

Learning and Use of Sensorimotor Schemata Maps

Claudius Gläser, Frank Joublin, Christian Goerick

2009

Preprint:

This is an accepted article published in Proceedings of the IEEE 8th International Conference on Development and Learning (ICDL). The final authenticated version is available online at: [https://doi.org/\[DOI not available\]](https://doi.org/[DOI not available])

Learning and Use of Sensorimotor Schemata Maps

Claudius Gläser, Frank Joublin, and Christian Goerick

Abstract—In this paper we present a framework for the learning and use of sensorimotor schemata. Therefore, we introduce the concept of a schema as a compact representation of an attractor dynamic and discuss how schemata, if embedded into the proposed architecture, can be used to produce, simulate, or recognize goal-directed behaviors. We further present a first implementation of the framework which incorporates well-founded biological principles. Firstly, we apply population coding for the representation of schemata in a neural map and, secondly, we use basis functions as flexible intermediate representations for sensorimotor transformations. Simulation results show that during an initial motor babbling phase the system is able to autonomously develop schemata which correspond to generic behaviors. Moreover, the learned sensorimotor schemata map is topologically ordered insofar as neighboring schemata represent similar behaviors. In accordance with biological findings on the motor system of vertebrates the schemata form a set of behavior primitives which can be flexibly combined to yield more complex behaviors.

I. INTRODUCTION

In our everyday life we apply rules when interacting with the physical world or social environment. These rules, also called *schemata*, are cognitive structures describing regularities within experiences [1]. They, thus, serve for the organization of our knowledge and mediate how we see and interpret the world [2]. Schemata are hierarchically organized, thereby they represent knowledge at all levels of abstraction [3].

Over the past, *Schema Theory* has been applied in a wide range of fields, one of them being behavior-based robotics. In [4] it has been proposed that *motor schemata* represent generic behaviors of an agent. Thereby, an instantiation of a schema determines action patterns appropriate for reaching the behavior's goal. Moreover, by using a hierarchically organized network of behaviors an agent's goal can be decomposed and accomplished by instantiating schemata which serve the corresponding subgoals [5].

This paper is concerned with the autonomous acquisition of low-level sensorimotor schemata which may serve as a basis for the build up of such schemata hierarchies. Schemata at the lowest level of a hierarchy have to describe the spatio-temporal sensorimotor patterns which an agent observes when he interacts with its environment. They, thus, segment the continuous stream of events into causal chunks. Here, we consider a schema to be a compact representation of an attractor dynamic where the attractor dynamic describes a generic behavior serving a certain goal. We will discuss how schemata, if embedded into the proposed architecture, can be used for the execution, simulation, and recognition of

the corresponding behaviors. Moreover, an implementation of the framework is presented. The implementation does, at its current state, not support a hierarchical organization of schemata. It rather focuses on the learning of low-level schemata as well as the incorporation of biologically plausible processing principles in a coherent framework. Finally, we will present first simulation results which show that the proposed framework is able to autonomously develop sensorimotor schemata which correspond to generic behaviors. Thereby, the learned mapping between schemata and attractor dynamics is topology preserving, i.e. neighboring schemata represent similar behaviors. The schemata further feature properties which are in accordance with biological findings.

The remainder of the paper is organized as follows. In section II we will review existing work on the learning of sensorimotor competencies. Motivated by biological evidence we will next introduce our schemata-based architecture in section III. After that, we will propose a detailed implementation of the framework in section IV before we will present simulation results in section V. Finally, we will give a summary, discuss our approach with respect to existing work, and outline our future work in section VI.

II. RELATED WORK

Based on the mixture of experts architecture [6] Wolpert et al. proposed the *MOSAIC* model [7]. Central to *MOSAIC* are multiple paired forward-inverse models, where the forward models concurrently try to describe the observed sensorimotor patterns and the inverse models cooperatively contribute to the overall control of the agent depending on their forward models' prediction quality. The proposed framework results in a concurrent learning scheme in which the different internal models compete for competence in describing regularities in the agent-environment interaction. Instances in time in which the responsibility of the internal models switch consequently signal behaviorally relevant events which segment the continuous stream of sensorimotor patterns into causal chunks. Following a similar spirit, Taniguchi et al. proposed a multiple module-based architecture which incrementally acquires its internal models [8]. Based on Piaget's theory [1] they proposed to use techniques of statistical hypothesis testing in order to decide whether a newly acquired experience should become assimilated by existing models or should result in the creation of a new model.

In contrast to the use of multiple internal modules, the *RNNPB* model [9] proposes to represent sensorimotor experiences within a single recurrent neural network (RNN), thus, in a distributed fashion. Tani et al. used parametric bias (PB) vectors as input to the RNN in order to determine the

The authors are with the Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach, Germany, phone: +49-69-89011-702, fax: +49-69-89011-749, e-mail: {firstname.lastname}@honda-ri.de.

network's behavior. PB vectors consequently act as switches between the multitude of sensorimotor patterns represented by the RNN. It has further been shown that during offline learning the mapping between PB vectors and sensorimotor patterns self-organizes. PB vectors, thus, serve as a compact description of causal sensorimotor chunks. Finally, Tani et al. proposed to use an inverse iterative search procedure to determine the PB vector which best describes an observation.

Both, MOSAIC and RNNPB, provide means by which an agent may acquire low-level sensorimotor competencies. These competencies, here called schemata, can serve as a basic behavior repertoire on which higher-level competencies can be build on. More precisely, high-level schemata may orchestrate the execution of low-level schemata by which more complex behaviors can be obtained. However, the learning of such *motor programs* is out of the scope of the present paper. Existing approaches concerned with this topic include HMOSAIC [10], HAMMER [11], Petitagé [12], and the developmental assembler of Grupen [13].

