the following we compare our place cell model and goal navigation strategies with other approaches. We also discuss our results in relation to biological data.

4.9.1 Place cell model

We modelled place cells from visual and olfactory cues using a feed-forward network based on radial basis functions. Here we used an abstract model excluding interactions between hippocampal layers. This is justified as we did not focus on the place model itself but rather on the contribution of sensory inputs to the formation of place cells and on the utilisation of place cells in spatial navigation. Our approach is similar to the model of O'Keefe and Burgess (1996) or Hartley et al. (2000), but we use n-dimensional radial basis functions instead of calculating the thresholded sum of the Gaussian tuning-curves of the rat's distance from each box wall (O'Keefe and Burgess, 1996). Our model differs from the augmented model of Hartley et al. (2000), where the firing rate of a place cell is modelled as the thresholded sum of *boundary vector cells* (BVCs). The response of a BVC is the product of two Gaussian tuning curves, where one is a function of the distance from the rat to the wall and the second is a function of the rat's head direction (Hartley et al., 2000). In these models, the amplitude and the width of the place field depend on the distance to the wall: the larger the distance, the lower the amplitude and the broader the field, and vice versa. In our model we keep the width of the place field σ_f fixed and the obtained place fields that vary in shape and amplitude because of the combination of different sensory inputs. We use a winner-takes-all mechanism for place field formation, which means that we do not change weights of neighbour neurons as in self-organising map (SOM) approaches (Chokshi et al., 2003; Ollington and Vamplew, 2004) as there are no obvious topographical relations between the positions of the place fields and the

anatomical locations of the place cells relative to each other within the hippocampus (O'Keefe, 1999).

In several studies (Arleo and Gerstner, 2000; Arleo et al., 2004; Sheynikhovich et al., 2005; Strösslin et al., 2005) self-motion cues have been used as an additional input to hippocampus to create place cells. The disadvantage of self-motion cues is that path integration leads to an accumulation of errors in direction and distance and needs to be re-calibrated according to position estimation from stable cues (Etienne et al., 1996, 2004). Save et al. (2000) have shown that path integration alone is insufficient to maintain the stability of place fields. If visual or olfactory sensory cues are available then these cues dominate over path integration information (Maaswinkel and Whishaw, 1999; Whishaw et al., 2001). In contrast to other models we use odour cues as an additional input to form place cells. For the sake of simplicity we model static odours. Models of dynamic odours are quite complex and include many parameters (Boeker et al., 2000). By using static odours we ignore odour patch development, and effects that might be induced by changes of odours in time. Here we concentrate only on an odour function as a reference cue that is sensed unambiguously by the rat, as opposed to visual cues, which might be mismatched, misinterpreted or not seen at all. Obtained place fields capture properties similar to those that were found in the rat hippocampus (Muller and Kubie, 1987; Muller et al., 1994; Wilson and McNaughton, 1993; O'Keefe, 1999).

Place cells tend to be less directional when navigating in an open environment as compared to navigation where the rat is forced to move along a specific direction (McNaughton et al., 1983; Muller et al., 1994; Markus et al., 1995). These properties has been also captured by the models of Sharp (1991) and Brunel and Trullier (1998). Here we have investigated the contribution of olfactory input to the directionality of place cells. From our analysis, we found that if olfactory cues are available for the formation of place cells, more omnidirectional fields develop. This agrees with observations of place fields by Battaglia et al. (2004) on cue-rich and cue-poor linear tracks. The proportion of omnidirectional cells over total spatially selective cells was $\approx 43\%$ in a cue-rich environment vs. $\approx 30\%$ in a cue-poor environment. We obtained more omnidirectional cells because cells tend to be more directional in eight-arm mazes or T-mazes compared to open environments (Muller et al., 1994; Markus et al., 1995). Our results support the notion that place cell directionality should influence goal directed behaviour as we obtained better performance in a goal navigation task when using place cells formed from both visual and combined stimuli than when using place cells formed from visual cues alone.

4.9.2 Goal navigation learning

In the second part of this chapter we presented different navigation strategies and compared them in a goal navigation task and in a remapping situation. Goal navigation based on place cells has previously been addressed by implementing reinforcement

learning algorithms (Arleo and Gerstner, 2000; Arleo et al., 2004; Foster et al., 2000; Strösslin et al., 2005; Sheynikhovich et al., 2005; Krichmar et al., 2005). We presented a new navigation mechanism that combined Q-learning with navigation based on self-generated odour patches in order to achieve better performance in goal directed navigation. Our approach differs from that of Russell (1995), who developed a robotic system where the robot is able to lay an odour trail on the ground and to follow the trail afterwards. In his approach the robot is not using odour marking to find a goal, whereas in our approach, the rat lays scent marks in order to find a goal and to create a trail, which leads to the food source. The proposed mechanism, based on self-marking, propagates scent marks backwards from the location of the reward as in reinforcement learning, but here we do not have predefined features, but rather create them "on the fly", and we do not directly memorise action values associated to states. The mechanism of RBF[2]-like features created on-line in action learning was used in several other studies (Kretchmar and Anderson, 1997; Atkeson et al., 1997). The method of updating odour marks resembles a TD(0) approach with function approximation (Sutton and Barto, 1998), where the weights towards the value function are increased if the following states have high values. The update rule in our study is different from the one used in TD. Here, updates of odour marks are made by a fixed amount based on the binary decision whether some odour is sensed at the current location or not.

Experimental data show that rats perform better in cue-rich environments compared to the cue-poor environments. Barnes et al. (1980) showed that if all of the extra-maze cues surrounding a circular maze were removed, rats made many more errors finding a goal location. Morris (1984) demonstrated that rats performed worse when he obscured some of the cues around the water maze by pulling the curtains 1/4 of the way around. When he obscured all of the extra-maze cues by pulling the curtains fully around, the rats performed very badly. Prados and Trobalon (1998) showed that rats could learn the platform location in a water maze if 4 or 2 extra-maze cues were available, but they were much worse if only 1 cue was present. We addressed these findings by testing the performance of our model rat with and without olfactory input where we observed that the model rat performed significantly better with both, visual and olfactory, cues compared to visual stimuli alone.

The experiments of Maaswinkel and Whishaw (1999) suggest that rats have a hierarchical preference in using sensory cues. In their experiments, rats ignored distortion in self-motion cues when they where moved to a new starting position or ignored distortion in odour cues (scent marks) when the apparatus was rotated suggesting that visual cues dominate over other cues whenever they are available. However, when blindfolded, the rats still performed well suggesting that they were using odour cues when available, and path integration when odour cues were disrupted. To address these findings we modified our combined navigation strategy by adding an input pref-

[2]RBF - radial basis function

erence component where the rat uses both environmental and self generated cues for the learning. After learning the rat prefers environmental cues if they are available and uses self-generated olfactory cues when visual cues are not available. By using such an modified strategy, we have demonstrated that the model rat succeeds in faster goal directed learning showing unaffected performance when environmental cues are changed. This is supported by the finding that a rat can find a goal when the scent trail is distorted or removed, or can find the route to the goal using self-laid odour cues when environmental cues are unavailable.

