Chapter 5
Anticipatory, Goal-Directed Behavior

Martin V. Butz, Oliver Herbort, and Giovanni Pezzulo

> As Man is a reasonable Being, and is continually in Pursuit of Happiness, which he hopes to find in the Gratification of some Passion or Affection, he seldom acts or speaks or thinks without a Purpose and Intention. He has still some Object in View; and however improper the Means may sometimes be, which he causes for the Attainments of his End, he never loses View of an End, nor will he so much as throw away his Thoughts or Reflections, where he hopes not to reap any Satisfaction from them. (Hume, 1748, pg. 33-34)

David Hume may be one of the first who thought about the causes that actually enable us to act goal-directedly in our pursuit of happiness. Besides having usually an end, or goal, in mind, Hume realized that the end must elicit those means that were learned to *correlate* with the end. Such correlation knowledge, according to Hume (1748), was based on three types of connecting "ideas": resemblance, contiguity in time and place, and cause and effect. Knowledge of correlations and cause-effect relations alone, though, do not directly lead to effective behavior. Thus, not only the question how we learn correlations in the environment needs to be addressed, but also how we can exploit the obtained knowledge, if learned properly. While Hume did mainly address the former question, the latter question was acknowledged by Hume only in so far that the acquired knowledge may be used to pursue our goals.

Another related line of research on causation is put forward by Kant (1998). Sloman (2006) contrasted a 'Humean' and a 'Kantian' view of understanding correlations and causations in particular. The former is evidence-based, probabilistic, and statistical. The latter is structure-based and deterministic. Kant highlights the role of concepts and necessity in contrast with the Humean emphasis on observation and correlation. The Kantian notion of causation is more complex and requires understanding of spatial structures and relationships as well as the capability to reason about what happens when they change. While humans are usually seen as explorers that learn correlations in the world based on experimentation—and thus more 'Humean' evidence-based—they rely on and inevitably detect the 'Kantian' a priori structures in our world given due to time, space, and physical constraints.

Just like humans, all other anticipatory systems, both biological and artificial, need to learn and store knowledge about themselves and the world. Only this knowl-

G. Pezzulo et al. (Eds.): The Challenge of Anticipation, LNAI 5225, pp. 85–113, 2008.

edge enables them to predict future events, gauge the consequences of one's own actions, and finally interact competently and intelligently with objects or other agents. Thus, one of the questions addressed in this chapter is how knowledge about the world or the self may be represented.

However, regardless if we acquire knowledge based on 'Humean' or 'Kantian' principles, or both, just having gathered knowledge about the world does not mean per se that our decisions will be wise and our actions will be appropriate. It is not even clear how predictive knowledge may be turned into actual decisions in general. The *ideomotor principle*, which dates back to the 19-th century (James 1890; Herbart 1825; cf. Hoffmann et al. 2004), suggests that actions are bi-directionally linked to the effects they usually produce. Thus, once a goal is chosen and activated, the bi-directional links point to those actions that previously caused the goal to come about. While this still does not clarify the actual mechanism of selecting the appropriate means, it implies that an inverse mechanism is necessary that stores means to achieve current goals.

With the ideomotor principle as the basic principle of goal-directed behavior in mind, this chapter analyzes related predictive and anticipatory systems that learn predictive representations of their environment and can use those to act goal-directedly. Predictive systems are systems that are able to predict sensory inputs or pre-processed, more abstracted perceptual input. From an adaptive behavioral perspective most important are systems that *learn* such predictive representation. These predictive capabilities are an important part of any goal-directed behavioral system that is explicitly anticipatory. However, as suggested in the comparison of the insights put forward by David Hume and the ideomotor principle, predictive capabilities are only the first step toward an anticipatory behavioral system. Thus, the second question that this chapter addresses refers to the structures and processes that enable the selection of actions or decision making, based on the acquired knowledge.

We identify two fundamental classes of approaches that realize action selection based on predictive representations of sensory-motor correlations. First, *schemas* form a mental (internal) predictive world model, which encodes all kinds of properties, independent of possible tasks and goals. Although the representation might correspond to an exhaustive internal world model, the schemas alone cannot be used directly for decision making or action selection. Before a decision and action is made, internal processes are required that evaluate possible means in the light of current behavioral goals and desired states. Even more so, to be able to make complex decisions or execute meaningful actions, many schemas may have to be combined. Thus, schema approaches generally build a forward model that is used inversely for action selection.

Second, *inverse models*—in contrast to schema approaches—encode direct connections between behavioral goals and actions. Thus, they may be directly used for action decision making without any further processing. Inverse models can be seen as the result of abstracting or aggregating schemas because they focus on a generalized, inverted representation of properties in the world. In this sense, they can also be considered a world model, which is however rather limited compared to the world models realized in schema approaches.

Accordingly, this chapter first gives an overview of several kinds of schema approaches and inverse modeling approaches. We classify each system from an anticipatory behavior perspective discussing how knowledge is represented and which processes are necessary to turn anticipatory knowledge into behavior. As we cannot review all architectures that have been proposed to date, we exemplify each class of anticipatory behavioral systems with a representative model. Examples are chosen to provide details of state-of-the-art models of goal-directed behavior and to cover a broad range of approaches, including symbolic, subsymbolic, and neural models as well as supervised, unsupervised, and reinforcement learning approaches.

In the next section, we first give a brief history of schemas and provide a definition. We then distinguish different schema system classes and give system examples. Similarly, we discuss inverse model approaches, combinations of both approaches and other advanced techniques. In the second part of the chapter, we assess weaknesses and strengths of the architectures in learning and representing predictions and in using those predictions for the generation of anticipatory cognitive functions. Finally, we contrast the systems' capabilities and give an outlook on potential future macroscopic organizational structures for anticipatory systems, especially highlighting hierarchical and modular structures.

5.1 A Brief History of Schemas

Originally, psychology and cognitive sciences suggested that knowledge about our world is represented by schemas. Drescher (1991) concisely defined a schema as a representation of a triple that links a situation or condition, an action, which may be carried out in this situation, and subsequent effects. How the knowledge is turned into actual behavioral decision making and control remains unspecified. However, any schema representation may be considered as a structure that represents sensory-motor correlations, that is, how motor activity usually affects the perceived environment.

Historically, the term 'schema' may have been firstly introduced by Bartlett (1932) referring to a map or structure of knowledge stored in long-term memory. Successively, Piaget (1954) described schemas in a more operational sense, roughly as mental representations of some physical or mental action that can be performed on an object or event. He considered schemas the building blocks of thinking and as the basic structure underlying behavior and cognition (in a process that he described as 'assimilation and accommodation'). Schmidt (1975) proposed complex schema structures, which encode generalized motor programs for a variety of tasks and internal models of the sensory inputs that accompany movement execution.

