Learning a world model and planning with a self-organizing, dynamic neural system

Marc Toussaint
Institut für Neuroinformatik
Ruhr-Universität Bochum, ND 04
44780 Bochum—Germany
mt@neuroinformatik.rub.de

Abstract

We present a connectionist architecture that can learn a model of the relations between perceptions and actions and use this model for behavior planning. State representations are learned with a growing self-organizing layer which is directly coupled to a perception and a motor layer. Knowledge about possible state transitions is encoded in the lateral connectivity. Motor signals modulate this lateral connectivity and a dynamic field on the layer organizes a planning process. All mechanisms are local and adaptation is based on Hebbian ideas. The model is continuous in the action, perception, and time domain.

1 Introduction

Planning of behavior requires some knowledge about the consequences of actions in a given environment. A *world model* captures such knowledge. There is clear evidence that nervous systems use such internal models to perform predictive motor control, imagery, inference, and planning in a way that involves a simulation of actions and their perceptual implications [1, 2]. However, the level of abstraction, the representation, on which such simulation occurs is hardly the level of physical coordinates. A tempting hypothesis is that the representations the brain uses for reasoning and planning are particularly designed (by adaptation or evolution) for *just* this purpose. To address such ideas we first need a basic model for how a connectionist architecture can encode a world model and how self-organization of inherent representations is possible.

In the field of machine learning, world models are a standard approach to handle behavior organization problems (for a comparison of model-based approaches to the classical, model-free Reinforcement Learning see, e.g., [3]). The basic idea of using neural networks to model the environment was given in [4, 5]. Our approach for a *connectionist world model* (CWM) is functionally similar to existing Machine Learning approaches with self-organizing state space models [6, 7]. It is able to grow neural representations for different world states and to learn the implications of actions in terms of state transitions. It differs though from classical approaches in some crucial points:

- The model is continuous in the action, the perception, as well as the time domain.

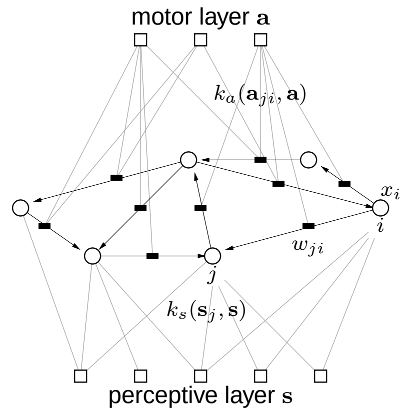

Figure 1: Schema of the CWM architecture.

- All mechanisms are based on local interactions. The adaptation mechanisms are largely derived from the idea of Hebbian plasticity. E.g., the lateral connectivity, which encodes knowledge about possible state transitions, is adapted by a variant of the temporal Hebb rule and allows local adaptation of the world model to local world changes.
- The coupling to the motor system is fully integrated in the architecture via a mechanism incorporating modulating synapses (comparable to shunting mechanisms).
- The two dynamic processes on the CWM, the "tracking" process estimating the current state and the planning process (similar to Dynamic Programming), will be realized by activation dynamics on the architecture, incorporating in particular lateral interactions, inspired by neural fields [8].

The outline of the paper is as follows: In the next section we describe our architecture, the dynamics of activation and the couplings to perception and motor layers. In section 3 we introduce a dynamic process that generates, as an attractor, a *value field* over the layer which is comparable to a state value function estimating the expected future return and allows for goal-oriented behavior organization. The self-organization process and adaptation mechanisms are described in section 4. We demonstrate the features of the model on a maze problem in section 5 and finally discuss the results and the model in general terms.

