Reward-Driven Learning of Sensorimotor Laws and Visual Features

Jens Kleesiek
and Andreas K. Engel
University Medical Center Hamburg-Eppendorf,
Dept. of Neurophysiology and Pathophysiology,
Martinistr. 52,
20246 Hamburg, Germany
Email: j.kleesiek@uke.uni-hamburg.de

Cornelius Weber
and Stefan Wermter
University of Hamburg,
Department of Informatics,
Vogt-Koelln-Str. 30
22527 Hamburg, Germany

***Abstract*—A frequently reoccurring task of humanoid robots is the autonomous navigation towards a goal position. Here we present a simulation of a purely vision-based docking behavior in a 3-D physical world. The robot learns sensorimotor laws and visual features simultaneously and exploits both for navigation towards its virtual target region. The control laws are trained using a two-layer network consisting of a feature (sensory) layer that feeds into an action (Q-value) layer. A reinforcement feedback signal (delta) modulates not only the action but at the same time the feature weights. Under this influence, the network learns interpretable visual features and assigns goal-directed actions successfully. This is a step towards investigating how reinforcement learning can be linked to visual perception.**

I. Introduction

For a long time philosophers have emphasized the active nature of perception and the intimate relation between action and cognition [1], [2]. However, it took almost a century for the notion of embodied cognition to establish itself in the field of modern cognitive science and robotics [3]–[8]. Varela *et al.* [3] coined the term *enactivism* - meaning that cognitive behavior results from interaction of organisms with their environment, which "appears to be filled with regularities" resulting from past experiences [9].

On the other hand, our environment is usually filled with a plethora of stimuli and to discover those "regularities" we constantly have to discriminate between relevant and irrelevant features. Especially action selection is generally based on specific sensory inputs. Therefore, sensory representations need to reflect a specific task, i.e. sensorimotor relations (laws) have to be learned.

Evidence for long-term changes of sensorimotor neural representations has been obtained during habit learning in the rat striatum [10]. The striatum receives direct cortical input and is part of the basal ganglia. Doya proposed that unsupervised learning happens in the cortex and reinforcement learning in the basal ganglia [11]. Accordingly, the cortex pre-processes data to yield a representation that is suitable for reinforcement learning (RL) by the basal ganglia [12].

The seven deep brain nuclei of the basal ganglia are involved in a variety of crucial brain functions (for a review see [13]) and are tightly linked to the dopaminergic neuromodulatory system, which plays a fundamental role in predicting future rewards and punishment [14], [15]. More precisely, the dopamine signal seems to represent the error between predicted future reward and actually received reward [14]. This has a direct analogy to the temporal difference error, δ, in reinforcement learning models [16], where this error is used to maximize future rewards and avoid punishment. The RL agent interacts with its environment, initially guided by trial and error, seeking to find a mapping between states and actions that will yield the maximal future reward. In other words, it tries to find an optimal motor strategy which is adequate for the given scenario.

However, it remains open how the relevant inputs from the cortex are determined, i.e. which features are read from the cortical activation pattern that are relevant for selecting actions and obtaining rewards. Experiments from Shuler and Bear suggest that RL also occurs in early sensory areas like the primary visual cortex of the rat [17]. This implies a link between RL and feature learning.

From a technical point of view it would be straightforward to first learn the state space, i.e. extract features, with an unsupervised method and then use RL on top of this to find the mapping between states and actions. This two-stage learning is a common approach in the literature. For instance, Legenstein *et al.* [18] train a simple neural network based on rewards on top of features, which before have been extracted with a hierarchical slow feature analysis network. In contrast, the attention-gated reinforcement learning (AGREL) model of Roelfsema *et al.* [19] represents a link between supervised and reinforcement learning. The learning rules lead to the same average weight changes as supervised backpropagation learning. However, learning is slower due to insufficient feedback when the network guesses incorrectly and hence the temporal credit assignment problem is not addressed with this model.