III. SCHEMATA-BASED ARCHITECTURE

A. Schema Definition

We assume that schemata describe generic behaviors of an agent. For this reason, we will use both terms interchangeably. Since a generic behavior only carries meaningful information if its application results in a specific situation, a schema is characterized by the goal which the application of such a generic behavior entails. Thereby, the term generic refers to the fact that the behavior can be applied in a variety of situations, but always yields a situation corresponding to its goal. For example a *GazeAtHand* schema should result in a situation where the agent sees its hand in the fovea. However, the spatio-temporal sensorimotor patterns the agent observes when applying the schema might be very different (e.g. depending on the initial gaze and hand position). A schema is, thus, a compact representation of an attractor dynamic which for various contexts describes how to reach a single equilibrium point. Thereby, the dynamic's equilibrium point represents the schema's goal.

B. System Components

Based on this definition of schemata we propose the architecture shown in Fig. 1. Besides a schemata map, whose activity reflects the current states of the generic behaviors (e.g. active or inactive), the architecture is composed of three integral parts. Firstly, given a situation (in terms of a sensory state) and a schema an *Inverse Model* has to select motor commands which are suitable for reaching the schema's goal. Secondly, given a situation an agent has to be able to predict the sensory consequences of applying a schema in this situation. This ability is implemented by the *Forward Model*. Lastly, an agent has to be able to recognize the schemata which best describe an observed stream of perceptual events. The *Schemata Recognizer* thus maps observations onto own experiences. The additionally shown *Switch* allows the agent to either rely on current or predicted observations. By using previously predicted consequences (imagined situations) as

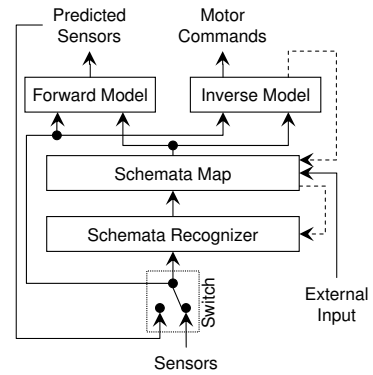


Fig. 1. The schemata-based architecture proposed in this paper. Thereby, the use of hierarchically structured schemata require additional processing pathways which are indicated by the dashed connections. These pathways may include even more functional modules, rather than being direct connection between the components shown in the figure.

input the forward modeling performs a look-ahead prediction of arbitrary time. It thus allows an agent to mentally simulate the application of schemata, which is a property being crucial for planning.

Even though our current implementation does not support the learning and use of schemata hierarchies, we will briefly discuss how the framework could be adapted in order to do so. When using schemata hierarchies an agent's high-level behavior can be decomposed into its subgoals. This means that the *Inverse Model* not only has to specify the motor commands to be executed, but it also has to select a schema's subgoals which can be achieved by other lower-level schemata. This behavior control via goal decomposition is indicated by the dashed arrow in Fig. 1. However, the arrow is not meant to represent a direct connection. Goal decomposition may rather include additional functional modules (e.g. a representation of goals or a schemata selection module). Following a similar argumentation, the *Schemata Recognizer* may not only rely on the observed stream of sensory patterns, but also on a stream of already recognized subgoals (at a lower level of the hierarchy). Consequently, schemata hierarchies could be used to recognize or infer the high-level goals of interaction partners even though the agent did not observe the whole behavior.

C. Biological Evidence

Strong evidence for schemata-based behavior control in vertebrates comes from studies on the spinal motor system of the frog. Bizzi and co-workers showed that microstimulation of neurons in the spinal cord evokes motor responses which can be described in terms of force fields [14]. More precisely, these force fields map the frog's initial hindlimb position on motor responses which direct the hindlimb towards an equilibrium point in space [15]. Therefore, these force fields perfectly fit our definition of schemata. It has further been shown that the simultaneous activation of multiple motor primitives results in force fields equivalent to a linear superposition of the single synergies [16]. Fig. 2 exemplifies the motor primitive theory by showing three force fields similar to those found in vertebrates. There, the force field shown in (c) is the summation of the force fields in (a) and (b).

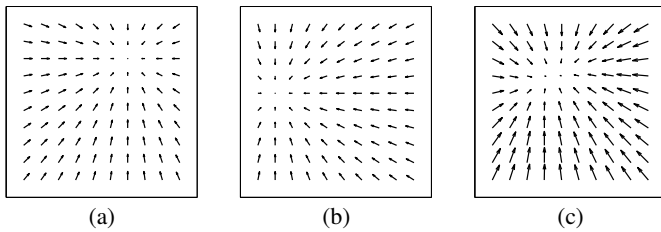


Fig. 2. The plots show force fields directed toward an equilibrium point in space. Thereby, the force field in (c) is a linear superposition of the force fields in (a) and (b).

Neuron populations featuring schemata-like properties were not only reported for the spinal motor system, but also for higher areas of the motor system, particularly the ventral premotor cortex (vPMC). For example it has been shown that neurons in area F4 of monkey vPMC code wrist movements directed toward specific locations in space [17]. Moreover, in area F5 neurons coding for action classes like ‘*grasping*’ or ‘*holding*’ have been reported [18]. These and other studies suggest a hierarchical organization of behaviors, starting from motor primitives coded in the spinal cord and the primary motor cortex, through action classes represented in vPMC, up to behavior plans stored in lateral prefrontal cortex [19].

Lastly, we would like to highlight the role of the parietal cortex. Beside the fact that the inferior parietal lobe is part of the mirror neuron circuit [20] the posterior parietal cortex plays a pivotal role in sensorimotor integration [21]. It integrates sensory information stemming from different modalities and in turn provide strong input to frontal brain regions. This frontoparietal circuit carries out sensorimotor transformations related to the control of different movements [22]. Recently, it has been suggested that computational models for sensorimotor transformation should use basis functions as flexible intermediate representations since they best describe the response characteristics of cortical neurons [23].