4.9.3 Remapping and goal navigation

The results for goal navigation with respect to remapping of place cells show that the rat can learn to find a goal in two environments, "A" and "B", by using Q-learning or combined navigation when the location of the goal is unchanged, but environmental cues are switched. Note that the rat can learn both environments only as long as different, partially overlapping subsets of place cells fire in the environment "A" and "B", i.e. most of the cells, which do not fire in environment "A", fire in environment "B". In case of cue rotation the rat would need to relearn the task all the time if the location of the goal is not rotated together with landmarks, because in both environments the same subset of place cells would be used. This is an equivalent of leaving the environment the same, but changing the location of the goal. Also in the Morris water-maze experiment (Morris, 1981) the rat has to relearn the location of the platform every time whenever it is moved to another location. When environments are substantially different and the cells remap, in our experiments the rat can easily find the food source in both environments even if the location of the goal is changed by employing the combined strategy, because the rat can use the trail of scent marks.

Our model predicts that the remapping of place fields would disrupt a previously learnt route to a goal. The closest empirical data addressing this prediction is a study by Jeffery et al. (2003), who examined the relationship between remapping and performance of a spatial navigation task. In their experiment, rats were trained to search for a food source in a black box, and subsequently tested in a white box. Jeffery et al. (2003) found that place cells re-mapped between the two boxes, and although the rats were slightly worse in the second environment, they still performed well. This finding suggests that, although the place cells may encode spatial contexts, they dont directly guide behaviour. One difference between the experimental situation of Jeffery et al. (2003) and that of the current model is that in the experimental situation there were no landmarks within the square apparatus. Instead, rats relied on spatial landmarks - posters on the curtains surrounding the apparatus - for orientation. So, in the Jeffery et al. (2003) experiment, unlike in our model, cues outside the immediate environment were the only way in which the animal could distinguish the correct corner. The results of Yoganarasimha and Knierim (2005) suggest that head direction cells are influenced by distal landmarks, whereas some place cells are influenced by

local landmarks. Thus it may be that the Jeffery et al. (2003) task was one that could not be solved using place cells, because there was no way of distinguishing one corner of the apparatus from the other because there were no local cues available within the square. Rats may have used a non-place cell representation - such as the head direction cell system - to solve the task. Had there been local cues inside the square enclosure and no cues outside the enclosure, a stronger link between remapping and disrupted navigation may have been observed. An acknowledged difficulty with this account, however, is that Jeffery et al. (2003) also show that this task is impaired by lesions of the hippocampus.

4.9.4 Predictions and suggested experiments

Present experimental studies on spatial learning in cue-rich-cue-poor environments are still based on visual cues alone (Barnes et al., 1980; Morris, 1984; Prados and Trobalon, 1998). They also test the performance of the rat after learning. It would thus be interesting to test whether real animals would learn the task faster in environments with additional olfactory cues compared to visual stimuli alone as our model predicts.

Experiments on self-marking behaviour in the process of learning would be useful to prove or disprove the proposed setup and hypothesis that self-marking behaviour speeds-up learning.

In the Jeffery et al. (2003) experiment on place cell remapping and goal navigation, it may be that the task was one that could not be solved using place cells, because there was no way of distinguishing one corner of the apparatus from the other because there were no local cues available within the square. It would be interesting to make more experiments in order to test the hypothesis whether remapping of place cells influences goal directed learning or not as our model predicts.

By using a combined strategy with hierarchical input preference the model rat creates two representations of the route to the goal: one is based on environmental cues while the other is based on self-generated scent marks. Our model predicts that in case of remapping, when the goal in two environments is at different locations, the rat would fail when moved back to the previous environment since it would prefer environmental cues. We would hypothesise that the rat could use the scent trail in the next trial after it fails to find a goal when using environmental cues. Experiments to test this hypothesis would also be of great interest.

5 Conclusion and Outlook

Each previous chapter contained its own extensive "Discussion" section where we compared our methods to other approaches and related them to biological data. In this chapter we will, thus, only briefly summarise presented work by highlighting all main findings, provide an outlook for future investigations, and conclude this thesis.

In this thesis we were investigating the development and utility of receptive fields in closed-loop behavioural systems. In the first part of the thesis (Chapter 2) we developed visual receptive fields from uni-modal sensory cues (visual) by temporal sequence learning. Here we have for the first time implemented a simple layered structure and obtained stable behaviour in a closed-loop scenario. We showed that chained architectures can be employed in order to obtain better behavioural performance as compared to the simple architecture where learning fails because of weak correlations. While the two chained architectures are still rather simple, we believe that this is nevertheless an important step towards more advanced networks of correlation based learning units. Furthermore, we could for the first time generate and stabilise secondary receptive fields in a closed-loop context.

In section 2.9 we showed that many system parameters influence development of receptive fields (RFs) which as a consequence will alter the agent's behaviour. One crucial parameter is the position of the receptive field in a camera image. On shallow tracks faster learning and more accurate driving behaviour is obtained when the RF is positioned further away from the reflex sensor field whereas for sharp tracks better performance is achieved when the RF is closer to the reflex sensor. This could be improved by using different shapes of receptive fields (e.g. elongated RFs) in order to avoid tuning of RF position. Alternatively, we could use several smaller receptive fields and place them at different positions in the camera image. Here we could use a separate unit for each receptive field where all outputs of such units would contribute to the final motor output which then would be used to control the agent.

Another limitation of our approach presented in this chapter is that the system can not unlearn. If we train the robot on sharp tracks and then use the learned weights for the shallow tracks then the robot behaves inefficiently by over-steering and consuming too much energy (see Fig. 2.10 and Fig. 2.26). This problem could be solved by augmenting our learning system by a weight decay (forgetting), i.e. by decreasing

weights by a reasonable small value $\Delta\rho$ every time step or if the corresponding input is not triggered for some time τ. In addition we could use an asymmetric instead of the symmetric setup where left and right RFs would develop separately. This would be beneficial in asymmetrical maze tracks, e.g. track which mainly contains shallow leftward and sharp rightward turns.

The presented learning architectures for receptive field development could be also used for tasks such as obstacle avoidance, food retrieval or combinations of those. This would only require usage of several sets of RFs where each of them would be responsible for a specific subtask and would develop independently. It should be also possible to apply such receptive fields for visually guided reaching and grasping where RFs would generate velocity and/or acceleration profiles for each joint of a robot arm (Tamosiunaite et al., 2009). We could also use chained learning architectures with different receptive fields in more complex navigation tasks, e.g olfactory (would require a set of odour sensors in order to build receptive field) and visual RFs where odour would predict specific visual targets which would be followed by a trigger of a negative (hitting an obstacle) or positive (receiving energy) reflex.

In the second part (Chapter 3) we have analysed closed loop behavioural systems which change by differential Hebbian learning. We were surprised to find that even these very simple systems are already too complex to fully deduct the system's behaviour from the initial setup of system and world. Only together with some information on the general structure of the development of their descriptive parameters, analytical solutions can be still found for their temporal development. By using energy, input/output ratio and entropy measures and investigating their development during learning we have shown that within well-specified scenarios there are indeed agents which are optimal with respect to their structure and adaptive properties. As a consequence, this investigation may help leading to better understanding of the complex dynamics of learning&behaving systems. The fact that with learning optimal agents will exist (probably under any measure of optimality!) may make it necessary to reconsider evolutionary approaches as cited above (Klyubin et al., 2007, 2008; Prokopenko et al., 2006) in light of a different fitness function, which also takes the learning into account (Baldwin Effect, Baldwin, 1896; Hinton and Nowlan, 1987). Finally, by applying system measures to the receptive field analysis we were able to demonstrate that receptive fields optimize the agents' behaviour with respect to the given task by minimising energy consumptions required to perform the task and maximising accuracy of the performance.