Also many approaches in the field of artificial intelligence are based on the schema notion, including *frames* (Minsky, 1988), *scripts* (Schank and Abelson, 1977), *schemas* (Arbib, 1992, 1989; Drescher, 1991; Neisser, 1976; Norman and Shallice, 1986; Pezzulo and Calvi, 2007b; Shapiro and Schmidt, 1982), *anticipatory classifiers* (Butz, 2002a; Butz and Hoffmann, 2002; Gérard and Sigaud, 2001; Gérard et al., 2005; Stolzmann, 1998), *neural schemas* (Mccauley, 2002), *semiotic schemas* (Roy, 2005), and *behaviors* (Brooks, 1991; Maes, 1990). Architectures including distributed and competitive functional units are often referred to as ‘behavior-based’ or ‘schema-based’. Several integrated frameworks have been proposed for designing them; among the most popular ones we can mention the behavior-based approach proposed in Arkin (1998), the *NSL/ASL* in Weitzenfeld et al. (2000), and the *Robot Schema (RS)*—a formal language for designing robot controllers proposed in Lyons and Arbib (1989), which includes perceptual and motor schemas. Drescher (1991) was one of the first who implemented a functional schema-based approach showing simple goal-directed behavioral capabilities.

Schema theories are strongly motivated by biological and ethological models—some of the first implementations intended to replicate the behavior of the cockroach or the praying mantis in robots (Arkin et al., 2000). In general, a schema links conditions, actions, and result components, which are sometimes also called (sub-) schemas. This enables the control system to execute those actions whose conditions are currently satisfied (that is, schemas that apply) and whose result components appear currently desirable. Often, schema theories stress the importance of procedural knowledge, that is, a schema constitutes the long term memory of perceptual or motor skills, or the structure coordinating such skills. Schemas are especially well-suited for parallel and distributed systems, since they can be seen as concurrent computing units.

While several researchers have described the usage of schemas in the perspective of reactive and behavior-based robotics (Arkin, 1998; Brooks, 1991), schemas embed a predictive component that is used for action selection. Moreover, perceptual schemas are often shaped for motor actions. That is, schema representations are usually essential for motor actions. Thus, although not necessarily explicitly anticipatory, schemas serve for the control of behavior with a predictive component. Which schema representations exists and how these representations may be turned into behavior is discussed in the following section on the basis of various system examples.

5.2 Schema Approaches

Schemas integrate situations, actions, and their effects, mostly independent of upcoming tasks, potential goals, or any constraints. However, as these schemas are not related or specific to a certain task or goal, they cannot be directly used for decision making or action selection. For example, consider a schema that specifies that when holding a cup (condition) and drinking out of it (action), thirst will decline (effect) and another schema that specifies that when in a kitchen and grasping a cup, the cup will be held by the hand. Now, given the goal of wanting to quench one’s thirst, and given further the fact of being currently in the kitchen, then both schemas my be integrated suggesting that when grasping the cup and then drinking from it may quench the thirst. However, note that there is no schema that directly specifies what to do given the goal of quenching thirst, rather, little pieces of schema-based information need to be integrated into a more complex decision or action.

In the following sections we review several classes of schema approaches that differ in both, the form of the knowledge representation and the processing of the knowledge. We begin with a review of symbolic knowledge representations, from which actions may be derived by comparing expected results, planning, or the generation of a behavioral policy. Then, neural network approaches are discussed, which ground schemas on simple perceptions and derive actions from planning processes or the preparation of a controller by dynamic programming.

5.2.1 Symbolic Schemas for Policy Learning

An approach for using world model representations to improve policy learning and effectively generating an action policy is the *tabular DYNA-PI* model (Sutton, 1990), which may be considered as one of the simplest schema-based approaches. As in all reinforcement learning approaches, the core is an actor-critic architecture (Sutton and Barto, 1998). The critic implements an "evaluation function" and the actor an "action policy". The evaluation function assigns a reward or reinforcement value to each possible state-action pair. The action policy determines which action to take in a specific state. In addition to this actor-critic model, DYNA-PI learns a predictive world model. This model is composed of two functions, a state transition function and a reward function (both functions may be stochastic). Both functions are learned by initially random interactions with the environment. The world model is used to predict the consequences of actions, in terms of reward and future states of the world. The action policy, due to a reward-backpropagation mechanism, realizes the inversion process. Behavior is triggered, by choosing that action that is expected to yield the highest reward in the long run, which is effectively a form of payoff anticipation.

The key idea of the DYNA-PI model is that an agent endowed with a world model can produce "simulated experiences", besides the experiences gathered during actual environment interactions. Thus, the evaluator and actor can be further trained on simulated experiences. If the learned world model is accurate enough, this "mental training" will speed-up the improvement of behavioral performance in the real world. Thus, besides payoff anticipations, DYNA-PI uses internal simulations of anticipated events to improve its behavior—a form of state anticipation.

Reinforcement learning approaches were recently also carried-over to logic-based representations, in which case they are often referred to as relational reinforcement learning. Kersting et al. (2004) applied reinforcement learning ideas to a logic-based, relational world model framework. Using reward propagation techniques and a matcher mechanism, desired goal states were activated and propagated through the logic-based relational world model. The first-order logic-based abstractions in the world model showed to improve behavior and planning capabilities significantly, also enabling generalizations to similar contexts dependent on the relational logic-based representation available. Thus, besides tabular representations, generalized relational schema-like representations can be applied effectively combining world model representations with policy learning. Another challenge, however, is to learn a suitable generalized schema representation from experience alone, which is addressed in the subsequent sections.

5.2.2 Symbolic Schemas and Prediction for Selection

A prominent example of an online generalizing world model learner are Anticipatory Learning Classifier Systems (ALCSs, Stolzmann, 1998). These learning systems are inspired by the psychological principle of anticipatory behavior control (Hoffmann, 1993; Hoffmann et al., 2004), as well as by the schema approach of Drescher (1991) and DYNA-PI. ALCSs learn a *generalized* predictive model of an environment online. Predictive knowledge is stored in condition-action-effect rules, called classifiers, that represent a schema-based world model. The ACS2 system (Butz, 2002a) combines heuristic search with genetic mechanisms to generalize the predictive world model online.

As DYNA-PI, ACS (Stolzmann, 1998) originally included reward values directly in the schema representations. Given a generalized schema representation, however, reward aliasing can occur in which case the schemas may be sufficiently accurate to predict action effects but may be over-general to represent an optimal behavioral policy (Butz, 2002a). Consequently, XACS (Butz and Goldberg, 2003) was developed, which separates state value and schema learning. XACS is a combination of ACS2's model learning capabilities with the evolutionary online generalizing RL mechanism XCS (Wilson, 1995). The system learns online a generalized state value function, which is represented by a set of condition-value tuples, using XCS-based techniques. Moreover it learns a generalized world model according to the model learning techniques of ACS2. In reinforcement learning terms, XACS learns generalized representations of the state transition function of a Markov decision process (MDP) as well as of the underlying value function.