Humanoid robots, like the Nao robot[1], are used in a growing number of ambient caregiver scenarios (e.g. KSERA project[2]).

[1] www.aldebaran-robotics.com
[2] www.ksera-project.eu

One frequently reoccurring task for a robot is autonomous navigation, which is often solved using a world model [20], i.e. the robot has a map of the surrounding area which allows it to do planning. A variant of navigation is docking, in which the robot navigates towards a goal position, e.g. to perform some action like grasping, user interaction or recharging. A docking task usually does not require a map, but is constrained by the affordances of the goal position, at which the robot often needs to arrive with a specific pose. This is a hard delayed RL-problem [21]. In our experiments we only rely on information that is directly available from the robot's own camera. The agent is supposed to learn the relevant visual features and develop sensorimotor laws based on its interactions with the environment. Initially, the robot does not know where the target region is and a reward is only received after the final movement leads to successful docking.

For this purpose, we apply an innovative algorithm that is capable of doing both, extracting task-relevant visual features as well as assigning adequate actions to those, all in a single-step procedure and within one united architecture. The network with winner-take-all-like layers considers goal-relevance of sensory input dimensions, and learns to neglect irrelevant parts of the input. To achieve this, the prediction error δ of the top layer (RL) is not only used to modulate learning of action weights that encode both, value function and action strategy (Q-values), it is also used to adapt the weights of the feature neurons of the lower layer, which are responsible for learning the action-relevant input manifold associated to a specific action. Previously, this approach has been successfully applied to learn action-relevant features of artificial stimuli [22], [23]. Now we demonstrate for the first time its applicability to a realistic robot scenario.

The paper is organized as follows. In section II we present the neural architecture and describe the scenario. Next, we report on two experiments in section III, discuss the results in section IV and conlcude with section V.

II. Scenario and Architecture

A. Architecture and Learning

The model is a two-layer feedforward network (schematically shown in Fig. 1) with full connectivity between adjacent layers. The input layer (320 neurons) holds a sensory vector I, representing a 32×10 pixel grayscale image (Fig. 2 C). A hidden feature layer (either 4 or 36 neurons) learns visual features within its weight matrix W and encodes this information in a state vector s, which is governed by a softmax activation. In turn, s is mapped via the action weights Q to the output layer (4 neurons) representing the currently selected action a.

The learning algorithm, which inherits the top-level structure of the SARSA algorithm [16], can be summarized as follows (for details and a derivation of the gradient descent learning rule see below, section II-B). At the beginning of each trial, the agent is placed at a random position, with the constraint that the landmark indicating the docking position is within its field of view. The agent reads sensor values I to

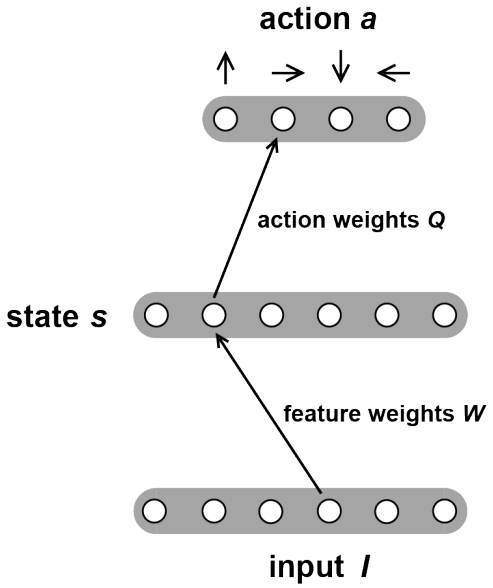

Fig. 1. **Schematic overview of the network architecture.** Only one example connection between any two layers is shown.