2 The model

The core of the *connectionist world model* (CWM) is a neural layer which is coupled to a perceptual layer and a motor layer, see figure 1. Let us enumerate the units of the central layer by $i = 1, .., N$. Lateral connections within the layer may exist and we denote a connection from the i-th to j-th unit by (ji). E.g., "$\sum_{(ji)}$" means "summing over all existing connections (ji)". To every unit we associate an activation $x_j \in \mathbb{R}$ which is governed by the dynamics

$$\tau_x \, \dot{x}_j = -x_j + k_s(\mathbf{s}_j, \mathbf{s}) + \eta \sum_{(ji)} k_a(\mathbf{a}_{ji}, \mathbf{a}) \, w_{ji} \, x_i \; , \tag{1}$$

which we will explain in detail in the following. First of all, x_i are the time-dependent activations and the dot-notation $\tau_x \, \dot{x} = F(x)$ means a time derivative which we algorithmically implemented by a Euler integration step $x(t) = x(t-1) + \frac{1}{\tau_x} F(x(t-1))$.

The first term in (1) induces an exponential relaxation while the second and third terms are the inputs. $k_s(\mathbf{s}_j, \mathbf{s})$ is the forward excitation that unit j receives from the perceptive layer.

Here, $\mathbf{s}_j$ is the codebook vector (receptive field) of unit j onto the perception layer which is compared to the current stimulus $\mathbf{s}$ via the kernel function k_s. We will choose Gaussian kernels as it is the case, e.g., for typical Radial Basis function networks.

The third term, $\sum_{(ji)} k_a(\mathbf{a}_{ji}, \mathbf{a})\, w_{ji}\, x_i$, describes the lateral interaction on the central layer. Namely, unit j receives lateral input from unit i iff there exists a connection (ji) from i to j. This lateral input is weighted by the connection's synaptic strength w_{ji}. Additionally there is another term entering *multiplicatively* into this lateral interaction: *Lateral inputs are modulated depending on the current motor activation.* We chose a modulation of the following kind: To every existing connection (ji) we associate a codebook vector $\mathbf{a}_{ji}$ onto the motor layer which is compared to the current motor activity $\mathbf{a}$ via a Gaussian kernel function k_a. Due to the multiplicative coupling, a connection contributes to lateral inputs only when the current motor activity "matches" the codebook vector of this connection. The modulation of information transmission by multiplicative or divisive interactions is a fundamental principle in biological neural systems [9]. One example is shunting inhibition where inhibitory synapses attach to regions of the dentritic tree near to the soma and thereby modulate the transmission of the dentritic input [10]. In our architecture, a shunting synapse, receiving input from the motor layer, might attach to only one branch of a (lateral) dentritic tree and thereby multiplicatively modulate the lateral inputs summed up at this subtree.

For the following it is helpful if we briefly discuss a certain relation between equation (1) and a classical probabilistic approach. Let us assume normalized kernel functions

$$k_s(\mathbf{s}_j, \mathbf{s}) = \frac{1}{\sqrt{2\pi}\,\sigma_s} \exp \frac{-(\mathbf{s}_j - \mathbf{s})^2}{2\sigma_s^2} \;, \quad k_a(\mathbf{a}_{ji}, \mathbf{a}) = \frac{1}{\sqrt{2\pi}\,\sigma_a} \exp \frac{-(\mathbf{a}_{ji} - \mathbf{a})^2}{2\sigma_a^2} \;.$$

These kernel functions can directly be interpreted as probabilities: $k_s(\mathbf{s}_j, \mathbf{s})$ represents the probability $P(\mathbf{s}|j)$ that the stimulus is $\mathbf{s}$ if j is active, and $k_a(\mathbf{a}_{ji}, \mathbf{a})$ the probability $P(\mathbf{a}|j, i)$ that the action is $\mathbf{a}$ if a transition $i \to j$ occurred. As for typical hidden Markov models we may derive the *prior* probability distribution $P(j|\mathbf{a})$, given the action:

$$P(j|\mathbf{a}, i) = \frac{P(\mathbf{a}|j, i)\; P(j|i)}{P(\mathbf{a}|i)} = k_a(\mathbf{a}_{ji}, \mathbf{a})\; \frac{P(j|i)}{P(\mathbf{a}|i)} \;,$$

$$P(j|\mathbf{a}) = \sum_i k_a(\mathbf{a}_{ji}, \mathbf{a})\; \frac{P(j|i)}{P(\mathbf{a}|i)}\; P(i) \;.$$