As opposed to selecting an action based on the best applicable schema, action selection then becomes a two-stage process in which all applicable schemas predict possible next outcomes and the schema is chosen for execution that predicts the maximally suitable outcome, that is, the outcome that is expected to yield the highest value according to the learned state value evaluation function. Thus, the inversion of the predictive capabilities takes place during action selection as well as by means of reward back-propagation mechanisms while learning the value function. More complex decisions or behaviors may be elicited if planning mechanisms are used to combine many schemas.

XACS has shown to be able to robustly learn compact representations of optimal behavioral policies. Policy learning was further sped-up by exploiting the knowledge of the predictive model using the DYNA-based update techniques discussed above (effectively speeding-up the adaptation of the value function). Thus, XACS is a schema system that combines a schema representation with a state value representation to learn a compactly represented optimal behavioral policy quickly, accurately, and reliably.

Recent gradient-based update mechanisms in XCS (Butz, 2006) can improve performance of XACS, so that XACS promises to serve as a robust learner in large, high dimensional MDP problems. With respect to behavioral plausibility, it was shown that ACS2 can be used to simulate the learning of behavioral patterns previously observed in rats (Butz and Hoffmann, 2002). Moreover, since XACS is a system that learns online and from scratch, the implementation of an enhanced XACS system is possible, which may comprise multiple, interacting reward learning modules that may be additionally controlled by motivational and emotional constraints. For example, dependent on the gained learning experiences, it is imaginable that the emotional patterns of the cognitive system may evolve differently resulting in, for example, a very “shy” or a very “bold” system.

5.2.3 Neural-Based Planning

Besides tabular and symbolic approaches, also a neural network-based schema approach (Baldassarre, 2001, 2003, 2002a,b) was implemented, which exploited the prediction and planning capabilities of the schema-based representation. The controller was tested on a simulated robot with a 1D surround camera that solves stochastic path-finding landmark navigation tasks (the robot moves in an arena with white walls and black pillar landmarks by selecting one of eight absolute-direction actions in each simulation time step). Unlike the DYNA-PI architectures, the controller can pursue arbitrary (novel) goals. In particular, the NN can plan with respect to the achievement of any externally or internally generated goal, thanks to the generation of internal rewards in association with them. Whereas DYNA learns to predict rewards assumed to be permanently associated to states, the NN planner is endowed with a “reward generator”, which dynamically generates an internal reward when the system achieves its current goal.

The controller builds an efficient “partial policy” by focusing on possible start-goal paths and is capable of deciding to re-plan if “unexpected” states are encountered (Baldassarre, 2003). The simple “forward planner” version of the controller iteratively plans by the generation of chains of predictions from the position currently occupied by the robot. The more sophisticated “backward-forward planner” version of the controller iteratively generates chains of predictions from both the position currently occupied by the robot and the goal state. In both cases, the pseudo-experience so generated is used to train the reactive components of the system as in the DYNA systems. The forward models are composed of neural networks trained to predict the perceptual consequence of action executions. The “backward models” are composed of neural networks trained to “predict” the “origin state” from which the robot might have arrived to a certain state given the execution of a certain action.

Another version of the controller implements a simple form of neural abstract planning that enhances the exploration and evaluation updating capabilities of the controller (Baldassarre, 2001). Abstraction is implemented in terms of planning on the basis of macro-actions (actions composed of *n* actions of the same type, such as, north-north-north) and action execution at the primitive level.

A more sophisticated modular version of the controller (Baldassarre, 2002a) allows the system to store information about achieved goals and to recall such information so as to decrease the planning burden when the same goals are assigned more than one time. In this case, the goals are not only used to plan but also to satisfy a “motivation” signal that allows the reactive components of the system to recall the knowledge related to previously achieved goals.

An earlier, similar NN planner (Schmidhuber, 1990b,a, 1991c) learned a recurrent NN model and could show capabilities of reinforcement learning and planning in dynamic environments. He also investigated the capabilities of simulating curiosity and boredom with the architecture. Recently, parts of that NN planner were used in some of the modules of a modular and hierarchical control architecture (Gloye et al., 2004) that won the robot soccer world cup (FU-Fighters, Small Size, 2004).

5.2.4 Neural Network-Based Dynamic Programming

Finally, neural networks may be used to integrate schemas into highly flexible movement plans by neurally implementing dynamic programming. A recent computational model of motor learning and control, SURE_REACH, explains the high flexibility of human motor behavior (Butz et al., 2007a). This hierarchical architecture stores an associative model of state transitions as well as a redundant associative mapping of hand locations with arm postures. Population-encoded spatial representations enable the application of dynamic programming techniques. To move the hand to a desired location, the hand position is first translated into a representation of the redundant postures that coincide with the target hand position. This redundant intrinsically encoded goal representations and the encoded state transition model is then used to generate a movement plan by neurally implemented dynamic programming.

Without additional constraints, the minimum path in posture space is executed. However, if the task imposes additional constraints, alternative action sequences may be generated by simple neural inhibitions. Thus, SURE_REACH is able to reach hand targets while incorporating task-specific constraints, for example, adhering to kinematic constraints, anticipating the demands of subsequent movements, avoiding obstacles, or reducing the motion of impaired joints (Butz et al., 2007a; Herbort and Butz, 2007). The approach is generally similar to early self-supervised control approaches (Kuperstein, 1988; Mel, 1991), but extends them to the sensorimotor control of redundant bodies. Compared to previous neural network models of motor learning and control, SURE_REACH accounts for higher behavioral flexibility and adaptivity without the need for relearning.

5.3 Inverse Model Approaches

Schema approaches may be used to represent a model of the world in a very fragmented way and they require complex processes to turn a goal into an action. A different approach of modeling goal-directed behavior and the function of executive modules is put forward by the notion of the inverse model (Kawato, 1999). An inverse model is an internal representation that inverts the flow from action to effect. It thus generates actions that are useful to reach a desired state. To follow the example mentioned above, an inverse model might specify that thirst may be quenched by drinking from a cup held in the hand and that when in the kitchen without a cup, a cup should be grasped. Note that in this example, the model directly specifies which action to execute given start and goal. The model does not specify the actual consequences of actions, though. Rather, it merely suggests that the action in the given circumstances is usually advantageous for achieving the specified goal.

Thus, the inverse model approach is fundamentally different from the schema approach. Whereas in the schema approach, the fragments of information stored in the schema have to be processed to arrive at a decision or to generate an action, an inverse model aggregates such a process in a direct mapping from situations and goals to actions. An inverse model may thus be seen as the result of an aggregation of many executed schema processes that are combined and generalized into a simple mapping. A drawback of inverse model approaches is that the acquired mapping is highly inflexible because it generates a rigid mapping from goals to actions. Thus, if the environment changes or novel tasks have to be solved, alternative behaviors may be required to maintain effective behavior. Inverse models cannot provide alternatives so that expensive relearning would be necessary without schema knowledge. Of course, a direct mapping has the advantage that no potentially costly planning or other preparatory processes are necessary to determine actions. Thus, while inverse models appear well-suited for rigid, quick, automatized control, inverse models alone are rather inflexible and essentially may hinder the quick adaptation to novel situations or tasks.