obtain the (internal) state activation s_j of neuron j via softmax:

$$h_j = \sum_n W_{jn} I_n \ , \tag{1}$$

$$s_j = \frac{e^{\beta^s_{h_j}}}{\sum_k e^{\beta^s_{h_k}}} \tag{2}$$

We use a large $\beta^s = 100$ for a winner-take-all-like behavior. Next, an action a_i for neuron i is chosen stochastically (via softmax):

$$h_i = \sum_j Q_{ij} s_j \ , \tag{3}$$

$$\texttt{Prob}(a_i{=}1) = \frac{e^{\beta^a h_i}}{\sum_k e^{\beta^a_{h_k}}} \ . \tag{4}$$

During training we use $\beta^a = 2$ to make the agent explore. For testing $\beta^a = 100$ was chosen to exploit the learned skills. Based on the state activation and on the chosen action the value v is computed:

$$v = \sum_{k,l} Q_{kl} a_k s_l \ . \tag{5}$$

The time-discounted (discount factor $\gamma = 0.9$) future value v' and the current value v are used to determine the prediction error δ. A reward $r = 1$ is assigned if the goal position has been reached, otherwise $r = 0$.

$$\delta = r + \gamma v' - v \ . \tag{6}$$

Using a δ-modulated Hebbian rule with state s and action a as pre- and post-synaptic values, respectively, the action layer weights Q can be updated:

$$\Delta Q_{ij} \propto \delta \, a_i s_j \ . \tag{7}$$

In addition to the normal SARSA-algorithm we use the δ signal to modulate learning globally and throughout learning also for the feature layer, even when no reward is given:

$$\Delta W_{jn} \propto \delta s_j I_n (Q_{ij} - \sum_k Q_{ik} s_k) . \quad (8)$$

In each phase of the learning algorithm the feature weights W are rectified to be positive and normalized to length 1, which ensures that a unit that wins for one data point will not also win for all others.

Through the softmax function (Eq. 2) the feature layer performs soft competitive learning. The δ term makes sure that the feature layer learns only when there is learning progress, that is, when currently relevant visual stimuli are encountered. Since unimportant components of the data are not correlated with the learning progress, on average, they will not contribute to learning.

B. Gradient Descent Learning

To get a better understanding of the learning rule (Eq. 7 and 8) and to justify its usage we derive it by performing gradient descent on an energy function.

Let us recall that the action weights Q estimate values v of actions a in states s. These values approximate a value function that increases toward the rewarded state, i. e. the goal of the agent's actions. The network parameters can be summarized with $\theta = (Q, W)$. Following Sutton and Barto [16] (Chapter 8), the values $v = v(\theta)$ will be updated to minimize a mean squared error:

$$E(\theta) = \frac{1}{2} \sum_{s,a} P^\pi(s,a) (V^\pi(s,a) - v(s,a))^2 . \quad (9)$$

$V^\pi(s,a)$ is the "true" value given an action policy π and $v(s,a)$ is the current estimate of the value function. The difference of both, the prediction error δ (see Eq. 6), can be used to improve the estimate v. In practice, V^π is approximated using the information of the better estimate v' obtained in the next time step:

$$V^\pi - v = r + \gamma v' - v = \delta . \quad (10)$$

The probability distribution $P^\pi(s,a)$ weighs this prediction error and represents an *on-policy* distribution of state-action pairs that influence the behavior of the agent. In the presented network the action selection, which in turn alters the visual sensation, is guided by the softmax function (Eq. 4). Both, action and sensation, determine the probability distribution $P^\pi(s,a)$. Hence, the on-line update of network parameters can be expressed as:

$$\Delta\theta \propto -\frac{\partial E}{\partial \theta} = (V^\pi - v) \frac{\partial}{\partial \theta} v = \delta \frac{\partial}{\partial \theta} v . \quad (11)$$

Using $v = \sum_{k,l} Q_{kl} a_k s_l$, as given in Eq. 5 we obtain the action weight update:

$$\Delta Q_{ij} \propto -\frac{\partial E}{\partial Q_{ij}} = \delta \, a_i \, s_j . \quad (12)$$