$P(\mathbf{a}|i)$ can be computed by normalizing $P(\mathbf{a}|j, i)\; P(j|i)$ over j such that $\sum_j P(j|\mathbf{a}, i) = 1$. What we would like to point out here is that in equation (1), the lateral input $\sum_{(ji)} k_a(\mathbf{a}_{ji}, \mathbf{a})\; w_{ji}\; x_i$ can be compared to the prior $P(j|\mathbf{a})$ under the assumption that x_i is proportional to $P(i)$ and if we have an adaptation mechanism for w_{ji} which converges to a value proportional to $P(j|i)$ and which also ensures normalization, i.e., $\sum_j k_a(\mathbf{a}_{ji}, \mathbf{a})\; w_{ji} = 1$ for all i and $\mathbf{a}$. This insight will help to judge some details of the next two section. The probabilistic interpretation can be further exploited, e.g., comparing the input of a unit j (or, in the quasi-stationary case, x_j itself) to the *posterior* and deriving theoretically grounded adaptation mechanisms. But this is not within the scope of this paper.

3 The dynamics of planning

To organize goal-oriented behavior we assume that, in parallel to the activation dynamics (1), there exists a second dynamic process which can be motivated from classical approaches to Reinforcement Learning [11, 12]. Recall the *Bellman equation*

$$V_\pi^*(i) = \sum_a \pi(a|i) \sum_j P(j|i, a) \Big[r(j) + \gamma\, V_\pi^*(j) \Big] \;, \tag{2}$$

yielded by the expectation $V^*(i)$ of the discounted future return $R(t) = \sum_{\tau=1}^{\infty} \gamma^{\tau-1}\, \varrho(t{+}\tau)$, which yields $R(t) = \varrho(t{+}1) + \gamma\, R(t{+}1)$, when situated in state i. Here, γ is the discount factor and we presumed that the received rewards $\varrho(t)$ actually depend only on the state and thus enter equation (2) only in terms of the reward function $r(i)$ (we neglect here that rewards may directly depend on the action). Behavior is described by a stochastic policy $\pi(a|i)$, the probability of executing action a in state i. Knowing the property (2) of V^* it is straight-forward to define a recursion algorithm for an approximation V of V^* such that V converges to V^*. This recursion algorithm is called *Value Iteration* and reads

$$\tau_v\, \Delta V_\pi(i) = -V_\pi(i) + \sum_a \pi(a|i) \sum_j P(j|i,a) \big[r(j) + \gamma\, V_\pi(j) \big]\ , \tag{3}$$

with a "reciprocal learning rate" or time constant τ_v. Note that (2) is the fixed point equation of (3).

The practical meaning of the state-value function V is that it quantifies how desirable and promising it is to reach a state i, also accounting for future rewards to be expected. In particular, if one knows the current state i it is a simple and efficient rule of behavior to choose that action a that will lead to the neighbor state j with maximal $V(j)$ (the greedy policy). In that sense, $V(i)$ provides a smooth gradient towards desirable goals. Note though that direct Value Iteration presumes that the state and action spaces are known and finite, and that the current state and the world model $P(j|i,a)$ is known.

How can we transfer these classical ideas to our model? We suppose that the CWM is given a goal stimulus $\mathbf{g}$ from outside, i.e., it is given the command to reach a world state that corresponds to the stimulus $\mathbf{g}$. This stimulus induces a *reward excitation* $r_i = k_s(\mathbf{s}_i, \mathbf{g})$ for each unit i. Now, besides the activations x_i, we introduce another field over the CWM, the *value field* v_i, which is in analogy to the state-value function $V(i)$. The dynamics is

$$\tau_v\, \dot v_i = -\, v_i + r_i + \gamma\ \max_{(ji)} (w_{ji}\, v_j)\ , \tag{4}$$