IV. IMPLEMENTATION

As previously noted our current implementation of the architecture does not include hierarchical dependencies between schemata. Therefore, the system presented in this section does, at its current state, not include all the functionalities which were discussed in the previous section. The focus of this work rather lays on the online learning of low-level sensorimotor schemata and the embedding of different processing principles within a coherent framework (see Fig. 3).

A. Implementation of the System Components

One of the principles we apply here is that of population coding [24]. More precisely, we assume that units distributed in a 2-dimensional map represent schemata. We further consider the activity within this map to code the multitude of simultaneously active schemata. What the framework then should achieve is to learn a topology preserving mapping from sensorimotor pattern sequences to schemata. In other words, schemata should be topographically organized such that neighboring units represent similar behaviors and therewith also serve similar goals.

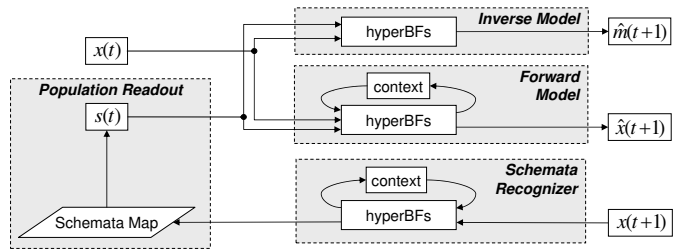


Fig. 3. The diagram illustrates the current implementation of our framework.

Secondly, we implemented the *Forward Model* by a single recurrent neural network (RNN). Therefore, our approach for sensory prediction is similar in spirit to the *RNNPB* model [9]. There, Tani et al. proposed that sensorimotor patterns are distributely represented within a single RNN, where parametric bias vectors determine the network’s dynamics. Tani et al. further suggested that higher-level modules set these biases and thereby drive the network in its corresponding mode. Here, we propose that the active schemata determine the network’s behavior. Thus, the schemata map activity acts as a parametric bias. In summary, the forward prediction is modeled via an RNN using one hidden layer and context units, where the context unit activity at the output is fed back to the context unit activity at the input. The sensory state $x(t)$ and the schemata map activity $s(t)$ serve as input to the RNN which in turn predicts the sensory state $x(t+1)$ at the next timestep.

Next, the *Inverse Model* has been implemented as a feed-forward neural network with one hidden layer. Similar to the *Forward Model* the sensory state $x(t)$ as well as the schemata map activity $s(t)$, which represents the currently applied behaviors, serve as input to the network. The *Inverse Model* finally produces motor commands $m(t)$ suitable for reaching the schemata’s goals.

Lastly, the *Schemata Recognizer* has been modeled by an additional RNN. One more time, the RNN consists of one hidden layer and context units, where the output context activity is used as input at the next timestep. The *Schemata Recognizer* maps an observation $x(t+1)$ onto own experiences insofar as it activates the schemata $s(t)$ which best describe the observation. In order to ensure that hidden layer and context unit activities reflect causalities in the observed sensory pattern stream the *Schemata Recognizer* also predicts the observation $x(t+2)$ at the next time step. This forward prediction thus serves computational purposes and may be omitted in future.

Motivated by the existence of basis function-like neuron discharges in parietal cortex [23] our architecture incorporates basis functions as flexible intermediate representations in the hidden layers. More precisely, we adopted the hyper basis function (HyperBF) framework [25] in order to implement the *Forward Model*, the *Inverse Model*, as well as the *Schemata Recognizer*.

B. Hyper Basis Function Networks

According to (1) a HyperBF network approximates a multivariate function $f(z)$ by a weighted combination of basis function activities as well as a bias b . Thereby, the weighted

norm in (2), which incorporates the basis functions' centers ξ_i and weighting matrices W_i , serves as an activation function, whereas the radial function G calculates the basis function activities. Here, G has been chosen according to (3).

$$f(z) \approx \hat{f}(z) = b + \sum_{i=1}^N \alpha_i \cdot G(\|z - \xi_i\|_{W_i}^2) \quad (1)$$

$$\|z - \xi_i\|_{W_i}^2 = (z - \xi_i)^T W_i^T W_i (z - \xi_i) \quad (2)$$

$$G(\lambda) = \exp^{-\lambda} \quad (3)$$

It is known that given a sufficiently high number of hidden units a HyperBF network can approximate any multivariate continuous function arbitrary well. Since the receptive fields of the basis functions are subject to change via some learning algorithm (see section IV-E), HyperBF networks perform a task-dependent clustering as well as dimensionality reduction [25]. Together with the biological evidence for basis function-like neuron discharges in parietal cortex these properties let HyperBF networks become well suited for sensorimotor transformations.

Theoretically the number of basis functions has to grow exponentially in the number of input dimensions, a problem known as the curse of dimensionality. Since HyperBF networks perform a dimensionality reduction, they are not as prone to this problem as other networks are. Nevertheless, we tried to minimize the number of input dimensions in order to make our implementation computationally feasible. Therefore, we do not feed the whole schemata map activity to the HyperBF networks; we rather perform a population readout on the schemata map and use the locations of the resulting peaks as input. By doing so, a 2-dimensional rather than a \mathcal{M} -dimensional input is used to represent the schemata map activity (where \mathcal{M} is the number of schemata).