5.3.1 Inverse Models in Motor Learning and Control

In computational neuroscience, inverse model approaches are implemented in feedback error learning (FEL) models of cerebral motor learning (Berthier et al., 1992, 1993; Barto et al., 1999; Haruno et al., 2001; Karniel and Inbar, 1997; Kawato et al., 1987; Kawato and Gomi, 1992; Schweighofer et al., 1998b,a; Wolpert and Kawato, 1998) In short, these model predicate that the cerebellum is an inverse model for goal-directed motor behavior. The cerebellum exerts control of goal-directed movements and adjusts its output according to an assumed cerebral linear feedback controller. During learning, the cerebellum thus learns a direct mapping from goals to motor outputs.

While FEL models rely on the accurate control of a simple controller, other inverse model paradigms learn their inverse models simply by the observation of randomly sampled actions or physical plant correlations. The most prominent class in these approaches are *direct inverse modeling* (DIM, Baraduc et al., 1999, 2001; Bullock et al., 1993; Kuperstein, 1988, 1991; Ognibene et al., 2006) and the related *resolved motion rate control* (RMRC) approaches (D'Souza et al., 2001; Jordan and Rumelhart, 1992; Whitney, 1969). Both techniques learn a situation-dependent mapping between goals and motor commands. For example, a non-redundant arm may learn its inverse kinematics by mapping a hand position goal to a corresponding arm posture, which may trigger suitable motor activity. RMRC is more robust in the face of redundant plants, storing that action for a particular goal and situation combination that was optimal during learning.

Redundant bodies or environments generally pose a problem to inverse modeling approaches, because one of many equivalent actions has to be associated with each particular goal. Thus, among all potential actions, those are stored in the inverse model that optimize additional criteria. These optimality criteria have to be defined to enable the acquisition of an inverse model (D'Souza et al., 2001; Engelbrecht, 2001; Todorov, 2004). An inverse model is thus only suited to optimize a single criterion, which was defined before learning. Changes in the criterion, for example, due to demands of novel tasks or changes in the environment, reduce the performance of an inverse model or may even render it completely incapable. In contrast, only the ability to adapt optimality criteria quickly from one movement to the next enables the flexibility of human behavior (Rosenbaum et al., 1995). Additionally, the need to adapt an inverse model to an optimality criterion seems to hinder unsupervised sensorimotor learning (Herbort and Butz, 2007). Thus, due to its inflexibility and limited learning capability, the inverse-model view of motor control has recently been challenged with the proposition of the SURE_REACH model (Butz et al., 2007a).

5.3.2 Inverse Models and Schema Approaches

Despite the principled difference that schema approaches store a general model of the world and inverse models encode preprocessed, task-specific goal-action links, both approaches are certainly strongly related.

First, inverse models and schema approaches may happen to represent identical sets of information, if a one-to-one mapping between goals and actions exists (of course, dependent on the situation). In such a context, each goal can only be pursued by a single action and executing this action is sufficient to reach that goal. Thus, inverse models and schema models inevitably represent the same information given that both models always yield the same action to execute. However, the general equivalence may only exist in rather abstract, artificial models, seeing that environments are usually continuously in flux.

Second, some schema approaches first prepare a behavioral policy, dependent on the goal and potential constraints, and then execute behavior accordingly (e.g., SURE_REACH). The resulting policy can be considered an ad-hoc inverse model, which has been generated solely and exclusively for the current goal and situational demands. In this sense, these approaches combine the advantages of inverse models with the flexibility of schema approaches.

Thus, it seems most plausible that efficient anticipatory behavioral control can only be accomplished with both representations present—schema approaches to know the environment and also to verify current action successes and inverse model approaches to effectively and progressively automatically control behavior.

5.4 Advanced Structures

The previous section has described a broad variety of approaches, which implement executive models and enable goal-directed behavior. In the following, we outline how the described approaches may be integrated and combined with predictive models to enhance performance and address more complex tasks.

It is evident from many lines of research in psychology, neuroscience, computer science, and engineering that efficient behavior is not only based on the quality of schemas or inverse models, but also on the quality of sensory data or the quality of the output processing. For example, sensory ambiguity may be reduce by integrating multiple sources of information or by predictive top-down connections. Likewise, motor control may be facilitated by being able to identify basic characteristics of plants or by dividing the generation of motor actions from high-level goals into computationally simpler sub-processes.

In this section, we want to highlight structures, in which schema approaches or inverse models may be combined or embedded to optimize behavioral control. First, we show that the combination of executive modules and predictive mechanisms can enhance behavioral performance. Second, predictive models and executive modules may be coupled to form higher order schemas, enabling effective behavior in varying contexts. Finally, hierarchical control structures may stabilize behavior and enable the solution of more complex types of problems.

5.4.1 Prediction and Action

The discussed inverse model approaches are capable of generating actions or behaviors to pursue certain goals. In this section, we discuss how these architectures may be integrated with predictive models, that is, forward models of schema approaches, to enhance control. Forward models enable the anticipation of changes of the environment or effects of one's own actions. In this section, we first introduce long short-term memory (LSTM) recurrent neural networks, which allow the prediction of a series of future events. Then, we give examples in which forward models provide internal feedback to stabilize and enhance control.

5.4.1.1 Recurrent Neural Network Approaches

Recurrent neural networks (RNNs) were proposed in Elman (1990) mainly as a language and grammar processing system. However, recent advances have applied RNNs to a variety of problems including time series analysis, speech processing, or robot navigation tasks. RNNs seem to have particularly strong potential for the formation of predictive and anticipatory structures. A good overview of a variety of RNNs can be found in Zappacosta et al. (2007). In the following, we focus on the LSTM system, which solves particularly hard grammatical problems as well as challenging time series analyses problems [1].

LSTM models are artificial RNN architectures that are endowed with neural gate-based structures (Hochreiter and Schmidhuber, 1997). Input gates and output gates guard input/output access to the internal states of neurons, enabling the algorithm to maintain memory over theoretically infinitely long periods of time. The networks effectively deal with the problem of vanishing gradients, which is usually a major

[1] Some of the other neural network approaches discussed below as well as the NN approaches discussed in relation to schema-based approaches also contain some form of recurrence but are usually not explicitly termed "recurrent" NNs.

limitation in other RNN structures especially in problems in which long-term dependencies need to be remembered. LSTM can remember and relate events distant in time. It is thus expected to be most suitable as a prediction tool for anticipatory systems that need to detect long-term dependencies (in memory) or that have to deal with partially observable MDPs (POMDPs).