Here we used input/output ratio in order to see how the influence of predictor and reflex on the system output changes over time. This measure could be also used to investigate the dynamics of systems with many different inputs (also without defining predictive and reflexive inputs and independent on system setup) in order to analise the contribution of different inputs to the performance during learning. For this, one

would need to calculate input/output ratio of each sensory input independently. In addition, measures like mutual information between system inputs could be included in the analysis in order to see how much information inputs contain about each other in order to detect and remove unwanted redundancies in the system. The mutual information between sensory inputs at time moment t and the inputs after some time, i.e. at the time moment $t+\tau$, should allow us to see whether these inputs can predict the upcoming sensory state, i.e. predicting the environment. An analysis of the system dynamics with multiple subtasks, e.g. obstacle avoidance (negative tropism) and food retrieval (positive tropism), or multiple agent systems, for investigation of cooperative behaviour, would be also of great interest.

Finally, in the third and the last part of this thesis (Chapter 4) we presented a place cell model where we modelled place fields from multi-modal sensory cues (visual and olfactory) by using a feed-forward network based on radial basis functions. This was motivated by the experimental data which show that olfactory cues play an important role for the stability of place fields (Markus et al., 1994; Save et al., 2000) and the navigation of rodents (Tomlinson and Johnston, 1991; Lavenex and Schenk, 1995, 1996, 1998; Wallace et al., 2002a, 2003). We have for the first time implemented an odour supported place cell model and applied it for goal navigation learning. Based on self-marking behaviour in rodents (Harley and Martin, 1999), we proposed a novel navigation mechanism which generates better performance in goal directed navigation. We predict that the use of environmental odour cues improves omni-directionality of place cells which, as a consequence, results in faster goal directed learning, whereas the use of self-generated scent marks results in even faster learning, and could serve as an additional information for path finding when environmental cues are not available.

We have demonstrated that additional sensory inputs improve spatial navigation. We believe that adding more sensory inputs (auditory, somatosensory, self-motion) would increase the performance even more. Including self-marks (scent marks) to place field formation would be an important upgrade of the model, too.

Tamosiunaite et al. (2008) have shown that number and size of place fields play quite an important role in spatial navigation with respect to speed and convergence of path learning. From place cell recordings it is known that place cells in ventral hippocampus fire with relatively large place fields (PFs) whereas dorsal PFs are smaller (Hasselmo, 2008; Kjelstrup et al., 2008). It is thought that bigger PFs are used for navigation on larger scales (e.g. associations with different rooms) whereas smaller PFs are used where higher precision is needed (e.g. to code for a specific location in the room). To take this into account we would need to augment our model with several layers of place fields with different PF sizes (see Fig. 5.1 B) in order to efficiently solve spatial tasks as shown in Fig. 5.1 A (for more details see figure legend). This could be done by implementing such learning mechanisms as adaptive tile cod-

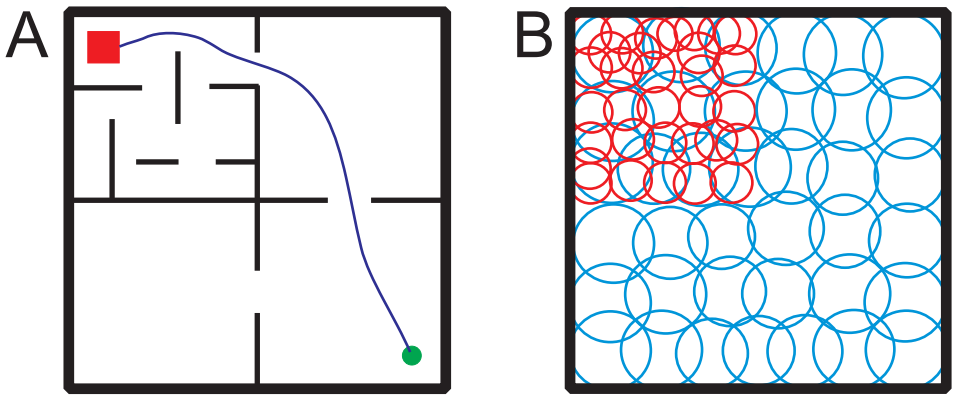

Figure 5.1: **A)** Depiction of an environment with three empty rooms and one maze room. Green point is the start position (home), red square is the target position (goal). The task of a rat is to find an optimal (shortest) path from home location to the goal as indicated by the blue line. **B)** Schematic diagram of configuration of place fields for the environment shown in A. Here larger place fields (blue circles) are used to differentiate between different rooms and smaller place fields are used for navigation inside the maze room.

ing (Whiteson et al., 2007) and hierarchical reinforcement learning (Botvinick et al., 2009).

We believe that the embedding of learning architectures into behaving systems, which close the loop between perception and action, is an important field of investigation leading away from the pure stimulus-response paradigm to a more ecological system's perspective. Our results suggest that it seems to be possible to achieve stable structural and functional development also in more complex architectures by rigorously embedding the learning neuronal system in its environment. While our scenarios are still relatively simple we think that this work may help leading to a better understanding of dynamics and behaviour of more complex closed-loop learning systems. We also stress the advantage of multi-modal sensory cues to receptive field development and behavioural performance which not only makes suggestions for new biological experiments but also for new approaches in autonomous behaving systems.

Bibliography

Agostini, A. Celaya, E. (2004). Trajectory tracking control of a rotational joint using feature-based categorization learning. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan. IEEE.

Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83(3):287–299.

Arleo, A., Smeraldi, F., and Gerstner, W. (2004). Cognitive navigation based on nonuniform Gabor space sampling, unsupervised growing networks, and reinforcement learning. *IEEE Transactions on Neural Networks*, 15(3):639–652.

Ashby, W. R. (1956). *An Introduction to Cybernetics*. Chapmann and Hall Ltd.

Atkeson, C. G., Moore, A. W., and Schaal, S. (1997). Locally weighted learning for control. *Artificial Intelligence Review*, 11(1):75–113.

Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63:329–339.

Bailey, C. H., Giustetto, M., Huang, Y. Y., Hawkins, R. D., and Kandel, E. R. (2000). Is heterosynaptic modulation essential for stabilizing Hebbian plasticity and memory? *Nat Rev Neurosci*, 1(1):11–20.

Balakrishnan, K., Bousquet, O., and Honavar, V. (1999). Spatial learning and localization in animals: A computational model and its implications for mobile robots. *Adaptive Behavior*, 7(2):137–216.

Baldwin, J. M. (1896). A new factor in evolution. *American Naturalist*, 30:441–451.

Barnes, C., Nadel, L., and Honig, W. (1980). Spatial memory deficit in senescent rats. *Canadian Journal of Psychology*, 34:29–39.

Barry, C., Hayman, R., Burgess, N., and Jeffery, K. J. (2007). Experience-dependent rescaling of entorhinal grids. *Nature Neuroscience*, 10(6):682–684.

Barto, A. (1995). Reinforcement learning in motor control. In Arbib, M., editor, *Handbook of Brain Theory and Neural Networks*, pages 809–812. MIT Press.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 13, pages 835–846.

Battaglia, F. P., Sutherland, G. R., and McNaughton, B. L. (2004). Local sensory cues and place cell directionality: additional evidence of prospective coding in the hippocampus. *The Journal of Neuroscience*, 24(19):4541–4550.

Bell, A. J. and Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Res*, 37(23):3327–3338.