C. Population Readout Mechanism

Let $p_i = (p_i^x, p_i^y)^T$ be the position of the schemata map's unit at grid index i . Furthermore, let $I_i(t)$ be the input to that unit at time t . Then we first apply a sigmoidal function on the input in order to ensure positive activities of the units.

$$NI_i(t) = \frac{1}{1 + \exp(-I_i(t))} \quad (4)$$

Next, we perform a population readout which is similar in spirit to [26]. Thereby, the map units interact via two types of lateral connections. Firstly, a pooling is accomplished via excitatory lateral weights $w_{i,j}^{exc}$ and, secondly, inhibitory weights $w_{i,j}^{inh}$ implement divisive normalization. We set both excitatory and inhibitory weights according to (5) where $\star \in \{exc, inh\}$ and $\sigma^{\star} = 2 \cdot \sigma^{exp}$.

$$w_{i,j}^{\star} = \exp \left[- \left(\frac{\|p_j - p_i\|_2}{\sigma^{\star}} \right)^2 \right] \quad (5)$$

Iterating equations (6) and (7) for \mathcal{K} times let the map activity

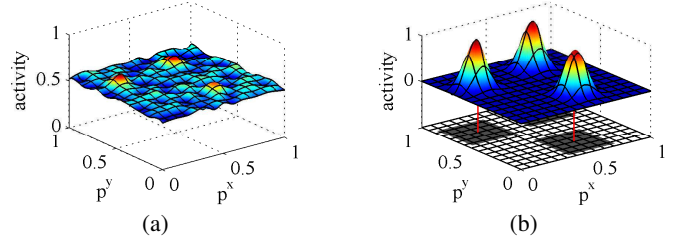


Fig. 4. The population readout mechanism let the initial schemata map activity (a) evolve to smooth peaks (b). In (b) the local neighborhoods of the units exhibiting peak responses are additionally shown.

$a_i(t)$ relax to smooth peaks.

$$u_i^{\kappa+1}(t) = \sum_j w_{i,j}^{exc} \cdot a_j^{\kappa}(t) \quad (6)$$

$$a_i^{\kappa+1}(t) = \frac{u_i^{\kappa+1}(t)^2}{\eta^{\kappa+1}(t) + \mu \cdot \sum_j w_{i,j}^{inh} \cdot u_j^{\kappa+1}(t)^2} \quad (7)$$

We set the initial activity $a_i^0(t) = NI_i(t)$, the divisive normalization weight $\mu = 1$, and $\eta^{\kappa}(t) = 4 \cdot \sum_j u_j^{\kappa}(t)/N$.

Let $\mathcal{P}(t)$ be the set of map indices whose units exhibit peak responses at time t . Then, the set of peak locations $\mathcal{S}(t)$ were obtained by calculating the center of masses within the local neighborhoods n of the units in $\mathcal{P}(t)$.

$$\mathcal{S}(t) = \left\{ s(t) = \frac{\sum_j n_{i,j} \cdot p_j \cdot a_j^{\kappa}(t)}{\sum_j n_{i,j} \cdot a_j^{\kappa}(t)} \mid \forall i \in \mathcal{P}(t) \right\} \quad (8)$$

$$n_{i,j} = \begin{cases} 1, & \text{if } \|p_j - p_i\|_2 \leq r \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Here, the radius r determining the size of the neighborhoods is set to $r = 3 \cdot \sigma^{exp}$. Fig. 4 illustrates the result of the population readout mechanism.

D. Handling Multiple Simultaneously Active Schemata

Let $z(t)$ be the input to a HyperBF network. The input is composed of the peak location $s(t)$ of the schemata map as well as other inputs $i(t)$. Assuming the population readout mechanism results in M peaks at time t the set of peak locations is $\mathcal{S}(t) = \{s_1(t), s_2(t), \dots, s_M(t)\}$. Then we define the set of inputs $\mathcal{Z}(t)$ at time t according to (10). Furthermore, we define the activity $G_j(t)$ of hyper basis function j at time t according to (11).

$$\mathcal{Z}(t) = \{z_i(t) = (s_i(t)^T, i(t)^T)^T \mid \forall s_i(t) \in \mathcal{S}(t)\} \quad (10)$$

$$G_j(t) = \max_{z \in \mathcal{Z}(t)} G(\|z - \xi_j\|_{W_j}^2) \quad (11)$$

E. Learning Sensorimotor Schemata

For learning the parameters of the *Forward Model*, of the *Inverse Model*, as well as of the *Schemata Recognizer* we assume that the agent observes a stream of sensorimotor patterns. Such a stream might be produced during an initial motor babbling phase or through direct guidance.

Then the following strategy has been applied. Given the observations the *Schemata Recognizer* activates the schemata

which best describe the sensory pattern sequence. The recognized schemata are in turn used by the *Forward* and *Inverse Model* in order to predict the sensorimotor patterns. Finally, we calculated the prediction error of the *Forward Model* as well as the *Inverse Model* and applied the Backpropagation Through Time (BPTT) algorithm in order to adjust all network parameters [27]. In order to make the learning algorithm capable for online operation the truncated version of the BPTT algorithm can be used. The important derivatives for error backpropagation are given in the appendix.

V. SIMULATION RESULTS

A. Experiment 1

As a first test of the proposed framework we produced sensorimotor pattern sequences using a predefined controller $\mathcal{C}(\chi, x)$. The controller dynamically changes the values of the state variables $x = (x_1, x_2)^T$ according to (12) such that the target values $\chi = (\chi_1, \chi_2)^T$ become reached. The target values were randomly chosen from the interval $[0, 10]^2$ and fed into the controller. Thereby, we set $\dot{x}_{max} = \ddot{x}_{max} = 200$ and sampled the dynamics with $dt = 0.01$ s.

$$\tau \dot{x} = -\dot{x}_{max} \cdot \left[\frac{2}{1 + \exp\left(-\frac{\ddot{x}_{max}}{\dot{x}_{max}} \cdot (x - \chi)\right)} - 1 \right] \quad (12)$$

This collection of experiences should model an initial motor babbling phase, where an agent randomly executes motor commands and observes their consequences. Here, the system observes the state variables $x = (x_1, x_2)^T$ as well as motor commands m which are assumed to equal the controller's parameters $m = (\chi_1, \chi_2)^T$.

The schemata map is composed of 100 units equally distributed on a 10x10 grid. Furthermore, each of the system components feature 30 hyper basis functions in their hidden layers. The RNNs of the *Forward Model* as well as the *Schemata Recognizer* additionally consist of 2 context units, respectively. The learning was carried out during online operation as described in section IV-E. Thereby, a truncation depth of 8 samples has been used.