To compute the derivative of the energy function with respect to the feature weights we need a differentiable transfer function on the feature layer. We choose a softmax function (Eq. 2), which becomes winner-take-all-like with a sufficiently large parameter β. Considering that W_{jn} influences not only s_j of feature unit j but the activations s_k of all feature layer units, we have

$$\begin{aligned} \Delta W_{jn} &\propto -\frac{\partial E}{\partial W_{jn}} = -\sum_k \frac{\partial E}{\partial s_k} \frac{\partial s_k}{\partial h_j} \frac{\partial h_j}{\partial W_{jn}} \\ &= \delta \sum_k Q_{ik} \frac{\partial s_k}{\partial h_j} I_n \end{aligned} \quad (13)$$

assuming that action unit i was the activated one. Using the following identities for the softmax function [24]

$$\frac{\partial s_j}{\partial h_j} = s_j (1 - s_j) \quad (14)$$

and

$$\frac{\partial s_j}{\partial h_{k,k \neq j}} = -s_k s_j \quad (15)$$

we obtain:

$$\begin{aligned} \Delta W_{jn} &\propto \delta Q_{ij} s_j (1 - s_j) I_n - \delta \sum_{k,k \neq j} Q_{ik} s_k \, s_j I_n \\ &= \delta Q_{ij} s_j I_n - \delta \sum_k Q_{ik} s_k \, s_j I_n \\ &= \delta s_j I_n (Q_{ij} - \sum_k Q_{ik} s_k) . \end{aligned} \quad (16)$$

The first term Q_{ij} that arises through backpropagation denotes how strong state neuron j contributes to the output. Since all weights tend to be non-negative when positive rewards are given, one might interpret this factor as influencing learning speed but not the final result. The second term represents a competitive decay term that has a larger suppressive effect if strong activations s_k in the feature layer are paired with large weights Q_{ik}. If a clear winner is found, i.e. exactly one feature unit is active ($s_j = 1$, and for all others $s_{k,k \neq j} = 0$), the first and the second term cancel each other out and learning has converged.

In contrast to the learning rule for the action weights (Eq. 7), the update of the feature weights (Eq. 8) represents a non-local learning rule, because i) the action layer weights Q are involved and ii) it is summed over all activations of the feature layer. Note, by omitting the non-local terms (aggregated in brackets in Eq. 8), we yield a purely local learning rule. This biologically more realistic approximation has been successfully applied to the first experimental scenario presented below. However, for the more difficult task, the modulatory effect on $\delta s_j I_n$ via the non-local terms has been included, mostly because of the necessity when performing vanilla gradient descent.

C. Scenario

Docking of a mobile robot is the initial problem that has to be solved before other applications, e.g. grasping, user

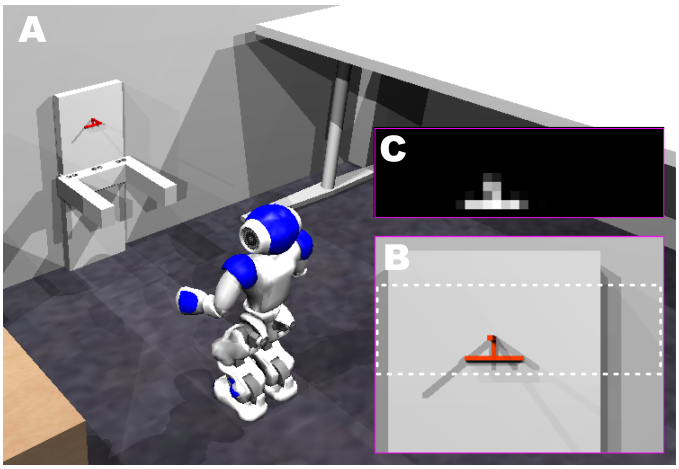

Fig. 2. **Nao robot in front of the docking position.** (A) Webots simulation environment representing a domestic situation. (B) Camera view from the robot. The landmark (red) is located within a white dotted rectangle reflecting the region that serves as an input to the network. (C) Input image I after pre-processing.