Blais, B. S., Intrator, N., Shouval, H., and Cooper, L. N. (1998). Receptive field formation in natural scene environments: Comparison of single cell learning rules. In *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Boeker, P., Wallenfang, O., Koster, F., Croce, R., Diekmann, B., Griebel, M., and Schulze-Lammers, P. (2000). The modelling of odour dispersion with time-resolved models. *Agrartechnische Forschung*, 4:E84–E89.

Botvinick, M. M., Niv, Y., and Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition*, 113:262–280.

Bousquet, O., Balakrishnan, K., and Honavar, V. (1998). Is the hippocampus a kalman filter? In *Proceedings of the Pacific Symposium on Biocomputing*, pages 655–666.

Box, G., Jenkins, G. M., and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*. Prentice-Hall.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press.

Brown, M. A. and Sharp, P. E. (1995). Simulation of spatial learning in the Morris water maze by a Neural Networksork model of the hippocampal formation and nucleus accumbens. *Hippocampus*, 5(3):171–188.

Brunel, N. and Trullier, O. (1998). Plasticity of directional place fields in a model of rodent CA3. *Hippocampus*, 8:651–665.

Calton, J., Stackman, R., Goodridge, J., Archey, W., Dudchenko, P., Taube, J., and Oman, C. (2003). Hippocampal place cell instability after lesions of the head direction cell network. *The Journal of Neuroscience*, 23:9719–9731.

Carvell, G. E. and Simons, D. J. (1990). Biometric analyses of vibrissal tactile discrimination in the rat. *The Journal of Neuroscience*, 10(8):2638–2648.

Chokshi, K., Wermter, S., and Weber, C. (2003). Learning localisation based on landmarks using self-organisation. In *ICANN*, pages 504–514.

Collett, T. S., Cartwright, B. A., and Smith, B. A. (1986). Landmark learning and visuo-spatial memories in gerbils. *Journal of Comparative Physiology*, 158(6):835–851.

Daugman, J. G. (1989). Entropy reduction and decorrelation in visual coding by oriented neural receptive fields. *IEEE Trans. on Biomedical Engineering*, 36(1):107–114.

Der, R., Güttler, F., and Ay, N. (2008). Predictive information and emergent cooperativity in a chain of mobile robots. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 166–172. MIT Press, Cambridge, MA.

Dragoi, V., Sharma, J., and Suri, M. (2003). Response Plasticity in Primary Visual Cortex and its Role in Vision and Visuomotor Behaviour: Bottom-up and Top-down Influences. *IETE Journal of Research*, 49(2):123–132.

Dudchenko, P. A. (2001). How do animals actually solve the T maze? *Behavioral Neuroscience*, 115:850–860.

Eichenbaum, H., Dudchenko, P., Wood, E., Shapiro, M., and Tanila, H. (1999). The hippocampus, memory, and place cells: is it spatial memory or a memory space? *Neuron*, 23(2):209–226.

Eilam, D. and Golani, I. (1989). Home base behavior of rats (rattus norvegicus) exploring a novel environment. *Behavioural Brain Research*, 34:199–211.

Einhäuser, W., Kayser, C., König, P., and Körding, K. P. (2002). Learning the invariance properties of complex cells from their responses to natural stimuli. *Eur J Neurosci*, 15(3):475–486.

Ekstrom, A. D., Kahana, M. J., Caplan, J. B., Fields, T. A., Isham, E. A., Newman, E. L., and Fried, I. (2003). Cellular networks underlying human spatial navigation. *Nature*, 425(6954):184–188.

Etienne, A. S. and Jeffery, K. J. (2004). Path integration in mammals. *Hippocampus*, 14(2):180–192.

Etienne, A. S., Maurer, R., Boulens, V., Levy, A., and Rowe, T. (2004). Resetting the path integrator: a basic condition for route-based navigation. *The Journal of Experimental Biology*, 207(Pt 9):1491–1508. Comparative Study.

Etienne, A. S., Maurer, R., and Seguinot, V. (1996). Path integration in mammals and its interaction with visual landmarks. *The Journal of Experimental Biology*, 199(Pt 1):201–209.

Felsen, G., Touryan, J., Han, F., and Dan, Y. (2005). Cortical sensitivity to visual features in natural scenes. *PLoS Biol*, 3(10).

Foster, D. J., Morris, R. G., and Dayan, P. (2000). A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16.

Franzius, M., Vollgraf, R., and Wiskott, L. (2007). From grids to places. *Journal of Computational Neuroscience*, 22(3):297–299.

Gaussier, P., Revel, A., Banquet, J. P., and Babeau, V. (2002). From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biological Cybernetics*, 86(1):15–28.

Gewirtz, J. C. and Davis, M. (2000). Using pavlovian higher-order conditioning paradigms to investigate the neural substrates of emotional learning and memory. *Learn. Mem.*, 7(5):257–266.

Gomi, H. and Kawato, M. (1993). Neural network control for a closed-loop system using feedback-error-learning. *Neural Netw.*, 6(7):933–946.

Hafting, T., Fyhn, M., Molden, S., Moser, M. B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436:801–806.

Harley, C. W. and Martin, G. M. (1999). Open field motor patterns and object marking, but not object sniffing, are altered by ibotenate lesions of the hippocampus. *Neurobiology of Learning and Memory*, 72(3):202–214.

Hartley, T., Burgess, N., Lever, C., Cacucci, F., and O'Keefe, J. (2000). Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus*, 10(4):369–379.

Hasselmo, M. E. (2008). Neuroscience. The scale of experience. *Science*, 321:46–47.

Hill, A. J. and Best, P. J. (1981). Effects of deafness and blindness on the spatial correlates of hippocampal unit activity in the rat. *Experimental Neurology*, 74(1):204–217.

Hines, D. J. and Whishaw, I. Q. (2005). Home bases formed to visual cues but not to self-movement (dead reckoning) cues in exploring hippocampectomized rats. *European Journal of Neuroscience*, 22:2363–2375.

Hinton, G. E. and Nowlan, S. J. (1987). How learning guides evolution. *Complex System*, 1:495–502.

Hölscher, C. (2003). Time, space and hippocampal functions. *Reviews in the Neurosciences*, 14(3):253–284.

Humeau, Y., Shaban, H., Bissiere, S., and Luthi, A. (2003). Presynaptic induction of heterosynaptic associative plasticity in the mammalian brain. *Nature*, 426(6968):841–845.

Hurri, J. and Hyvärinen, A. (2003). Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 15(3):663–691.

Iglesias, R., Nehmzow, U., and Billings, S. A. (2008). Model identification and model analysis in robot training. *Robotics and Autonomuos Systems*, 56:1061–1067.

Ikeda, H., Akiyama, G., Fujii, Y., Minowa, R., Koshikawa, N., and Cools, A. (2003). Role of AMPA and NMDA receptors in the nucleus accumbens shell in turning behaviour of rats: interaction with dopamine and receptors. *Neuropharmacology*, 44:8187.

Jara, E., Vila, J., and Maldonado, A. (2006). Second-order conditioning of human causal learning. *Learning and Motivation*, 37:230–246.

Jay, T. (2003). Dopamine: a potential substrate for synaptic plasticity and memory mechanisms. *Prog Neurobiol*, 69(6):375390.

Jeffery, K., Gilbert, A., Burton, S., and Strudwick, A. (2003). Preserved performance in a hippocampal-dependent spatial task despite complete place cell remapping. *Hippocampus*, 13:175–189.

Jeffery, K. and O'Keefe, J. (1999). Learned interaction of visual and idiothetic cues in the control of place field orientation. *Experimental Brain Research*, 127:151–161.