and well comparable to (3): One difference is that v_i estimates the "current-plus-future" reward $\varrho(t) + \gamma R(t)$ rather than the future reward only—in the upper notation this corresponds to the value iteration $\tau_v\ \Delta V_\pi(i) = -V_\pi(i) + r(i) + \sum_a \pi(a|i) \sum_j P(j|i,a) \big[\gamma\ V_\pi(j) \big]$. As it is commonly done for Value Iteration, we assumed π to be the greedy policy. More precisely, we considered only that action (i.e., that connection (ji)) that leads to the neighbor state j with maximal value $w_{ji}\, v_j$. In effect, the summations over a as well as over j can be replaced by a maximization over (ji). Finally we replaced the probability factor $P(j|i,a)$ by w_{ji}—we will see in the next section how w_{ji} is learned and what it will converge to.

In practice, the value field will relax quickly to its fixed point $v_i^* = r_i + \gamma\ \max_{(ji)}(w_{ji}\, v_j^*)$ and stay there if the goal does not change and if the world model is not re-adapted (see the experiments). The quasi-stationary value field v_i together with the current (typically non-stationary) activations x_i allow the system to generate a motor signal that guides towards the goal. More precisely, the value field v_i determines for every unit i the "best" neighbor unit $k_i = \mathrm{argmax}_j\ w_{ji}\, v_j$. The output motor signal is then the activation average

$$\mathbf{a} = \sum_i x_i\, \mathbf{a}_{k_i i} \tag{5}$$

of the motor codebook vectors $\mathbf{a}_{k_i i}$ that have been learned for the corresponding connections. Hence, the information flow between the central layer and the motor system is in both ways: In the "tracking" process as given by equation (1) the information flows from the motor layer to the central layer: Motor signals activate the corresponding connections and cause lateral, predictive excitations. In the action selection process as given by equation (5) the signals flow from the central layer back to the motor layer to induce the motor activity that should turn predictions into reality.

Depending on the specific problem and the representation of motor commands on the motor layer, a post-processing of the motor signal $\mathbf{a}$, e.g. a competition between contradictory motor units, might be necessary. In our experiments we will have two motor units and will always normalize the 2D vector $\mathbf{a}$ to unit length.

4 Self-organization and adaptation

The self-organization process of the central layer combines techniques from standard self-organizing maps [13, 14] and their extensions w.r.t. growing representations [15, 16] and the learning of temporal dependencies in lateral connections [17, 18]. The free variables of a CWM subject to adaptation are (1) the number of neurons and the lateral connectivity itself, (2) the codebook vectors $\mathbf{s}_i$ and $\mathbf{a}_{ji}$ to the perceptive and motor layers, respectively, and (3) the weights w_{ji} of the lateral connections. The adaptation mechanisms we propose are based on three general principles: (1) the addition of units for representation of novel states (*novelty*), (2) the fine tuning of the codebook vectors of units and connections (*plasticity*), and (3) the adaptation of lateral connections in favor of better prediction performance (*prediction*).

Novelty. Mechanisms similar to those of FuzzyARTMAPs [15] or Growing Neural Gas [16] account for the insertion of new units when novelty is detected. We detect novelty in a straight-forward manner, namely when the difference between the actual perception and the best matching unit becomes too large. To make this detection more robust, we use a low-pass filter (leaky integrator). At a given time, let z be the best matching unit, $z = \mathrm{argmax}_i\, x_i$. For this unit we integrate the error measure e_z

$$\tau_e\, \dot{e}_z = -e_z + (1 - k_s(\mathbf{s}_z, \mathbf{s}))\ .$$

We normalize $k_s(\mathbf{s}_z, \mathbf{s})$ such that it equals 1 in the perfect matching case when $\mathbf{s}_z = \mathbf{s}$. Whenever this error measure exceeds a threshold called *vigilance*, $e_z > \nu$, $\nu \in [0, 1]$, we generate a new unit j with the codebook vector equal to the current perception, $\mathbf{s}_j = \mathbf{s}$, and a connection from the last best matching unit $z^\dagger$ with the codebook vector equal to the current motor signal, $\mathbf{a}_{jz^\dagger} = \mathbf{a}$. The errors of both, the new and the old unit, are reset to zero, $e_z \leftarrow 0$, $e_j = 0$.