The learning algorithm outlined above should autonomously develop sensorimotor schemata corresponding to generic behaviors. It should further self-organize a mapping between schemata and attractor dynamics which is topology preserving. Once the system acquired the schemata it can use them to recognize, reproduce, or simulate the corresponding behaviors. Here, we first show the results for the simulation of the behaviors. Therefore, after learning the network parameters were frozen. Next, we activated each schema in different initial situations $x(0)$ and recorded the sensory pattern sequences $(x(1), x(2), \dots)$ which the *Forward Model* produced using look-ahead prediction (i.e. the prediction at time t has been used as input to the *Forward Model* at time $t + 1$).

Given the predicted sensory pattern sequences we calculated the equilibrium points which the applications of the different schemata entail. The equilibrium points thus describe the goals of the schemata. Fig. 5 shows these goals (blue circles) in the x_1 - x_2 -plane as well as the sensory pattern sequences of five

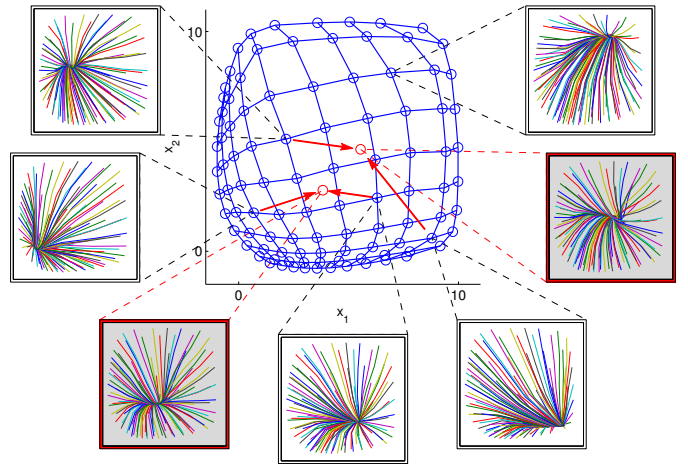


Fig. 5. The attractor dynamics represented by the different schemata are illustrated. Therefore, the white insets show sensory pattern sequences produced by the *Forward Model* when activating one out of five exemplarily chosen schemata in different situations, respectively. Blue circles correspond to the equilibrium points of the dynamics represented by the schemata. They, thus, illustrate the goals of the schemata. The plot further shows that the learned mapping is topology preserving insofar as neighboring schemata serve similar goals. Lastly, the gray insets show two attractor dynamics which were obtained when simultaneously activating pairs of schemata. As can be seen, the simultaneous activation of schemata entails goals (red circles) different from those obtained when activating single schemata (blue circles).

exemplarily chosen schemata (white insets). As can be seen, the learning algorithm developed generic behaviors insofar as the application of a schema in different situations drives the predictions to a single equilibrium point.

For each pair of neighboring schemata (where the neighborhood is given by the 2D-grid topology of the schemata map) we additionally connected the corresponding goals. As can be seen, the goals of the different schemata adequately sample the target space $[0, 10]^2$. Moreover, the resulting map is nicely ordered which means that the learned mapping between attractor dynamics and schemata is topology preserving, i.e. neighboring schemata represent similar attractor dynamics.

When activating multiple schemata simultaneously, attractor dynamics different from those obtained by activating single schemata can be produced. This fact is also illustrated in Fig. 5. There, we exemplarily show the sensory pattern sequences (gray insets) as well as the corresponding goals (red circles) for two pairs of simultaneously activated schemata, respectively. These examples illustrate that the learned schemata form a basis set of attractor dynamics. By differently combining schemata other attractor dynamics, which might be even more complex, can be produced. The learned schemata, thus, nicely resemble the motor primitives found in vertebrates (see Section III-C).

Next, we demonstrate the performance of the *Schemata Recognizer*. The recognizer should activate the schemata which best describe an observed sensory pattern sequence. Therefore, we produced an example sensor trajectory which is shown in the top panel of Fig. 6. Thereby, the vertical bars indicate instances in time where we switched the attractor dynamic. This sensory pattern sequence was fed into the *Schemata Recognizer*. Fig. 6 shows the resulting locations of

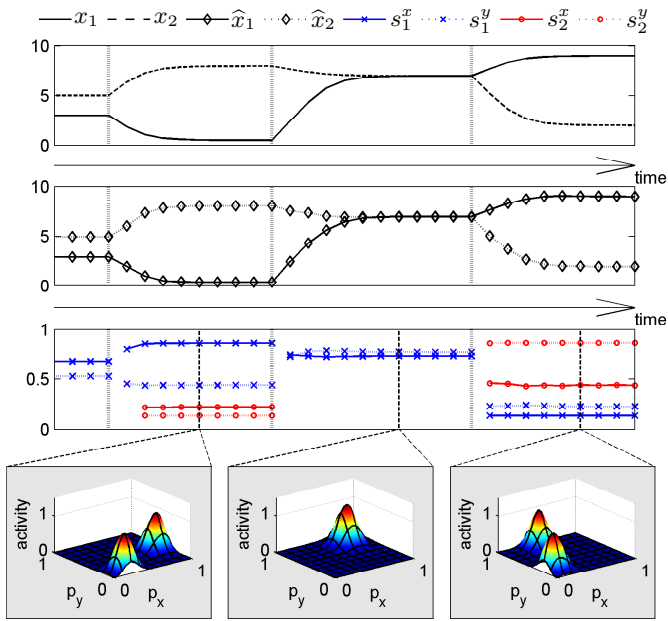


Fig. 6. The recognition of schemata is illustrated. The top panel shows an example trajectory of sensor patterns x where vertical bars indicate time instances when the attractor dynamic has been switched. The bottom panel shows the peak locations s_1 and s_2 in the schemata map as produced by the recognition module. Furthermore, the insets at the bottom show the schemata map activity corresponding to the shown peak locations for three instances in time. Finally, by using the recognized schemata as input the *Forward Model* predicted a sequence of sensor patterns \hat{x} . This predicted sequence is shown in the mid panel. As can be seen, the observed and predicted sensor trajectories are very similar which demonstrates that the schemata were correctly identified by the *Schemata Recognizer*.