Jodogne, S., Scalzo, F., and Piater, J. H. (2005). Task-driven learning of spatial combinations of visual features. In *Proc. of the IEEE Workshop on Learning in Computer Vision and Pattern Recognition*, San Diego (CA, USA). IEEE.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2000). *Principles of Neural Science*. McGraw-Hill, New York.

Kelley, A. E. (1999). Functional specificity of ventral striatal compartments in appetitive behaviors. *Ann N Y Acad Sci*, 877:71–90.

Kjelstrup, K. B., Solstad, T., Brun, V. H., Hafting, T., Leutgeb, S., Witter, M. P., Moser, E. I., and Moser, M. B. (2008). Finite scale of spatial representation in the hippocampus. *Science*, 321:140–143.

Klopf, A. H. (1988). A neuronal model of classical conditioning. *Psychobiol.*, 16(2):85–123.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2004). Organization of the information flow in the perception-action loop of evolved agents. In *2004 NASA/DoD Conference on Evolvable Hardware. IEEE Computer Society*, pages 177–180.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: A universal agent-centric measure of control. In *IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 128–135.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2007). Representations of space and time in the maximization of information flow in the perception-action loop. *Neural Comput*, 19:2387–2432.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2008). Keep your options open: an information-based driving principle for sensorimotor systems. *PLoS ONE*, 3:e4018.

Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L. (1995). Place cells, head direction cells, and the learning of landmark stability. *The Journal of Neuroscience*, 15(3 Pt 1):1648–1659.

Knierim, J. J., Kudrimoti, H. S., and McNoughton, B. L. (1998). Interaction between idiothetic cues and external landmarks in the control of place cells and head direction cells. *Journal of Neurophysiology*, 80:425–446.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, third, extended edition.

Kolodziejski, C., Porr, B., and Wörgötter, F. (2009). On the asymptotic equivalence between differential Hebbian and temporal difference learning. *Neural Computation*, 21:1173–1202.

Körding, K., Kayser, C., Einhäuser, W., and König, P. (2004). How are complex cell properties adapted to the statistics of natural stimuli? *J Neurophysiol*, 91:206212.

Kosco, B. (1986). Differential Hebbian learning. In Denker, J. S., editor, *Neural networks for computing: AIP Conference Proc. proceedings*, volume 151. New York: American Institute of Physics.

Kretchmar, R. and Anderson, C. (1997). Comparison of cmacs and radial basis functions for local function approximators in reinforcement learning. In *Proceedings of the IEEE International Conference on Neural Networksorks*, pages 834–837, Houston, TX.

Krichmar, J. L., Seth, A. K., Nitz, D. A., Fleischer, J. G., and Edelman, G. M. (2005). Spatial navigation and causal analysis in a brain-based device modeling cortical-hippocampal interactions. *Neuroinformatics*, 3(3):197–221. Comparative Study.

Kulvicius, T., Porr, B., and Wörgötter, F. (2007). Chained learning architectures in a simple closed-loop behavioural context. *Biol Cybern*, 97:363–378.

Kyriacou, T., Nehmzow, U., Iglesias, R., and Billings, S. A. (2008). Accurate robot simulation through system identification. *Robotics and Autonomuos Systems*, 56:1082–1093.

Lavenex, P. and Schenk, F. (1995). Influence of local environmental olfactory cues on place learning in rats. *Physiology & Behavior*, 58(6):1059–1066.

Lavenex, P. and Schenk, F. (1996). Integration of olfactory information in a spatial representation enabling accurate arm choice in the radial arm maze. *Learning & Memory*, 2(6):299–319.

Lavenex, P. and Schenk, F. (1998). Olfactory traces and spatial learning in rats. *Animal Behaviour*, 56(5):1129–1136.

Lungarella, M., Pegors, T., Bulwinkle, D., and Sporns, O. (2005). Methods for quantifying the informational structure of sensory and motor data. *Neuroinformatics*, 3:243–262.

Lungarella, M. and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Comput. Biol.*, 2:e144.

Maaswinkel, H. and Whishaw, I. Q. (1999). Homing with locale, taxon, and dead reckoning strategies by foraging rats: sensory hierarchy in spatial navigation. *Behavioural Brain Research*, 99:143–152.

Manoonpong, P., Geng, T., Kulvicius, T., Porr, B., and Wörgötter, F. (2007). Adaptive, Fast Walking in a Biped Robot under Neuronal Control and Learning. *PLoS Computational Biology*, 3(7):e134 doi:10.1371/journal.pcbi.0030134.

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215.

Markus, E., Barnes, C., McNaughton, B., Gladden, V., and Skaggs, W. (1994). Spatial information content and reliability of hippocampal CA1 neurons: effects of visual input. *Hippocampus*, 4:410–421.

Markus, E. J., Qin, Y. L., Leonard, B., Skaggs, W. E., McNaughton, B. L., and Barnes, C. A. (1995). Interactions between location and task affect the spatial and directional firing of hippocampal neurons. *The Journal of Neuroscience*, 15(11):7079–7094.

McClelland, J. L., Rumelhart, D. E., and Hinton, G. E. (1987). *Parallel Distributed Processing - Vol. 1*. MIT Press.

McFarland, D. J. (1971). *Feedback Mechanisms in Animal Behaviour*. Academic Press, London.

McKinstry, J. L., Edelman, G. M., and Krichmar, J. L. (2006). A cerebellar model for predictive motor control tested in a brain-based device. *Proc Natl Acad Sci U S A*, 103(9):3387–3392.

McNaughton, B., Battaglia, F., Jensen, O., Moser, E., and Moser, M. (2006). Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7:663–678.

McNaughton, B. L., Barnes, C. A., and O'Keefe, J. (1983). Plasticity of directional place fields in a model of rodent CA3. *Experimental Brain Research*, 52(1):41–49.

Montague, P. R., Dayan, P., Person, C., and Sejnowski, T. J. (1995). Bee foraging in uncertain environments using predictive hebbian learning. *Nature*, 377:725–728.

Morris, R. (1981). Spatial localization does not require the presence of local cues. *Learning and Motivation*, 12:239–260.

Morris, R. (1984). Developments of a water-maze procedure for studying spatial learning in the rat. *The Journal of Neuroscience Methods*, 11(1):47–60.

Muller, R. U., Bostock, E., Taube, J. S., and Kubie, J. L. (1994). On the directional firing properties of hippocampal place cells. *The Journal of Neuroscience*, 14(12):7235–7251.

Muller, R. U. and Kubie, J. L. (1987). The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *The Journal of Neuroscience*, 7(7):1951–1968.

Muller, R. U., Ranck, J. B., and Taube, J. S. (1996). Head direction cells: properties and functional significance. *Current Opinion in Neurobiology*, 6(2):196–206.

Nakanishi, J. and Schaal, S. (2004). Feedback error learning and nonlinear adaptive control. *Neural Networks*, 17:1453–1465.

Nemati, F. and Whishaw, I. Q. (2007). The point of entry contributes to the organization of exploratory behaviour of rats on an open field: An example of spontaneous episodic memory. *Behavioural Brain Research*, 182:119–128.

Niv, Y., Joel, D., Meilijson, I., and Ruppin, E. (2002). Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior*, 10(1):5–24.

O'Keefe, J. (1999). Do hippocampal pyramidal cells signal non-spatial as well as spatial information? *Hippocampus*, 9:352–365.

O'Keefe, J. and Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175.

O'Keefe, J. and Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Oxford University Press.

O'Keefe, J. and Speakman, A. (1987). Single unit activity in the rat hippocampus during a spatial memory task. *Experimental Brain Research*, 68(1):1–27.