LSTM RNNs use "memory units" that use the "constant error carousel" (CEC) to propagate error, theoretically, infinitely back in time. The memory units are protected by an input gate and an output gate that multiply incoming activation and outgoing activation to effectively "gate" the memory information so that it can apply only when necessary. In later papers, a "forget-unit" was added that makes the timing of the memory unit more precise and allows learning from continuous data streams (Gers et al., 2000). Kalman-filter enhanced learning (Pérez-Ortiz et al., 2003) increases learning speed by orders of magnitude but also increases the learning complexity of the system.

While LSTM RNNs thus showed to be highly useful in predicting challenging context-free grammars as well as real-valued data streams, it remains to be shown how the learned structures may be inverted to trigger effective online action selection and motor control.

5.4.1.2 Internal Feedback

After given an example of a predictive model, we now explain how such models may be used to improve behavior. In order to efficiently select effective actions, it is essential to know the current state of the world as exactly as possible. However, biological and artificial sensor systems are prone to noise, and information about the environment may only reach the executive modules through time-consuming processing pathways. In this case, a predictive model may compensate for these effects. For example, if sensor data is noisy or ambiguous, a prediction of the current state of the environment based on recent perceptions and recently executed actions may enable the formation of a concise representation of the environment. Thus, forward models may help to implement Kalman filtering approaches. Likewise, decisions or actions may be adjusted according to their expected impact on the world before feedback from the environment is available or even before the action is executed.

A processing control model, partially based on internal feedback, was proposed by Kawato et al. (1987), who applied it to arm movements. In the model, control is initially exerted by a linear feedback controller. The controller is not very efficient, though, because the delayed feedback results in a slow control process. A forward model is gradually learned and improves the control process by providing internal feedback, thus enabling a much faster control process. Finally, an inverse model replaces the linear controller to enable maximally effective movements.

In this setup, the forward model provides almost instant feedback about the expected consequences of the generated motor command. Thus, the motor system does not have to wait for external feedback to adjust its output but can adjust motor commands on their expected effects. As a result, the motor commands are much more accurate and movement dexterity increases. Later on, internal feedback can still level out small inaccuracies of the inverse model and compensate for forward model estimation errors.

Several recent studies have suggested that forward models, as body emulators, are essential for efficient body control (Kawato, 1999; Wolpert et al., 2001). Moreover, various studies suggest that internal forward model feedback is used to estimate spatial body location (Wolpert et al., 1995) as well as to improve behavioral control of fast reaching movement (Desmurget and Grafton, 2000) or of pole balancing (Mehta and Schaal, 2002). Grush (2004) relates such representation also to higher level cognitive processes. Thus, internal, anticipated sensory feedback appears to play an important role in behavioral control, state estimation, as well as higher cognition.

5.4.2 Coupled Forward-Inverse Models

In the review of schema approaches, the condition part of a schema mostly referred to a specific sensory state or situation, whereas the action part referred to a single action entity. However, it is also possible to form schemas from more complex notions of perception and action. Forward-inverse models directly couple forward and inverse model information essentially representing context in current forward model accuracy.

Adaptive agents should be able to operate in different contexts or environments and should quickly adjust to changes. Thus, it has been proposed that multiple executive modules, schema approaches or inverse models, may be represented for different contexts. However, this requires the quick identification of contexts to enable the selection of appropriate modules. Predictive models may play a key role to identify contexts and participate in the selection of the right executive module for the right task.

Several researchers have proposed such decentralized architectural schemes for the control of action that are based on coupling forward and inverse models, both, in a localized (Demiris and Khadhouri, 2005; Pezzulo and Calvi, 2006a; Tani and Nolfi, 1999; Wolpert and Kawato, 1998) and in a distributed fashion (Tani, 2003; Tani et al., 2004). The former approaches are based on the mixture of experts architecture (Jacobs et al., 1991) while the latter are based on the self-organization of the representational space in RNNs.

In these approaches, forward models are coupled to executive modules (that is, some form of inverse model), representing a higher level schema. Such a schema is applicable if the predictive model makes continuously accurate predictions. The condition part of a schema is a forward model, which enables the identification of the underlying properties of a situation, for example, how objects or bodies react to certain actions. These underlying properties may not be evident from regarding a single instance of the perceptual input. Likewise, the action part of a high level schema might not be a single motor command but an entire controller, which is especially suited to exert control in a specific context. Thus, the accuracy of the predictive model indicates the suitability of the executive module in each schema. As predictive model and executive module of a schema are trained exclusively in parallel and in an assigned context, the predictive model will only be able to make valid predictions in the context, for which also the executive module was trained.

These architectural schemes have been used for multiple purposes. For example, they were used to select actions appropriate to the context (Wolpert and Kawato, 1998), they were used for action observation and execution (Demiris and Khadhouri, 2005), and they were combined with a motivational system in which active drives influence action selection (Pezzulo and Calvi, 2006a) . When behaviors can be combined linearly, the models can also generalize behaviors. Algorithms for learning and combining contexts in non-linear dynamics have also been proposed (Vijayakumar et al., 2005).

This integration of predictive models and executive modules into a schema may help stabilize the selection of control strategies, even in noisy contexts. Furthermore, it is possible to deduce abstract properties of a situation, which may not be directly concluded from sensory input. A drawback can be that the difficulty between learning the inverse model and the forward model may differ significantly, so that the current accuracy of the forward model may not necessarily reflect the current suitability of the coupled inverse model.

5.4.3 Hierarchical Anticipatory Systems

Each of the so-far presented approaches is limited to comparatively small problem domain, such as the control of simple movements, planning a chain of actions, maze navigation, or the prediction of events of a certain kind. Whereas each approach may be well suited to solve the problems in its domain, it cannot be easily extended to the high variety of tasks that humans or autonomous artificial systems face, ranging from the need to determine long-term goals to the accurate control of basic actuators. This limitations can be overcome by integrating the described approaches into a hierarchical framework. Many neurological and psychological studies and models suggest that effective cognition and consequent behavior is based on hierarchically structured systems, for example, accounting for complex sensory processing, cognition, and behavior (Giese and Poggio, 2003; Koechlin and Summerfield, 2007; Loeb et al., 1999; Poggio and Bizzi, 2004; Powers, 1973; Riesenhuber and Poggio, 1999; Todorov, 2004; Wolpert et al., 2003).