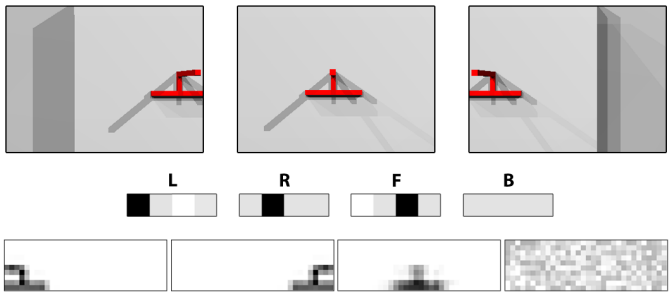

Fig. 3. **Raw visual input (top), receptive fields of action neurons (middle) and hidden feature neurons (bottom) after 100 steps of training.** The raw camera image (top) shows three exemplary situations the robot might encounter (corresponding to a robot position left, in front and to the right of the landmark). The hidden neurons (bottom) code for a specific state, which is then mapped correctly via the action weights Q (middle) to an adequate action, e. g. moving left (L), right (R), forward (F) or backward (B). The action weights for the backward action show no structure, because the backward action is hardly ever executed in this simple scenario. Correspondingly, one hidden unit that is not used by any action unit has no structure. Strong weights are displayed dark.

interaction or recharging can be performed. Therefore, we modeled a general docking situation in a Webots [25] simulation environment (Fig. 2). As a landmark, signaling the target region, serves a 3-D geometric shape with several beneficial attributes. First, depending on the perspective, it generates a different visual impression. From this, the algorithm needs to extract location-specific relevant features and assign them to an adequate action. Next, it can be pre-processed easily. The raw camera image is simply cropped and color-thresholded. After downsizing (32×10 pixels) and a grayscale conversion, it is then directly used as input to the network (Fig. 2 C).

In the simulation the robot performs four actions – moving forward, backward, right and left. In one trial a maximum of 25 steps are allowed for reaching the goal. The robot is randomly initialized in a trapezoidal region in front of the target. Two scenarios have been simulated. In the first experiment the robot is only initialized in close proximity to the target, so that the resolution of the visual input is optimal for the extraction of the visual features. It is a common practice to start with easier situations and then gradually move towards more and more difficult ones. Asada *et al.* coined the term "Learning from Easy Missions" (LEM) for this procedure [26]. Hence, in the second simulation the region is incrementally enlarged during learning to finally span a distance of up to 1.5 m. At a larger distance the robot camera is not able to discriminate the geometrical properties of the stimulus anymore.

III. Results

In the performed Webots simulations the Nao robot is trained to navigate towards the docking position solely based on visual input. In the first experiment it is placed in close proximity to the docking position and encounters visual input similar to the one shown in the top of Fig. 3. After reaching its goal position and receiving a reward for about 25 times, the robot is already able to master the simple task successfully in 100 % of the trials. The bottom part of Fig. 3 shows the receptive fields of the hidden neurons after 100 training steps. The visual features relevant for determining its state and for performing effective navigation have been extracted and stored in the weights connecting the input with the hidden state neurons. In the receptive field depicted in the lower right no structure has evolved. This is due to the fact that i) the state space can be covered completely with the three states captured in the other RFs and ii) the backward action is hardly ever executed in the simple scenario.

In the second simulation the possible initialization region of the robot is gradually increased and due to the vastly growing state space a much harder problem is given. Nevertheless, after training (2000 trials, $\approx$ 2 days[3]) it is able to reach the goal position in 95 % of the cases (690 out of 725 trials). It is capable of identifying the relevant visual features, as shown by the evolved receptive fields (RFs) of the feature and action layer in Fig. 4 and to generate task-specific sensorimotor laws needed for navigation. However, these features are not clearly reflecting the shape of the landmark anymore. Due to the large variations of the landmark's position, scale and perceived shape, the network is not capable of representing all combinations. Therefore, now not only a single state is linked to a specific action, but a mixture of different ones (Fig. 4 top). This "population" coding might be useful for resolving ambiguities. Note, the predominant visual feature for a specific action can still be recognised in the receptive fields (Fig. 4 bottom, RFs framed in red).