Plasticity. We use simple Hebbian plasticity to fine tune the representations of existing units and connections. Over time, the receptive fields of units and connections become more and more similar to the average stimuli that activated them. We use the update rules

$$\tau_s\, \dot{\mathbf{s}}_z = -\mathbf{s}_z + \mathbf{s}\ , \quad \tau_a\, \dot{\mathbf{a}}_{zz^\dagger} = -\mathbf{a}_{zz^\dagger} + \mathbf{a}\ ,$$

with learning time constants τ_s and τ_a.

Prediction and a temporal Hebb rule. Although perfect prediction is not the actual objective of the CWM, the predictive power is a measure of the correctness of the learned world model and good predictive power is one-to-one with good behavior planning. The first and simple mechanism to adapt the predictive power is to grow a new lateral connection between two successive best matching units $z^\dagger$ and z if it does not yet exist. The new connection is initialized with $w_{zz^\dagger} = 1$ and $\mathbf{a}_{zz^\dagger} = \mathbf{a}$. The second, more interesting mechanism addresses the adaptation of w_{ji} based on new experiences and can be motivated as follows: The temporal Hebb rule strengthens a synapse if the pre- and post-synaptic neurons spike in sequence, depending on the inter-spike-interval, and is supposed to roughly describe LTP and LTD (see, e.g.,[19]). In a population code model, this corresponds to a measure of correlation between the pre-synaptic and the delayed post-synaptic activity. In our case we additionally have to account for the action-dependence of a lateral connection.

We do so by considering the term $k_a(\mathbf{a}_{ji}, \mathbf{a})\ x_i$ instead of only the pre-synaptic activity. As a measure of temporal correlation we choose to relate this term to the *derivative* $\dot{x}_j$ of the post-synaptic unit instead of its delayed activation—this saves us from specifying an ad-hoc "typical" delay and directly reflects that, in equation (1), lateral inputs relate to the derivative of x_j. Hence, we consider the product $\dot{x}_j\ k_a(\mathbf{a}_{ji}, \mathbf{a})\ x_i$ as the measure of correlation. Our concrete implementation is a robust version of this idea:

$$\tau_w\ \dot{w}_{ji} = \kappa_{ji}\ [c_{ji} - w_{ji}\ \kappa_{ji}]\ , \quad \text{where}$$
$$\tau_\kappa\ \dot{c}_{ji} = -c_{ji} + \dot{x}_j\ k_a(\mathbf{a}_{ji}, \mathbf{a})\ x_i\ , \quad \tau_\kappa\ \dot{\kappa}_{ji} = -\kappa_{ji} + k_a(\mathbf{a}_{ji}, \mathbf{a})\ x_i\ .$$

Here, c_{ji} and κ_{ji} are simply low-pass filters of $\dot{x}_j\ k_a(\mathbf{a}_{ji}, \mathbf{a})\ x_i$ and of $k_a(\mathbf{a}_{ji}, \mathbf{a})\ x_i$. The term $w_{ji}\ \kappa_{ji}$ ensures convergence (assuming quasi static c_{ji} and κ_{ji}) of w_{ji} towards c_{ji}/κ_{ji}. The time scale of adaptation is modulated by the recent activity κ_{ji} of the connection.

5 Experiments

To demonstrate the functionality of the CWM we consider a simple maze problem. The parameters we used are

τ_x	η	$2\,\sigma_s^2$	$2\,\sigma_a^2$	τ_v	γ	τ_e	τ_s	τ_a	τ_w	τ_κ
2	0.1	0.01	0.5	2	0.8	10	20	5	10	100

Figure 2a displays the geometry of the maze. The "agent" is allowed to move continuously in this maze. The motor signal is 2-dimensional and encodes the forces $\mathbf{f}$ in x- and y-directions; the agent has a momentum and friction according to $\ddot{\mathbf{x}} = 0.2\,(\mathbf{f} - \dot{\mathbf{x}})$. As a stimulus, the CWM is given the 2D position $\mathbf{x}$.