Ollington, R. and Vamplew, P. (2004). Learning place cells from sonar data. In *AISAT2004: International Conference on Artificial Intelligence in Science and Technology*, pages 126–131.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Pomerleau, D. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97.

Pomerleau, D. (1995). Neural network vision for robot driving. In Arbib, M., editor, *The Handbook of Brain Theory and Neural Networks*.

Porr, B., Egerton, A., and Wörgötter, F. (2006). Towards closed loop information: Predictive information. *Constructivist Foundations*, 1(2):83–90.

Porr, B. and Wörgötter, F. (2003a). Isotropic sequence order learning. *Neural Comp.*, 15:831–864.

Porr, B. and Wörgötter, F. (2003b). Isotropic sequence order learning in a closed loop behavioural system. *Roy. Soc. Phil. Trans. Math., Phys. & Eng. Sci.*, 361(1811):2225–2244.

Porr, B. and Wörgötter, F. (2006). Strongly improved stability and faster convergence of temporal sequence learning by utilising input correlations only. *Neural Comp.*, 18(6):1380–1412.

Poupart, P. and Boutilier, C. (2002). Value-directed compression of POMDPs. In *Advances in Neural Information Processing Systems*, volume 15, pages 1547–1554.

Prados, J. and Trobalon, J. (1998). The location of an invisible goal requires the presence of at least two landmarks. *Psychobiology*, 26:42–48.

Prokopenko, M., Gerasimov, V., and Tanev, I. (2006). Evolving spatiotemporal coordination in a modular robotic system. In *SAB 2006*, pages 558–569.

Recce, M. and Harris, K. D. (1996). Memory for places: a navigational model in support of Marr's theory of hippocampal function. *Hippocampus*, 6(6):735–748. Comparative Study.

Rescorla, R. A. (1980). *Pavlovian Second-Order Conditioning: Studies in Associative Learning.* Erlbaum, Hillsdale.

Reynolds, S. I. (2002). The stability of general discounted reinforcement learning with linear function approximation. In *UK Workshop on Computational Intelligence (UKCI-02)*, pages 139–146, Birmingham, UK.

Russell, R. A. (1995). Laying and sensing odor markings as a strategy for assisting mobilerobot navigation tasks. *Robotics & Automation Magazine, IEEE*, 2(3):3–9.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B., Witter, M. P., Moser, E. I., and Moser, M. B. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312:758–762.

Saudargiene, A., Porr, B., and Wörgötter, F. (2004). How the shape of pre- and postsynaptic signals can influence STDP: a biophysical model. *Neural Comput*, 16:595–625.

Saudargiene, A., Porr, B., and Wörgötter, F. (2005). Synaptic modifications depend on synapse location and activity: a biophysical model of STDP. *BioSystems*, 79:3–10.

Save, E., Cressant, A., Thinus-Blanc, C., and Poucet, B. (1998). Spatial firing of hippocampal place cells in blind rats. *J. Neurosci.*, 18:1818–1826.

Save, E., Nerad, L., and Poucet, B. (2000). Contribution of multiple sensory information to place field stability in hippocampal place cells. *Hippocampus*, 10(1):64–76.

Schultz, W. and Suri, R. E. (2001). Temporal difference model reproduces anticipatory neural activity. *Neural Comp.*, 13(4):841–862.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Shapiro, M. L. and Hetherington, P. A. (1993). A simple network model simulates hippocampal place fields : parametric analyses and physiological predictions. *Behavioral Neuroscience*, 107:34–50.

Shapiro, M. L., Tanila, H., and Eichenbaum, H. (1997). Cues that hippocampal place cells encode: dynamic and hierarchical representation of local distal stimuli. *Hippocampus*, 7:624–642.

Sharp, P. E. (1991). Computer simulation of hippocampal place cells. *Psychobiology*, 19(2):103–115.

Sheynikhovich, D., Chavarriaga, R., Strösslin, T., and Gerstner, W. (2005). Spatial Representation and Navigation in a Bio-inspired Robot. In *Biomimetic Neural Learning for Intelligent Robots: Intelligent Systems, Cognitive Robotics, and Neuroscience*, pages 245–264.

Simoncelli, E. P. and Olshausen, B. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216.

Slonim, N., Somerville, R., Tishby, N., and Lahav, O. (2001). Objective classification of galaxy spectra using the information bottleneck method. *Monthly Notes of the Royal Astronomical Society*, 323:270–284.

Slonim, N. and Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *Proc. of the 23rd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*.

Slonim, N. and Tishby, N. (2001). The power of word clustering for text classification. In *Proc. of the 23rd European Colloquium on Information Retrieval Research*.

Strösslin, T., Sheynikhovich, D., Chavarriaga, R., and Gerstner, W. (2005). Robust self-localisation and navigation based on hippocampal place cells. *Neural Networks*, 18(9):1125–1140.

Sugita, Y. (1996). Global plasticity in adult visual cortex following reversal of visual input. *Nature*, 380(6574):523–526.

Suri, R. E. and Schultz, W. (1998). Learning of sequential movements by neural network model with dopamine-like reinforcement signal. *Exp. Brain Res.*, 121:350–354.

Sutton, R. and Barto, A. (1981). Towards a modern theory of adaptive networks: Expectation and prediction. *Psychol. Review*, 88:135–170.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3:9–44.

Sutton, R. S. and Barto, A. G. (1990). Time-derivative models of Pavlovian reinforcement. In Gabriel, M. and Moore, J., editors, *Learning and computational neuroscience:Foundation of adaptive networks*. MIT Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Bradford Books, MIT Press, Cambridge, MA, 2002 edition.

Takács, B. and Lőrincz, A. (2006). Independent component analysis forms place cells in realistic robot simulations. *Neurocomputing*, 69:1249–1252.

Tamosiunaite, M., Ainge, J., Kulvicius, T., Porr, B., Dudchenko, P., and Wörgötter, F. (2008). Path-finding in real and simulated rats: assessing the influence of path characteristics on navigation learning. *J Comput Neurosci*, 25:562–582.

Tamosiunaite, M., Asfour, T., and Wörgötter, F. (2009). Learning to reach by reinforcement learning using a receptive field based function approximation approach with continuous actions. *Biol Cybern*, 100:249–260.

Tanila, H., Shapiro, M. L., and Eichenbaum, H. (1997). Discordance of spatial representations in ensembles of hippocampal place cells. *Hippocampus*, 7:613–623.

Taube, J. S., Muller, R. U., and Ranck, J. B. J. (1990a). Head direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *The Journal of Neuroscience*, 10:420–435.

Taube, J. S., Muller, R. U., and Ranck, J. B. J. (1990b). Head direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations. *The Journal of Neuroscience*, 10:436–447.

Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377.

Tomlinson, W. T. and Johnston, T. D. (1991). Hamsters remember spatial information derived from olfactory cues. *Animal Learning and Behavior*, 19:185–190.

Touchette, H. and Lloyd, S. (2000). Information-theoretic approach to the study of control systems. *Physica A*, 331:140–172.

Touretzky, D. S. and Redish, A. D. (1996). Theory of rodent navigation based on interacting representations of space. *Hippocampus*, 6(3):247–270.

Tsukamoto, M., Yasui, T., Yamada, M. K., Nishiyama, N., Matsuki, N., and Ikegaya, Y. (2003). Mossy fibre synaptic NMDA receptors trigger non-Hebbian long-term potentiation at entorhino-CA3 synapses in the rat. *J Physiol*, 546(3):665–675.