By structuring a cognitive architecture, separate problems in tasks like sensory processing or motor control may be solved by different modules. For example, it was demonstrated that the spinal circuity is able to counteract some perturbations on its own, thus making motor control easier for the central nervous system (Loeb et al., 1999). Accordingly, the CNS provides a basic control strategy, for example, by setting reflex gains or muscle stiffness. This enables the CNS to flexibly adjust motor control to varying tasks without the necessity to react to small perturbation, whose compensation is left to the spinal system, which is well suited for this task due to its fast feedback control loops. Models of central motor control suggest that the cerebellum replaces and further optimizes cortical motor networks in the control of frequent, well-trained movements (Berthier et al., 1993; Barto et al., 1999; Kawato et al., 1987; Kawato and Gomi, 1992; Schweighofer et al., 1998b,a). Furthermore, also movement specifications may be based on incorporating movement selection biases on multiple levels (Cisek, 2006; Herbort et al., 2007). Also, current hierarchical models of vision (Riesenhuber and Poggio, 1999; Giese and Poggio, 2003) were suggested to be extended to motor control problems Poggio and Bizzi (2004).

In conclusion, hierarchical, modular systems can address specific computations in specific modules, consequently reducing the complexity of each computation and enhancing stability due to the partial autonomy of each module. However, in most approaches information flows in a single direction, for example, from visual sensors to abstract representations or from behavioral goals to movements. Systems in which layers bidirectionally influence each other seem to be more promising for the understanding of complex perception, cognition, and behavior (Hinton et al., 1995; Hinton, 2007; Rao and Ballard, 1997). Higher layers may try to model the behavior of the lower layers, correcting lower layer states when the lower layers do not have the knowledge of predicting their own state. In other words, higher layers may correct the state of lower layers by, for example, resolving ambiguity. Uncertainty measures of each module's state and also attentional influence may further modify the influence layers have on each other. In sum, combinations and integrations of effective sensory processing and motor control modules promise to yield highly flexible adaptive decision making and control structures that go far beyond the competency of a flat architecture.

In the following section, we now first evaluate the predictive and anticipatory capabilities of each considered system and then discuss correlations and contrast the distinct features of each system. In particular, we first list predictive and anticipatory capability criteria. Next, we discuss the various schema approaches and inverse model approaches with respect to these criteria. The subsequent discussion draws the attention to the currently missing system capabilities and proposes various future research options.

5.5 Evaluation of Predictive and Anticipatory Capabilities

We now evaluate the predictive and anticipatory capabilities of the introduced approaches on the basis of the taxonomies of predictive and anticipatory capabilities introduced in Chapter 2. To do so, we consider the capabilities of each system individually and finally discuss their correlations and differences as well as potential complementarities.
More concretely, we distinguish and discuss the following predictive qualities:

- *Symbolic vs. real valued predictions*: Does the system form symbolic, real-valued, or both types of predictions?
- *Discrete vs. continuous predictions*: Does the system form predictions about a discrete next time step or can it form continuous predictions over time?
- *Noise robustness*: Are learning and representation of predictions noise robust?

- *Sensory vs. payoff predictions*: Does the system form sensory predictions or payoff predictions?
- *Single vs. multiple predictions in representation spaces*: Does the system form one single prediction or multiple predictions (e.g. concrete and abstract)?
- *Full vs. partial predictions*: Does the system attempt to predict complete future states or is it able to focus on sub-states?
- *Exact vs. fuzzy, distributed predictions*: Does the system form on exact predictive representation or does it also estimate the confidence of its predictions?
- *Immediate vs. long term predictions*: Does the system predict only next states or is it able to form immediate long-term predictions (without chaining immediate predictions)?
- *Generalization capabilities in predictions*: Is the system able to generalize its predictions to similar states in the environment?
- *Single vs. multiple sources of information*: Does the system distinguish between multiple sources of information such as sensory, context, and motor activity information?
- *Markov-dependent vs. independent predictions (MDP vs. POMDP)*: Does the system rely on full observability to be able to reliably learn accurate predictions?
- *Distinction between self and other*: Does the system distinguish between predictive representations about own future states and other future states?

With respect to anticipatory capabilities, we distinguish the following qualities:

- *Direct vs. indirect inversion*: Are goals directly (inversely) mapped to actions, or is the mapping done indirectly?
- *Reward vs. plan-based inversion*: Is the inversion accomplished by means of back-propagated payoff representations or by means of explicit representations of future states?
- *Planning capabilities*: Is the system able to plan?
- *Full vs. partial planning*: Can the system also generate partial, abstract plans?
- *Online vs. offline representations*: Is the system bounded to generate future representations based on the current state or can is also generate anticipated representations offline?
- *Flexible goal-oriented behavior*: Can the system flexibly pursue novel goals?
- *Adjust to new task constraints*: Can the system immediately adjust behavior to novel task constraints?
- *Curious behavior*: Can the system act curiously, directedly exploring novel territory or environmental properties?
- *Epistemic actions*: Can the system act pro-actively in order to search for missing information?
- *Surprise mechanisms*: Can the system implement surprise mechanisms upon unexpected perceptions?
- *Motivational goals*: Can goals be chosen or pursued based on the system's current motivations?

With these distinctions of predictive and anticipatory system capabilities in mind, we now evaluate the considered systems.

5.5.1 Schema-Based Systems

5.5.1.1 Predictive Capabilities

Schema-based systems exist for symbolic and for real-valued representations as well as for discrete and continuous predictive representations. Generally, schema systems focus on sensory predictions and, dependent on the schema representation, this prediction can comprise multiple predictions in space as well as in time. Also, partial predictions are generally possible and the predictions often contain confidence estimates and thus represent fuzzy predictions. A tenet of schema approaches is that condition structures try to focus on those sources of information that are maximally suitable to generate representations of future states. Thus, schema systems are able to generalize, dependent on the representation and learning mechanisms employed. Multiple sources of information are usually considered—albeit not necessarily processed by different modules. Except for successful applications in deterministic POMDP problems (Holmes and Isbell, 2005; Landau et al., 2003; Métivier and Lattaud, 2003; Stolzmann, 2000), the successful development of internal states for the solution of stochastic POMDP problems still remains an open challenge. Finally, schema representations may also be projected, or mirrored, onto other entities in the environment, representing their sensory-motor correlations and internal states. To the best of our knowledge, currently no system exists that accomplishes such a task autonomously.

DYNA-PI More concretely, the tabular DYNA-PI model may be considered the most restricted system amongst the considered schema-based systems. It is able to work only on discrete, distinct, and symbolic state representations and generally cannot be consider noise robust. The prediction of the next state may be a distribution over states but originally was also restricted to exact next states. The original DYNA-PI approach does not contain any generalizations, either, which clearly poses a huge scalability problem. The table essentially grows linearly in the number of distinguishable states and actions. Provided diverse real-valued sensory input, the number of states may be infinite, which is an obvious limitation of tabular approaches. Thus, tabular DYNA-PI is only suitable to investigate internal planning and reinforcement propagation mechanisms rather than to actually apply the system to real-world adaptive system control tasks. More recent schema-based approaches tackle this limitation with various generalization mechanisms.