In Fig. 5 sample trajectories of the robot are shown. Green trajectories were successful trials, whereas the red ones represent failures. Note that an identical initialization point can result in a completely different trajectory. This is mainly due to noise in movements that is imposed by the Webots simulator to

[3]The combination of the Webots simulator with the Aldebaran Nao API runs in real-time only

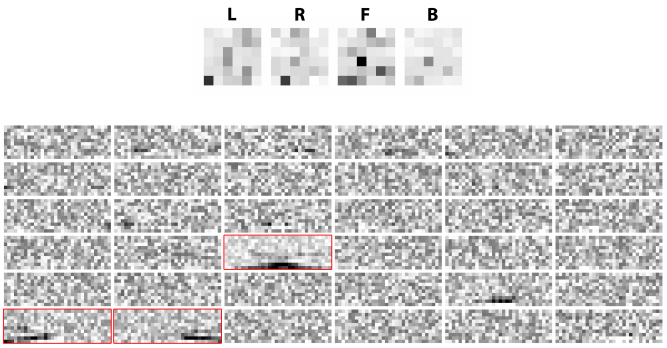

Fig. 4. **Receptive fields of hidden feature neurons (bottom) and action neurons (top) after 2000 steps of training.** The RFs of the action units correspond to left (L), right (R), forward (F) and backward (B) movement. The RFs of the feature neurons that have the strongest contribution on the action units for L,R & F are framed in red. Strong weights are displayed dark.

reflect real-world robot behaviour. Furthermore, this noise can lead to a rotation of the robot, which currently is not compensated, because no rotation movements are implemented. This is actually the reason for most of the unsuccessful trials (red trajectories, Fig. 5).

After training, a receptive field is linked to a specific action, jointly composing a sensorimotor law. Now, if the robot is confronted with a (previously unseen) input, the winning feature unit is the one where the receptive field is most similar to the current input. Hence, the properties (e.g. shape) of this input will trigger the movement embedded in the sensorimotor law. This perceived shape reflects the robot position in relation to the goal.

IV. Discussion

We report on a Webots simulation for navigation and docking towards a virtual target. In the experiments we apply a previously developed two-layer network [22], integrating feature and motor learning in a single-step procedure. As a landmark we use a 3-D geometrical shape (Fig. 2), which leads to perspective distortions depending on robot position and locomotion (Fig. 3). The network learns to exploit this for navigation.

In general, a robot should be aware of the effects of its own actions on objects in the environment and be able to consequently use this knowledge in a goal-directed behaviour. This is achieved by the presented architecture. The network discovers relevant sensory features and stores this information in the weights of the hidden layer (see RFs in Fig. 3 and 4). Simultaneously, sensorimotor laws are learned, which link the current state (comprising physical properties of the object and the sensor, as well as the position of the robot) to a goal-directed action.

Another restriction imposed on an autonomous agent is that noise (e.g. other red objects within the visual field) should not affect performance. This criterion is also fulfilled by our model [22]. The sensorimotor laws acquired during learning are grounded in perception. The prediction error δ modulates the learning in a top-down, action-driven way and allows to