Verschure, P. and Althaus, P. (2003). A real-world rational agent: unifying old and new AI. *Cognitive Science*, 27:561–590.

Verschure, P. and Coolen, A. (1991). Adaptive fields: Distributed representations of classically conditioned associations. *Network*, 2:189–206.

Vinje, W. E. and Gallant, J. L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–6.

Wallace, D. G., Gorny, B., and Whishaw, I. Q. (2002a). Rats can track odors, other rats, and themselves: implications for the study of spatial behavior. *Behavioural Brain Research*, 131(1-2):185–192.

Wallace, D. G., Hines, D. J., and Whishaw, I. Q. (2002b). Quantification of a single exploratory trip reveals hippocampal formation mediated dead reckoning. *The Journal of Neuroscience Methods*, 113:131–145.

Wallace, D. G., Kolb, B., and Whishaw, I. Q. (2003). Odor tracking in rats with orbital frontal lesions. *Behavioral Neuroscience*, 117(3):616–620.

Walter, W. G. (1950). An imitation of life. *Scient. Am.*, 182:42–45.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge, Cambridge, England.

Watkins, C. J. C. H. and Dayan, P. (1992). Technical note:Q-Learning. *Mach. Learn.*, 8:279–292.

Webb, B. (2002). Robots in invertebrate neuroscience. *Nature*, 417:359–363.

Weber, C. and Obermayer, K. (1999). Orientation selective cells emerge in a sparsely coding Boltzmann machine. In *Artificial Neural Networks - ICANN 99*, pages 286–291.

Whishaw, I. Q., Hines, D. J., and Wallace, D. G. (2001). Dead reckoning (path integration) requires the hippocampal formation: evidence from spontaneous exploration and spatial learning tasks in light (allothetic) and dark (idiothetic) tests. *Behavioural Brain Research*, 127(1-2):49–69.

Whiteson, S., Taylor, M. E., and Stone, P. (2007). Adaptive tile coding for value function approximation. Technical Report AI-TR-07-339, University of Texas at Austin.

Wiener, N. (1961). *Cybernetics — or control and communication in the animal and the machine*. The M.I.T. Press, Cambridge, Massachusetts, 2 edition.

Wilson, M. A. and McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space. *Science*, 261(5124):1055–1058.

Witten, I. H. (1977). An adaptive optimal controller for discrete-time Markov environments. *Information and Control*, 34:86–295.

Wörgötter, F. and Porr, B. (2005). Temporal sequence learning for prediction and control - a review of different models and their relation to biological mechanisms. *Neural Comp.*, 17:245–319.

Wyss, R., König, P., and Verschure, P. F. (2006). A Model of the Ventral Visual System Based on Temporal Stability and Local Memory. *PLoS Biol.*, 4(5):e120.

Yoganarasimha, D. and Knierim, J. (2005). Coupling between place cells and head direction cells during relative translations and rotations of distal landmarks. *Experimental Brain Research*, 160:344–359.

A Appendix

A.1 Pattern inconsistency measure

For the structure evaluation and ordering of the receptive fields (see Fig. 2.15) we used pattern inconsistency measure PI which is defined as the average distance to the neighbouring weights computed over all weights within the receptive field (similar to the grey scale values used in self-organising maps to measure the similarity across neighbours, Kohonen, 2001):

$$PI = \frac{1}{N^2} \sum_{i,j} \frac{1}{m} \sum_{k,l} |\rho_{i,j} - \rho_{k,l}|, \tag{A.1}$$

where $i, j = 1 \ldots N, N = 15$ is the size of the receptive field, $k = i - 1 \ldots i + 1$, $l = j - 1 \ldots j + 1$, and m is the number of neighbouring units. Indexes k and l define the neighbouring weights $\rho_{k,l}$ of the weight $\rho_{i,j}$ in the receptive field, and we used eight adjacent neighbours ($m = 8$) as shown in Fig. A.1 A. Note that at the corners of RF we have only tree neighbouring units ($m = 3$) and at the borders we have five neighbouring units ($m = 5$). We also normalised receptive field weights ($\rho_{i,j}$) between zero and one before calculating PI. $PI = 0$ if receptive field is homogeneous ($\rho_{i,j} = const.$) and $PI \approx 0.33$ if the receptive field has random structure with weights from a uniform distribution (see Fig. A.1 B).

A.2 Input intensity map

Values of the input intensity maps (Fig. 2.18 A) were calculated by the following equation:

$$IM_{i,j} = \frac{\sum_t f(x^L_{1_{i,j}}(t)) + \sum_t f(x^L_{1_{i,j}}(t))}{\max_{i,j} IM_{i,j}}, \quad with \tag{A.2}$$

$$f(x) = \begin{cases} 1 & \text{if } |x| > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Here $i, j = 1 \ldots 15$ denote the indices of the receptive field pixels (inputs) and are ordered as shown in Fig. 2.13 A, $t = 1 \ldots N$ is the time measured in steps. Note, since we have a symmetrical setup, here we sum number of inputs from corresponding

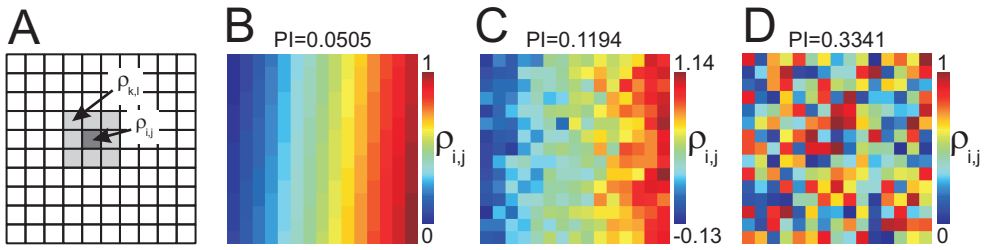

Figure A.1: **A)** Diagram shows neighbouring weights $\rho_{k,l}$ (marked by light grey) of the weight $\rho_{i,j}$ (dark grey) in the receptive field. **B-D)** Examples of receptive fields with different structure. **B)** Gradient-like receptive fields with values from a uniform distribution. **C)** The same as in panel B only here we added Gaussian noise with the mean $\mu = 0$ and the standard deviation $\sigma = 0.1$. **D)** Random receptive field with values from a uniform distribution. Value at the top of the RF represents the pattern inconsistency measure.

pixels from both the left and the right RF and obtain only one intensity map (shown as the right intensity map).

A.3 Robot's deviation from the track

Robot's deviation from the track is simply defined as the average deviation of the robot's position (defined by the mass centre of the robot) from the track obtained from the robot's driving trajectory and is calculated by Euclidian distance:

$$\Psi = \sum_{t=0}^{N-1} \sqrt{(x_r(t) - x_t(t))^2 + (y_r(t) - y_t(t))^2}\ units, \tag{A.3}$$

where $x_r(t)$ and $y_r(t)$ are the coordinates of the robot's position at time moment t, $x_t(t)$ and $y_t(t)$ are the coordinates of the track point from which the distance to the robot's position is minimal, and $t = 0 \ldots N-1$ denote the driving duration and is measured in steps.