Figure 2a also displays the (lateral) topology of the central layer after 30 000 time steps of self-organization, after which the system becomes quasi-stationary. The model is learned from scratch, initialized with one random unit. During this first phase, behavior planning is switched off and the maze is explored with a random walk that changes its direction only with probability 0.1 at a time. In the illustration, the positions of the units correspond to the codebook vectors that have been learned. The directedness and the codebook vectors of the connections can not displayed.

After the self-organization phase we switched on behavior planning. A goal stimulus corresponding to a random position in the maze is given and changed every time the agent reaches the goal. Generally, the agent has no problem finding a path to the goal. Figure 2b already displays a more interesting example. The agent has reached goal A and now seeks for goal B. However, we blocked the trespass 1. Starting at A the agent moves normally until it reaches the blockade. It stays there and moves slowly up an down in front of the blockade for a while—this while is of the order of the low-pass filter time scale τ_κ. During this time, the lateral weights of the connections pointing to the left are depressed and after about 150 time steps, this change of weights has enough influence on the value field dynamics (4) to let the agent chose the way around the bottom to goal B. Figure 2c displays the next scene: Starting at B, the agent tries to reach goal C again via the blockade 1 (the previous adaptation depressed only the connections from right to left). Again, it reaches the blockade, stays there for a while, and then takes the way around to goal C. Figures 2d and 2e repeat this experiment with blockade 2. Starting at D, the agent reaches the blockade 2 and eventually chooses the way around to goal E. Then, seeking for goal F, the agent reaches the blockade first from the left, thereafter from the bottom, then from the right, then it tries from the bottom again, and finally learned that none of these paths are valid anymore and chooses the way all around to goal F. Figures 2f shows that, once the world model has re-adapted to account for these blockades, the agent will not forget about them: Here, moving from G to H, it does not try to trespass block 2.

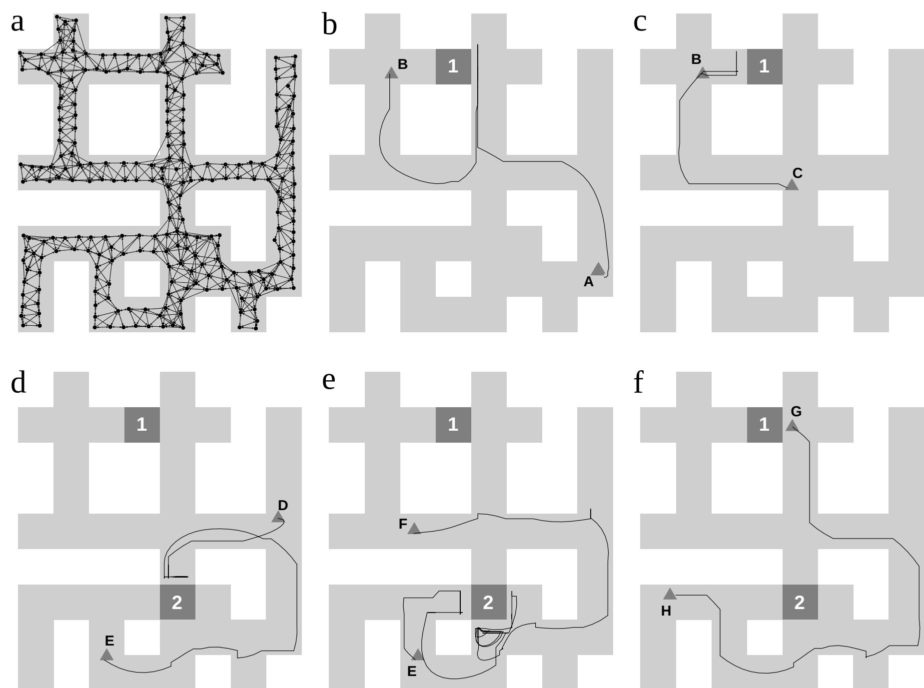

Figure 2: The CWM on a maze problem: (a) the outcome of self-organization; (b-c) agent movements from goal A to B to C, here, the trespass 1 was blocked and requires readaptation of the world model; (d-f) agent movements that demonstrate adaptation to a second blockade. Please see the text for more explanations.