the peaks in the schemata map (bottom panel) as well as the corresponding schemata map activity for three instances in time (insets at the bottom). Finally, the peak locations were used as input to the *Forward Model* which in turn predicted the sensory pattern sequence shown in the mid panel of Fig. 6. Since the predicted sensor trajectory nicely resembles the observations, the *Schemata Recognizer* correctly identified the applied attractor dynamic. More precisely, the peaks in the schemata map are adjusted during the initial time steps after the attractor dynamic has been switched. This is indicated by the observed discontinuities in peak locations (bottom panel) and corresponds to an initial guess of which schemata have been applied. Later on, when more patterns of the dynamic have been observed, the initial guess is just slightly adjusted insofar as peak locations just slightly change.

B. Experiment 2

In a second experiment we increased the complexity of the task insofar as an agent capable of performing gaze and hand movements (in two dimensions) has been simulated. Both types of movements were controlled according to (12). Consequently, the agent observed the hand and gaze positions via its sensory variables $x = (x_1^{gaze}, x_2^{gaze}, x_1^{hand}, x_2^{hand})^T$ as well as the corresponding motor commands. Similar to the first experiment we trained the system components during an initial motor babbling phase and performed a post-training analysis in order to evaluate the acquired schemata.

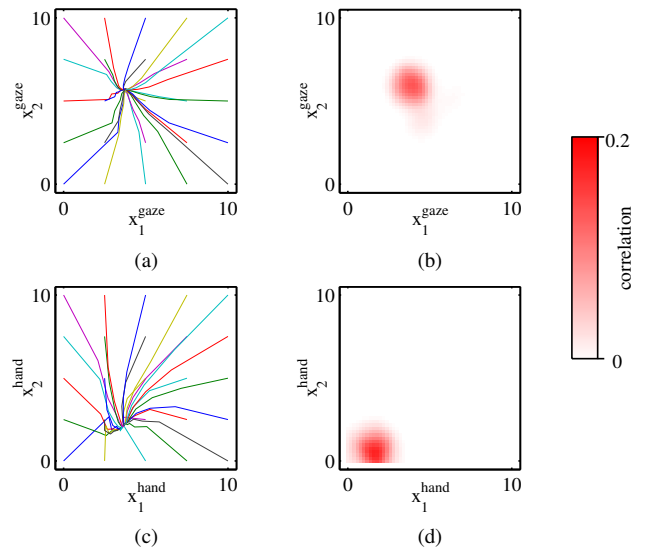


Fig. 7. The activation of a specific schema resulted in gaze and hand movement dynamics depicted in (a) and (c), respectively. The correlation matrices in (b) and (d) show that the schema participates in both, the execution of gaze movements directed towards the center as well as the execution of hand movements towards the lower left corner.

As before the single schemata were activated in different initial situations $x(0)$ and fed to the *Forward Model* which predicted sensory pattern sequences $(x(1), x(2), \dots)$. This analysis allowed us to assess the attractor dynamics which are represented by the different schemata. However, this type of analysis is not valid for cases in which a behavior is distributedly represented, i.e. represented by the simultaneous activation of multiple schemata. For this reason we additionally produced goal-directed sensory pattern sequences and recorded the schemata activities as produced by feeding the observations into the *Schemata Recognizer*. After that, we calculated the correlation between the activity of the different schemata and the target locations of either gaze or hand movements. By doing so, we were able to assess which schemata participate in the execution of which behavior.

Fig. 7 summarizes the results obtained for one particular schema. As shown in (a), the activation of the schema produces goal-directed gaze movements similar to those obtained in the first experiment. Furthermore, the correlation matrix shown in (b) nicely matches the dynamic of (a), insofar as the target location with maximum correlation coincides with the equilibrium point of the dynamic. However, this schema not only participates in the execution of a particular gaze behavior, but also in the execution of hand movements directed toward the lower left corner. This is indicated by the attractor dynamic and correlation matrix shown in (c) and (d), respectively.

From these results we can derive two hypotheses: Firstly, schemata seem to represent a continuum between gaze and hand movement dynamics rather than being related to either a gaze or a hand movement. And secondly, the results suggest a distributed representation of behaviors since only the schemata map activity as a whole may decide whether a gaze or a hand movement will be executed.

In order to prove the former hypothesis we calculated the

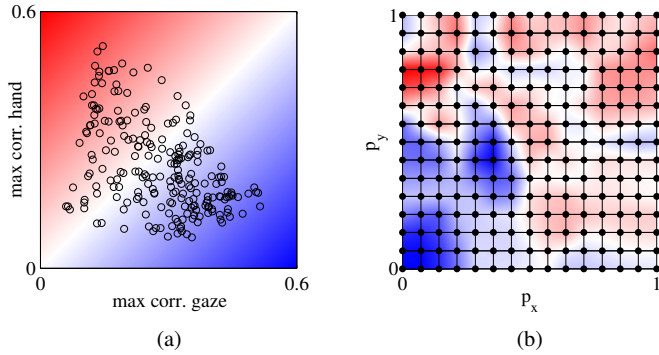


Fig. 8. The maximum correlation of each schema with gaze and hand movements is plotted in (a). Thereby, the color coding indicates that blue-colored schemata are more related to the execution of gaze movements whereas red-colored schemata are more related to the execution of hand movements. In (b) we projected the color code onto the schemata map.

maximum correlations of schemata with the target locations of both types of movements and plotted them in Fig. 8 (a). We further applied a color coding which indicates that blue-colored schemata are more related to gaze movements than hand movements whereas the contrary holds for red-colored schemata. The plot confirms our hypothesis insofar as a clear distinction between the two movement classes cannot be observed. We rather found a multitude of schemata (white-colored) which participate in both, gazing and reaching behaviors. In Fig. 8 (b) we additionally applied the color coding on the schemata map. As can be seen, schemata located in the upper right of the map mainly participate in the execution of hand movements whereas gaze movements are mainly represented by schemata located in the lower left of the map.