A.4 Contrast measure

We obtained values for the colour-coding in Fig. 3.2 by the following equations:

$$Z_1(k) = \sum_{t=k}^{k+c_w-1} |\omega_1(t) \cdot x_1(t)|,$$

$$Z_0(k) = \sum_{t=k}^{k+c_w-1} |\omega_0(t) \cdot x_0(t)|,$$

$$R(k) = \frac{Z_1(k) - Z_0(k)}{Z_1(k) + Z_0(k)}, \tag{A.4}$$

where $k = 0 \ldots N - w_r$, $N = 10^5$ is the length of the input sequence, and $c_w = 5 \times 10^3$ is the size of the sliding time window. Note that we normalised values of R between zero and one.

A.5 Analytical calculation of the temporal development

Here we provide with the analytical calculation of the temporal development including the temporal dependence of τ on k. From equations 3.22 and 3.26 we derive three second order differential equations, which we solve independently:

if $0 \leq k \leq k_p$

$$\omega'(k) = \mu \frac{2\, a_0\, A_1\, \sigma_0}{\sigma_1^2} \left(\tau_b + (\tau_p - \tau_b)\, \frac{k}{k_p}\right) \left(1 - \frac{\omega(k)}{\omega_f}\right), \tag{A.5}$$

if $k_p < k \leq k_b$

$$\omega'(k) = \mu \frac{2\, a_0\, A_1\, \sigma_0}{\sigma_1^2} \left(\tau_p - \frac{\tau_p - \tau_b}{k_p - k_b}\, k_p + \frac{\tau_p - \tau_b}{k_p - k_b}\, k\right) \left(1 - \frac{\omega(k)}{\omega_f}\right), \tag{A.6}$$

if $k > k_b$

$$\omega'(k) = \mu \frac{2\, a_0\, A_1\, \sigma_0}{\sigma_1^2} \left(\tau_b\, \frac{k}{k_b}\right) \left(1 - \frac{\omega(k)}{\omega_f}\right). \tag{A.7}$$

The solution of these differential equations are as follows:

if $k \leq k_p$

$$\omega(k) = \omega_f \left(1 - \exp\left[-\mu\, \tilde{\lambda} \frac{2\, k_p\, \tau_b + k(\tau_p - \tau_b)}{2\, k_p}\, k\right]\right), \tag{A.8}$$

if $k_p < k \leq k_b$

$$\omega(k) = \omega_f \left(1 - \exp\left[-\mu\, \tilde{\lambda} \frac{(k^2 - 2\, k_p\, k + k_p\, k_b)\, \tau_b - (k^2 - 2\, k\, k_b + k_p\, k_b)\, \tau_p}{2\, (k_b - k_p)}\right]\right), \tag{A.9}$$

if $k > k_b$

$$\omega(k) = \omega_f \left(1 - \exp\left[-\mu\, \tilde{\lambda} \frac{2\, k\, \tau_b + k_b(\tau_b - \tau_b)}{2}\right]\right), \tag{A.10}$$

where $\tilde{\lambda} = \lambda/\tau$ (see equation 3.25).

B

Curriculum Vitae

TOMAS KULVICIUS

Research Assistant at the Bernstein Center for Computational Neuroscience
Georg-August-Universität Göttingen
III Physikalisches Institut - Biophysik
Friedrich-Hund Platz 1
37077 Gttingen

Date and place of birth:	24 December 1978 Kaunas, Lithuania
Citizenship:	Lithuanian
E-mail:	tomas@physik3.gwdg.de
Tel.:	+49(0) 551 3910 762

EDUCATION

2005 Jul – 2010 May	PhD Student at the Department of Computer Science* Georg-August-Universität Göttingen Germany
2004 Nov – 2005 Jun	PhD Student at the Department of Computational Neuroscience University of Stirling Scotland, UK
2001 Sep – 2003 Jun	M.Sc. in Computer Science Vytautas Magnus University Kaunas, Lithuania
1997 Sep – 2001 Jun	B.Sc. in Computer Science Vytautas Magnus University Kaunas, Lithuania

* Continuation of studies started in Stirling University

AWARDS

- 2001 Premium Award for the Student Research Work in Computer Science.

- 2002/2003 Carol Martin Gruodis memorial scholarship.

RESEARCH INTERESTS

Closed loop behavioural systems
Learning algorithms
Receptive Fields
Robotics
Biosignal analysis
Biological system modelling

LIST OF PUBLICATIONS

Journal Papers

- Kulvicius, T., Kolodziejski, C., Tamosiunaite, M., Porr, B. and Wörgötter, F. Behavioral analysis of differential hebbian learning in closed-loop systems. *Biological Cybernetics*, accepted for publication.

- Kulvicius, T., Tamosiunaite, M., Ainge, J., Dudchenko, P. and Wörgötter, F. (2008). Odor supported place cell model and goal navigation in rodents. *Journal of Computational Neuroscience*, 25(3), 481-500.

- Tamosiunaite, M., Ainge, J., Kulvicius, T., Porr, B., Dudchenko, P. and Wörgötter, F. (2008). Path-finding in real and simulated rats: Assessing the influence of path characteristics on navigation learning. *Journal of Computational Neuroscience*, 25(3), 562-582.

- Kulvicius, T., Porr, B. and Wörgötter, F. (2007). Chained learning architectures in a simple closed-loop behavioural context. *Biological Cybernetics*, 97(5), 363-378.

- Manoonpong, P., Geng, T., Porr, B., Kulvicius, T. and Wörgötter, F. (2007). Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Computational Biology*, 3(7), e134 doi:10.1371/journal.pcbi.0030134.

- Kulvicius, T., Porr, B. and Wörgötter, F. (2007). Development of receptive fields in a closed-loop behavioural system. *Neurocomputing*, 70(10-12), 2046-2049.

- Porr, B., Kulvicius, T. and Wörgötter, F. (2007). Improved stability and convergence with three factor learning. *Neurocomputing*, 70(10-12), 2005-2008.

- Kulvicius T., Tamosiunate, M. and Vaisnys, R. (2005). T Wave Alternans Features for Automated Detection. *Informatica*, 16(4), 587-602.

Book Chapters

- Markelic, I., Kulvicius, T., Tamosiunaite, M., and Wörgötter, F. (2009). Anticipatory Driving for a Robot-Car Based on Supervised Learning. *Lecture Notes in Computer Science: Anticipatory Behavior in Adaptive Learning Systems*, 267-282.

Conference Papers

- Abramov A., Kulvicius T., Wörgötter F. and Dellen B. (2010). Real-time image segmentation on a GPU. Accepted for publication in *Facing the Multicore-Challenge Conference*.

- Tamosiunaite M., Vaisnys R., Kulvicius T., Urbonaviciene G., Kaminskiene S. and Bluzaite I. (2002). Search for T Wave Alternans by Multiplicative Factor Method. In Analysis of Biomedical Signals and Images: Proceedings of 16-th International EURASIP Conference on Biosignal Analysis, 26-28 June 2002, Brno, Czech Republic, pp. 103-105.

- Tamosiunaite M., Vaisnys R., Kulvicius T., Urbonaviciene G., Kaminskiene S. and Bluzaite I. (2002). Search for T Wave Alternans by Multiplicative Factor Method. In Analysis of Biomedical Signals and Images: Proceedings of 16-th International EURASIP Conference on Biosignal Analysis, 26-28 June 2002, Brno, Czech Republic, pp. 103-105.

- Tamosiunaite M., Vaisnys R., Jogminas D., Kulvicius T., Urbonaviciene G., Plisiene J. and Bluzas J. (2001). Problems of T Wave Alternans Detection in Rest ECG. In Biomedical Engineering: Proceedings of the International Conference on Biomedical Engineering, 18-19 October, 2001 Kaunas, Lithuania, pp. 11-14.