The reader is encouraged to also refer to the movies of these experiments, deposited at www.marc-toussaint.net/03-cwm/, which visualize much better the dynamics of self-organization, the planning behavior, the dynamics of the value field, and the world model readaptation.

6 Discussion

The goal of this research is an understanding of how neural systems may learn and represent a world model that allows for the generation of goal-directed behavioral sequences. In our approach for a connectionist world model a perceptual and a motor layer are coupled to self-organize a model of the perceptual implications of motor activity. A dynamical value field on the learned world model organizes behavior planning—a method in principle borrowed from classical Value Iteration. A major feature of our model is its adaptability. The state space model is developed in a self-organizing way and small world changes require only little re-adaptation of the CWM. The system is continuous in the action, perception, and time domain and all dynamics and adaptivity rely on local interactions only.

Future work will include the more rigorous probabilistic interpretations of CWMs which we already indicated in section 2. Another, rather straight-forward extension will be to replace random-walk exploration by more directed, information seeking exploration methods as they have already been developed for classical world models [20, 21].

Acknowledgments

I acknowledge support from the German Bundesministerium für Bildung und Forschung (BMBF).

References

[1] G. Hesslow. Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6:242–247, 2002.

[2] Rick Grush. The emulation theory of representation: motor control, imagery, and perception. *Behavioral and Brain Sciences*, 2003. To appear.

[3] M.D. Majors and R.J. Richards. Comparing model-free and model-based reinforcement learning. Cambridge University Engineering Department Technical Report CUED/F- IN-ENG/TR.286, 1997.

[4] D.E. Rumelhart, P. Smolensky, J.L. McClelland, and G. E. Hinton. Schemata and sequential thought processes in PDP models. In D.E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, volume 2, pages 7–57. MIT Press, Cambridge, 1986.

[5] M. Jordan and D. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992.

[6] B. Kröse and M. Eecen. A self-organizing representation of sensor space for mobile robot navigation. In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS 1994)*, 1994.

[7] U. Zimmer. Robust world-modelling and navigation in a real world. *NeuroComputing*, 13:247–260, 1996.

[8] S. Amari. Dynamics of patterns formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[9] W.A. Phillips and W. Singer. In the search of common foundations for cortical computation. *Behavioral and Brain Sciences*, 20:657–722, 1997.

[10] L.F. Abbott. Realistic synaptic inputs for network models. *Network: Computation in Neural Systems*, 2:245–258, 1991.

[11] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[12] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, 1998.

[13] C. von der Malsburg. Self-organization of orientation-sensitive cells in the striate cortex. *Kybernetik*, 15:85–100, 1973.

[14] T. Kohonen. *Self-organizing maps*. Springer, Berlin, 1995.

[15] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 5:698–713, 1992.

[16] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[17] C.M. Bishop, G.E. Hinton, and I.G.D. Strachan. GTM through time. In *Proc. of IEEE Fifth Int. Conf. on Artificial Neural Networks*. Cambridge, 1997.

[18] J.C. Wiemer. The time-organized map algorithm: Extending the self-organizing map to spatiotemporal signals. *Neural Computation*, 15:1143–1171, 2003.

[19] P. Dayan and L.F. Abbott. *Theoretical Neuroscience*. MIT Press, 2001.

[20] J. Schmidhuber. Adaptive confidence and adaptive curiosity. Technical Report FKI-149-91, Technical University Munich, 1991.

[21] N. Meuleau and P. Bourgine. Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Machine Learning*, 35:117–154, 1998.