XACS The XACS system is a purely symbolic, discrete prediction learner. It combines model and RL learning to learn sensory predictions and payoff predictions. The sensory predictive module relies on determinism in the environment but can ignore fluctuations of irrelevant sensors (Butz, 2002a). The reward learning part is rather noise robust (Butz et al., 2004a, 2003a).

Besides concrete predictions, XACS also provides the certainty of its state predictions and the corresponding state value predictions. The predictions are generalized in that irrelevant attributes can be ignored. The predictive model is usually significantly smaller than a completely specified model. Currently, the architecture is flat without hierarchies. Irrelevant attributes are ignored and can be explicitly identified as irrelevant for accurate predictions or as unpredictable.

Experimental evaluations have shown that the system can ignore irrelevant attributes and, given irrelevant attributes, it beats learners that learn a tabular problem representation. The predictive models are always generalized and are usually much more compact than tabular approaches.

There are no hierarchies and predictions are currently on one time scale only. However, predictive chains can be generated so that long-term predictions are possible in a limited sense. The generalization mechanisms in XACS focuses on the attributes that are relevant for accurate predictions of the next sensory inputs and the consequent reward, respectively. Thus, regularities are detected and object clusters are expected to be identifiable by the mechanisms. More sophisticated actions such as hierarchical option-type actions (Barto and Mahadevan, 2003) or motor programs have not been investigated so far. Also contextual information, except for action codes, has not been treated separately from pure sensory inputs in any way.

Sequence learning capabilities or performance in POMDP problems were not investigated with XACS so far. Currently, the predictive learning capabilities are restricted to MDP problems, because no internal states are used. Since actions are directly included in the classifier structure, discrete action codes currently need to be used currently and encountered changes are related to one's own actions only.

Neural Based Planning Whereas the original tabular DYNA-PI architecture is a tabular lookup system, which operates on discrete symbolic inputs, the NN-based architecture is a continuous predictive system, which predicts feature-like sensory inputs or even continuous sensory changes. The NN-based architecture can be considered rather noise robust. It forms stochastic predictions. However, the forward models of the implemented architecture currently produce deterministic predictions of the full sensory input. It cannot ignore irrelevant or unpredictable input, nor can it focus on the prediction of partial input. Confidence values and fuzzy predictions may be employed by using an appropriate error functions and related weight update mechanism. As in DYNA-PI and XACS, the NN-based schema architecture represents concrete predictions in terms of sensory input and the architecture is flat enabling only immediate predictions (but see Baldassarre 2003, for a preliminary investigation of a NN planner whose predictions span further time steps ahead).

The NN-based system can generalize to similar events and similar sequences, but it does not develop object-oriented representations (no clustering) nor any type of grammar representation (no recurrence), except for potential emergent representations in the hidden layer of the NN due to back-propagation. Thus, only the back-propagation algorithm in the implemented NNs may be able to ignore sensory inputs.

The system distinguishes between two sources of information, sensory inputs, which are continuous, and action inputs, which are discrete and are used to select expert forward models dedicated to them. If continuous actions were included, then actions should be considered as an additional input or may be coded in a population code fashion. So far, there were no investigations that would include other context-based information or that would learn about environmental dynamics that occur independent of the system's own actions.

As there are no internal state representations, the neural-based planner relies on the Markov property. Moreover, the system is completely self-centered being currently unable to project one's own predictive knowledge onto other entities in the world.

SURE_REACH Devised as a model for movement generation rather than prediction, SURE_REACH does not explicitly implement any predictive mechanisms. Nevertheless, the neural networks that are used for control are also suitable to predict future states, the developed sensorimotor model is purely associative. The kinematic mapping may be used to predict hand locations from (predicted) postures and vice versa. Likewise, the sensorimotor model may be used to predict the trajectory that results from issuing a sequence of motor commands. Due to the population encoding of these internal models, also the uncertainty of predictions may be represented.

In sum, the system can be used to form real-valued predictions on a continuous time scale, predicting perceptions. It can be considered rather noise-robust and may even form, albeit limited, multiple predictive representations (arm postures and hand locations). These predictions are full but, due to the population encoding, inherently represent fuzzy distributions of states. Besides the option to chain predictions, as in all other schema systems, the predictions are immediate and restricted to one time scale. As the other approaches, it also integrates action and sensory information for prediction. SURE_REACH does not address partial observability nor the challenge to form distinct representations of self and other.

5.5.1.2 Anticipatory Capabilities

The schema-based systems have several anticipatory capabilities in common—most obvious is the fact that all rely on indirect inversions to trigger anticipatory behavior. That is, they need to use their predictive capabilities in some way to trigger goal-oriented behavior. This inversion mechanism is sometimes purely reward-based, sometimes plan-based and sometimes a combination of both. Moreover, all schema systems have the possibility to plan, albeit usually generating full plans only. Also the offline generation of mental representations is usually possible. Dependent on the involved inversion mechanisms, goal-oriented behavior can be more or less flexibly adjusted. Curious behavior is generally implementable, however, epistemic actions as well as surprise mechanisms require further additions. Motivational goals may also be included in each of the systems.

DYNA-PI DYNA-PI models are able to form explicit predictions online and offline. Thus, situation-dependent planning as well as offline reflections are possible and also implemented. DYNA-PI models are generally goal-oriented anticipatory system. However, DYNA-PI models do not have the flexibility to account for novel goals without any significant re-planning effort. That is, DYNA-PI models usually learn one (or a couple of alternative) behavioral policies. If none of the policies is currently suitable because, for example, the goal is relocated to a novel location or multiple goals are suddenly distributed over the environment, complete re-planning is often necessary. System behavior is not directly goal-initiated but goal-oriented, due to the learning of a behavioral policy. Thus, it is the reward inversion that results in the generation of goal-oriented behavior. Curious behavior has been implemented in some DYNA-based approaches, however, epistemic actions as well as surprise mechanisms remain to be further investigated. DYNA-PI may learn policies for several distinct goal representations so that it may choose between the pursuance of available goals based on the system's current motivational state.

XACS The anticipatory capabilities of XACS are generally similar to those of DYNA-PI with the advantage that the system is able to learn generalized policies in partially noisy environments that contain many additional, irrelevant sensory inputs. Moreover, XACS does not learn a pure policy representation and thus does not rely purely on reward inversion but also incorporates a plan-based inversion. That is, anticipated next states and their associated state values trigger action decisions. Partial planning is possible, but it is dependent on the chosen representation. However, the learned state value function cannot be changed without significant relearning effort. As the DYNA-PI model, the XACS model lacks the anticipatory flexibility to account for any possible goal, but it learns to adapt only to the goals, that is, the rewards, encountered during learning.