We could confirm our second hypothesis regarding the distributed representation of behaviors via visual inspection. Here, we want to illustrate this fact on the example of a gaze movement to the upper right corner. Therefore, Fig. 9 shows the correlation matrices of the three schemata whose simultaneous activation resulted in such a movement. As can be seen, a distributed representation has been learned insofar as a movement to the upper right is represented via a combination of movements to the top and a movement to the right. This finding provides even stronger evidence for our proposal that the architecture is able to acquire a combinatorial code composed of behavior primitives which can be flexibly combined to yield other behaviors.

VI. DISCUSSION & FUTURE WORK

In this paper we presented our efforts towards an implementation of a schemata-based architecture. Therefore, we first introduced the concept of a schema and discussed how schemata, if embedded into an adequate architecture, can be used for goal-directed behavior control, planning, as well as action understanding. As a first step we implemented the proposed architecture for non-hierarchically organized schemata where we applied two well-founded biological principles: population coding for the representation of schemata in a neural map as well as basis functions as intermediate representations for sensorimotor transformations. We presented simulation results

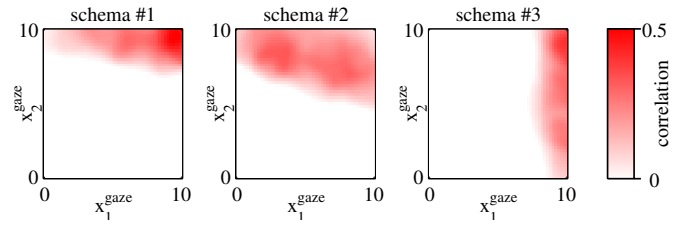


Fig. 9. The correlation matrices of three schemata, whose simultaneous activation produced a gaze movement to the upper right corner, are shown. This plot indicates that a distributed representation of behaviors has been learned. More precisely, the gaze movement to the top and to the right is represented by a combination of movements to the top and a movement to the right.

which showed that the framework autonomously develops schemata during an initial motor babbling phase. Additionally, the learning resulted in a topologically ordered map of schemata insofar as neighboring units represent similar generic behaviors. We further highlighted the fact that the learned schemata form a set of behavior primitives. The architecture consequently applies a combinatorial code in which behaviors are represented by a combination of synergies. The schemata therefore feature properties similar to the motor primitives reported for vertebrates.

Despite the fact that existing work on the learning and segmentation of sensorimotor sequences does not incorporate the concept of a schema, the differences of our framework compared to these approaches are manifold. Firstly, compared to *MOSAIC* we do not use multiple internal models. Rather our framework strives for a distributed representation of the different dynamics and therefore follows the spirit of *RNNPB*. We think that a distributed representation of dynamics will ultimately be more advantageous than the memorization of them in local modules. This may particularly be the case with respect to generalization capabilities. However, in contrast to *RNNPB* our framework uses schemata (behaviors) instead of PB vectors in order to switch between the different dynamics. Next, our architecture incorporates a module for the recognition of schemata which therefore replaces the iterative inverse search procedure of *RNNBP*. Most importantly, our model allows multiple schemata to be simultaneously active. These schemata cooperatively predict the consequences of their application as well as cooperatively contribute to the overall control of the system. The architecture consequently allows us to apply a distributed coding of behaviors which results in a compact yet flexible representation. The implicit redundancy within the representation may further contribute to an increased robustness of the system.

Our future work will include the testing of our framework in more complex scenarios. Furthermore, we will extend our current implementation of the architecture such that hierarchies of schemata can be learned and used. The incorporation of such hierarchical structures is fundamental for the organization of behavior in order to let the approach scale up to more complex scenarios. Different approaches for the extension of behaviors based on a basic repertoire of control routines have been proposed [10], [11], [12], [13]. Most of these methods rely on the assumption that single behaviors are encapsulated

in local modules. They therefore may not be able to cope with a distributed representation. Finally, we strive for an embodied implementation on a robot where we would like to exploit the use of schemata for goal-directed behavior control and situated language understanding.

APPENDIX

For a hyper basis function network according to (1)-(2) we define the error functional E to be $E = \frac{1}{2} \sum_{j=1}^M e_j^2$ where $e = f(z) - \hat{f}(z)$ denotes the prediction error. Then the important derivatives for learning via error backpropagation are:

$$\frac{\partial E}{\partial b} = e \quad (13)$$

$$\frac{\partial E}{\partial \alpha_i} = -e G(\|z - \xi_i\|_{W_i}^2) \quad (14)$$

$$\frac{\partial E}{\partial \xi_i} = 2(e^T \alpha_i) G'(\|z - \xi_i\|_{W_i}^2) W_i^T W_i (z - \xi_i) \quad (15)$$

$$\frac{\partial E}{\partial W_i} = -2(e^T \alpha_i) G'(\|z - \xi_i\|_{W_i}^2) W_i (z - \xi_i) \quad (16)$$

$$\frac{\partial E}{\partial z} = -2 \sum_{i=1}^N (e^T \alpha_i) G'(\|z - \xi_i\|_{W_i}^2) W_i (z - \xi_i) \quad (17)$$