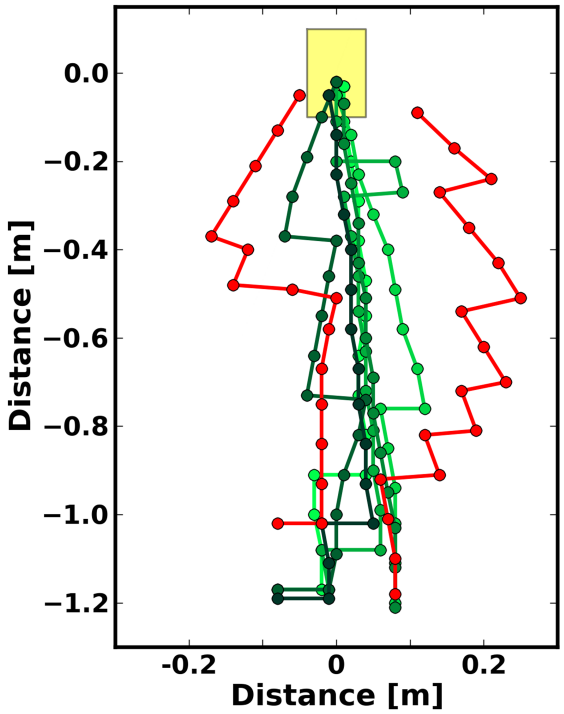

Fig. 5. **Sample trajectories of the robot.** Different shades of green represent successful trials and red failures. The yellow docking region measures 8 × 20 cm.

distinguish between relevant and irrelevant sensory features, despite the fact that the information stemming from the sensory apparatus of the agent can be ambiguous, incomplete and noisy. To even improve robustness, a memory layer could be added to the network [23], which helps to maintain focus on relevant features in phases where they are masked by noise or incomplete.

The presented network learns goal-relevant features within a single united framework. However, if one is willing to give up on training the network with one energy function, a self-organizing map [27] could be used to first learn the visual features, and the resulting map could represent the state space for RL in a succeeding step. There are further methods of unsupervised learning (e. g. exploiting sparseness or a slowness principle) that could also be used to learn the perceptual features. All these models have in common that they do not discriminate between action-relevant and irrelevant features nor do they solve the delayed RL-problem. Therefore, they would be separate parts that would need to be linked *ad hoc* to the reinforcement learning architecture.

Once the sensorimotor laws have been learned and the visual features needed for navigation have been captured in the weights the robot is able to navigate to any position, where a landmark with the same geometrical properties is located. This is a clear advantage compared to models that rely on a world model.

The presented results are in agreement with the theory of sensorimotor contingencies (SMCs) proposed by O'Regan

and Noë [28]. According to this theory, actions are fundamental for perception and help to distinguish the qualities of sensory experiences in different sensory channels (e.g. "seeing","hearing" or "touching"). On this account, it is also imaginable to use e.g. the sonar sensors of the robot for navigation. This could be done in addition, i.e. multimodal, or separately. As a matter of fact, the architecture should also be capable to learn without any landmark at all, but instead exploiting the geometrical shape of the surrounding area of the docking position. However, this is not feasible due to the narrow field-of-view of the built-in camera. While getting closer, the target region eventually is too close to be captured in toto.

Finally, it should be noted, that the proposed model can cope easily with changes of the object, sensors or actuators of the robot.

V. Conclusion & Outlook

We presented an innovative two-layer network that solves the hard delayed RL-problem of visual-based robot docking. The presented architecture is capable of learning sensorimotor laws and visual features simultaneously. However, the focus of the presented work is not to compete with technical solutions, but to present a biologically plausible network model that can learn sensorimotor laws and decision relevant features based only on internally available information.

Currently, we are implementing a real-world precision approach including turning of Nao. Future work aims at allowing continuous actions. This should result in even more accuracy and a shorter path due to exact and oriented movements.

Acknowledgment

This work was supported by the Sino-German Research Training Group CINACS, DFG GRK 1247/1 and 1247/2, by the EU projects RobotDoc under 235065 and KSERA under 2010-248085. We thank S. Heinrich, D. Jessen, R. Luckey, N. Navarro and J. Zhong for assistance with the robot and simulation environment.

References

[1] J. Dewey, "The reflex arc concept in psychology," *Psychological Review*, vol. 3, pp. 357–370, 1896.

[2] M. Merleau-Ponty, *The structure of behavior*. Boston: Beacon Press, 1963.

[3] F. J. Varela, E. Thompson, and E. Rosch, *The embodied mind: cognitive science and human experience*. Cambridge, Mass.: MIT Press, 1991.

[4] A. Clark, *Being there: putting brain, body, and world together again*. Cambridge, Mass.: MIT Press, 1997.

[5] A. Noë, *Action in perception*. Cambridge, Mass.: MIT Press, 2004.

[6] R. Pfeifer, J. Bongard, and S. Grand, *How the body shapes the way we think: a new view of intelligence*. Cambridge, Mass.: MIT Press, 2007.

[7] A. K. Engel, "Directive minds: how dynamics shapes cognition," in *Enaction: towards a new paradigm for cognitive science*, J. Stewart, O. Gapenne, and E. Di Paolo, Eds. MIT Press, 2010, pp. 219–243.

[8] S. Beckhaus and J. Kleesiek, "Intuitive Interaction: Tapping into body skills to find rich and intuitive interaction methods for Virtual Reality," in *CHI 2011 Workshop on "Embodied Interaction"*. to appear, 2011.

[9] H. R. Maturana and F. J. Varela, *The tree of knowledge: the biological roots of human understanding*, rev. ed. Boston: Shambhala, 1992.

[10] M. S. Jog, Y. Kubota, C. I. Connolly, V. Hillegaart, and A. M. Graybiel, "Building neural representations of habits," *Science*, vol. 286, no. 5445, pp. 1745–9, Nov 1999.

[11] K. Doya, "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?" *Neural Networks*, vol. 12, pp. 961–74, 1999.

[12] C. Weber, M. Elshaw, S. Wermter, J. Triesch, and C. Willmot, *Reinforcement Learning: Theory and Applications*, 2008, ch. Reinforcement Learning Embedded in Brains and Robots, pp. 119–42.

[13] V. S. Chakravarthy, D. Joseph, and R. S. Bapi, "What do the basal ganglia do? A modeling perspective," *Biol Cybern*, vol. 103, no. 3, pp. 237–53, Sep 2010.

[14] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–9, Mar 1997.

[15] W. Schultz, "Predictive reward signal of dopamine neurons," *J Neurophysiol*, vol. 80, no. 1, pp. 1–27, Jul 1998.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. Cambridge, Mass.: MIT Press, 1998.

[17] M. G. Shuler and M. F. Bear, "Reward timing in the primary visual cortex," *Science*, vol. 311, no. 5767, pp. 1606–9, Mar 2006.

[18] R. Legenstein, N. Wilbert, and L. Wiskott, "Reinforcement learning on slow features of high-dimensional input streams," *PLoS Comput Biol*, vol. 6, no. 8, 2010.

[19] P. R. Roelfsema and A. van Ooyen, "Attention-gated reinforcement learning of internal representations for classification," *Neural Comput*, vol. 17, no. 10, pp. 2176–214, Oct 2005.

[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005.

[21] H.-M. Gross, V. Stephan, and M. Krabbes, "A neural field approach to topological reinforcement learning in continuous action spaces," in *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, vol. 3, May 1998, pp. 1992 –1997 vol.3.

[22] C. Weber and J. Triesch, "Goal-directed feature learning," in *IJCNN'09: Proceedings of the 2009 International Joint Conference on Neural Networks*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3355–3362.

[23] S. Saeb, C. Weber, and J. Triesch, "Goal-directed learning of features and forward models," *Neural Networks*, vol. 22, no. 5-6, pp. 586 – 592, 2009.

[24] M. Nguyen, "Cooperative coevolutionary mixture of experts. A neuro ensemble approach for automatic decomposition of classification problems," Ph.D. dissertation, University of Canberra, Australia, 2006.

[25] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[26] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive behavior acquisition for a real robot by vision-based reinforcement learning," *Machine Learning*, vol. 23, pp. 279–303, 1996.

[27] T. Kohonen, *Self-Organizing Maps*, 3rd ed., ser. Springer Series in Information Sciences. Springer, Berlin, Heidelberg, New York, 2001, vol. 30.

[28] J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness," *Behav Brain Sci*, vol. 24, no. 5, pp. 939–73; discussion 973–1031, Oct 2001.