Let $\delta s_{i,t} = \partial E / \partial s_i(t)$ be the error backpropagated to the peak location $s_i(t)$ at time t which corresponds to the peak response at unit i of the schemata map. Then the important derivatives for the population readout mechanism (4)-(9) are:

$$\frac{\partial E}{\partial a_j^{\kappa}(t)} = \frac{n_{i,j} \cdot (p_j - s_i(t))^T}{\sum_k n_{i,k} \cdot a_k^{\kappa}(t)} \delta s_{i,t} \quad (18)$$

$$\frac{\partial a_i^{\kappa}(t)}{\partial u_j^{\kappa}(t)} = \begin{cases} \frac{2u_j^{\kappa}(t) - u_j^{\kappa}(t) a_i^{\kappa}(t) \cdot (\frac{4}{N} + 2w_{i,j}^{inh})}{\eta^{\kappa}(t) + \mu \cdot \sum_k w_{i,k}^{inh} \cdot u_k^{\kappa}(t)^2}, & i = j \\ \frac{-u_j^{\kappa}(t) a_i^{\kappa}(t) \cdot (\frac{4}{N} + 2w_{i,j}^{inh})}{\eta^{\kappa}(t) + \mu \cdot \sum_k w_{i,k}^{inh} \cdot u_k^{\kappa}(t)^2}, & i \neq j \end{cases} \quad (19)$$

$$\frac{\partial u_i^{\kappa}(t)}{\partial a_j^{\kappa-1}(t)} = w_{i,j}^{exc} \quad (20)$$

$$\frac{\partial a_i^0(t)}{\partial I_i(t)} = a_i^0(t) \cdot (1 - a_i^0(t)) \quad (21)$$

REFERENCES

- [1] J. Piaget, *The origins of intelligence in children*, International Universities Press, 1952.
- [2] M. Johnson, *The body in the mind: The bodily basis of meaning, imagination, and reason*, University of Chicago Press, 1987.
- [3] D. Rumelhart, *Schemata: The building blocks of cognition*, pp. 33–58, Erlbaum, 1980.
- [4] Ronald C. Arkin, “Motor schema based navigation for a mobile robot: An approach to programming by behavior,” in *Proc. ICRA*, 1987.
- [5] R. Cooper and T. Shallice, “Contention scheduling and the control of routine activities,” *Cogn Neuropsych*, vol. 17, no. 4, pp. 297–338, 2000.
- [6] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts,” *Neural Computation*, vol. 3, pp. 79–87, 1991.
- [7] D. Wolpert and M. Kawato, “Multiple paired forward and inverse models for motor control,” *Neural Netw*, vol. 11, no. 7-8, pp. 1317–1329, 1998.
- [8] T. Taniguchi and T. Sawaragi, “Incremental acquisition of multiple nonlinear forward models based on differentiation process of schema model,” *Neural Netw*, vol. 21, no. 1, pp. 13–27, 2008.
- [9] J. Tani, M. Ito, and Y. Sugita, “Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB,” *Neural Netw*, vol. 17, no. 8-9, pp. 1273–1289, 2004.
- [10] Masahiko Haruno, Daniel M. Wolpert, and Mitsuo Kawato, “Hierarchical mosaic for movement generation,” *International Congress Series*, vol. 1250, pp. 575–590, 2003.
- [11] Y. Demiris and B. Khadhour, “Hierarchical attentive multiple models for execution and recognition of actions,” *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 361–369, 2006.
- [12] G. Stojanov, “Petitag: A case study in developmental robotics,” in *Proc. 1st Int. Workshop on Epigenetic Robotics*, 2001.
- [13] R. Grupen, “A developmental organization for robot behavior,” in *Proc. 3rd Int. Workshop on Epigenetic Robotics*, 2003.
- [14] E. Bizzi, A. D’Avella, P. Saltiel, and M. Tresch, “Modular organization of spinal motor systems,” *Neuroscientist*, vol. 8, pp. 437–442, 2002.
- [15] F. A. Mussa-Ivaldi and E. Bizzi, “Motor learning through the combination of primitives,” *Philos Trans R Soc Lond B Biol Sci*, vol. 355, no. 1404, pp. 1755–1769, 2000.
- [16] T. Flash and B. Hochner, “Motor primitives in vertebrates and invertebrates,” *Curr Opin Neurobiol*, vol. 15, no. 6, pp. 660–666, 2005.
- [17] S. Kakei, D. S. Hoffman, and P. L. Strick, “Direction of action is represented in the ventral premotor cortex,” *Nat Neurosci*, vol. 4, no. 10, pp. 1020–1025, 2001.
- [18] G. Rizzolatti, L. Fogassi, and V. Gallese, “Motor and cognitive functions of the ventral premotor cortex,” *Curr Opin Neurobiol*, vol. 12, no. 2, pp. 149–154, 2002.
- [19] J. Tanji, K. Shima, and H. Mushiake, “Concept-based behavioral planning and the lateral prefrontal cortex,” *Trends Cogn Sci*, vol. 11, no. 12, pp. 528–534, 2007.
- [20] G. Rizzolatti and L. Craighero, “The mirror-neuron system,” *Annu Rev Neurosci*, vol. 27, pp. 169–192, 2004.
- [21] R. A. Andersen and C. A. Buneo, “Sensorimotor integration in posterior parietal cortex,” *Adv Neurol*, vol. 93, pp. 159–177, 2003.
- [22] G. Rizzolatti, L. Fogassi, and V. Gallese, “Parietal cortex: from sight to action,” *Curr Opin Neurobiol*, vol. 7, no. 4, pp. 562–567, 1997.
- [23] A. Pouget and L. H. Snyder, “Computational approaches to sensorimotor transformations,” *Nat Neurosci*, vol. 3 Suppl, pp. 1192–1198, 2000.
- [24] A. Pouget, P. Dayan, and R. Zemel, “Information processing with population codes,” *Nat Rev Neurosci*, vol. 1, no. 2, pp. 125–132, 2000.
- [25] T. Poggio and F. Girosi, “Networks for approximation and learning,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.
- [26] S. Deneve, P. E. Latham, and A. Pouget, “Reading population codes: a neural implementation of ideal observers,” *Nat Neurosci*, vol. 2, no. 8, pp. 740–745, 1999.
- [27] D. Rumelhart, G. Hinton, and R. Williams, “Learning internal representations by error propagation,” in *Parallel Distributed Processing*, chapter 8. MIT Press, 1986.