The behavioral policy can be improved to cause curious behavior (improving model and policy learning) as well as greedy behavioral patterns (improving and speeding-up policy learning, Butz, 2002b). The usage of a list of currently least accurate predictions combined in a priority list, similar to the work done by Moore and Atkeson (1993) in their prioritized sweeping mechanism, showed to improve behavioral learning even further (Butz and Goldberg, 2003). As in DYNA-PI, though, neither epistemic actions nor surprise mechanisms were further investigated so far. Currently only one RL module was implemented but different RL modules for different motivations are easily interpretable and combinable in the XACS framework. Thus, the system has strong potentials to study multiple motivational influences and emotional integrations. For example, opportunistic behavior may be triggered by combining current motivational utility measures with current predictions.

In sum, the XACS approach has shown that the combination of learning generalized representations of both a predictive model and a state-value representation is a highly suitable approach that yields effective anticipatory action decision making and control. However, as in the DYNA-PI approach, without additional mechanisms or representations, the system cannot adapt to novel task constraints or goal representations effectively.

Neural Based Planning Since the NN planner generates reward internally in correspondence to (externally or internally generated) goals by means of a specific module, the NN planner can self-generate reward. This reward is associated with any possible state that the system might happen to pursue as a goal. Thus, goals may be associated with motivations, triggering internal reward and also shaping behavior internally. During planning, the NN planner focuses on states that lie between the starting position and the goal, and those around them. When in planning mode, the actor is a model of itself acting in reality. In this sense, the system actually predicts its own choices in states potentially experienced in the future. This is similar to the XACS model, in which predictions are generated and compared to the state-value representations choosing that action that leads to the anticipated highest state value.

Since the NN planner does not distribute reward values, behavior may be more flexible but requires expensive, potentially exponential online planning upon the activation of a novel goal representation. Action decision is based on a plan-based inversion in that hypotheses are generated looking ahead, distributing reward, and assigning maximally effective actions to NN-based states. While the DYNA-PI system is a very simple planning system that only predicts (potentially stochastically distributed) concrete next states, the NN-based system plans based on predictable state aspects. Unpredictable inputs are ignored but partial predictions are currently not possible. Planning is quite robust in the sense that it involves all states that might likely be visited during action execution. It takes place precisely as an offline improvement of the control policy in relation to the assigned goal.

Curious behavior can be included easily by directing the behavior during exploration to NN regions in which the predictions have high uncertainty. Epistemic actions were not investigated so far and also surprise mechanisms have not been considered in further detail. However, goals can become motivations in the multi-goal versions of the planner so that motivational goals can be readily incorporated and the goal-based planning mechanism can also account for novel goals flexibly, even though this may be computationally very demanding.

With respect to action initiation, the system learns to associate goals with actions once a planning step has been applied successfully. In this case, goals trigger actions / motor (control) programs, especially in the multi-goal version of the system when same goals are assigned more than once. The planning process "compiles" goal-related information into the reactive components of the system. Thus, relying on schema representations, the neural-based planner actually forms a goal-dependent inverse model mapping.

In conclusion, NN-based DYNA-PI is a typical schema-based approach that is capable of online planning and of generating simulated experiences offline. The NN-based approach has the advantage of implicitly generalizing, filtering noisy inputs, and ignoring unpredictable inputs. Hierarchical implementations of the approach, possibly in combination with recurrent structures, await future research effort.

SURE_REACH The SURE_REACH model transforms spatial population-encoded goals (that is, hand locations) into intermediate, redundant population-encoded goals (that is, a corresponding subspace of arm postures). The posture representation is then used to generate motor commands by means of dynamic programming-based planning. Thus, also SURE_REACH is an indirect inversion model that uses its schema representations encoded in its inverse kinematics and inverse sensorimotor model to map goals to actions. The inversion is reward-based in the sense that goal activation, which represents reward, is inversely propagated through the population-encoded posture space. However, due to the population encoding, the operation of this mechanism is not exponential but polynomial and thus efficiently executable online upon goal activation. Thus, the system can yield highly flexible goal-oriented behavior being able to approach any reachable goal in space. Additional constraints can modify internal target representations or movement preparations to adjust behavior to situational demands. For example, new task constraints—be it disabled joints, preferred arm postures, or anticipated subsequent goals— can flexibly modify the unfolding anticipatory behavioral control structure.

Although not further investigated so-far, curious behavior patterns may be included as well—potentially causing movements to spatial regions that have not been sufficiently explored or had not been reached for an extended period in time. Epistemic actions, however, will require the incorporation of additional mechanisms. Also surprise mechanisms have not been further investigated, yet. Motivational goals, however, may be easily incorporated and a first goal-selection mechanism was coupled with the SURE_REACH model, showing avoidance behavior and the preference to reach rewarding locations in space (Herbort et al., 2007).

Thus, SURE_REACH may be considered the most flexible architecture of the schema systems considered. However, so-far SURE_REACH has only been applied to a rather restricted arm reaching tasks. Thus, the generality of the approach as well as the scalability of the representation to higher dimensional problems needs to be further investigated and developed (cf. Butz et al., 2007a).

5.5.2 Inverse Model Approaches

5.5.2.1 Predictive Capabilities

In the discussed inverse adaptive control approaches, the motor control capability is the most relevant and most investigated system part. Thus, the systems' predictive capabilities are of lesser importance and an inverse model may very well be combined with other predictive systems to yield stable online control in addition to the effective inverse control mechanisms, as done in the discussed forward-inverse model approaches.

Generally, DIM, RMRC, and FEL all work on continuous valued inputs and represent continuous changes. No discretization takes place. In general, also payoff representations are not included. All systems can be considered rather noise robust but usually have complete representations and have no mechanisms to focus on partial environmental inputs only. Thus, focusing capabilities are restricted to motor activity and goal dependence. Furthermore, inverse model approaches usually do not rely on exact representations and thus form rather fuzzy associations. The continuous changes are usually represented on one time scale so that more abstract concepts of change are not represented in space nor in time. Nonetheless, the goal representation and the implicit "belief" that the associated goal will be achieved may serve as an interesting representation of a desired future state.

All systems known to us process information in one task-context only and cannot switch between different contexts with different optimality criteria. Finally, the capability of forming internal predictions is restricted to the possibility of activating (desired) goal states, effectively predicting that an activated goal state will become actual. Triggering goal states based on current internal states, which might depend on current internal motivations and emotions, is another future research challenge. Also the distinction between self and other and the potential projection of the inverse model on observed behavioral patterns awaits future research investigations.

Anticipatory Capabilities The anticipatory capabilities focus on decision making and action initiation. As suggested by the name itself, direct inversions from goals to actions trigger behavior upon goal activation. Thus, the inversion is direct, neither reward- nor plan-mediated. The consequence is that the systems also do not have any explicit planning capabilities, and essentially also no re-planning capabilities. Behavior is either successful or fails. Upon failure, further learning is required. However, since most inverse modeling approaches are state dependent closed-loop control mechanisms, disturbances during behavioral control can usually be compensated for. In fact, all three approaches discussed learn inverse mappings from goal representations to motor activity that are conditioned on the current sensory input, which represents the current state of the body in the environment.

Environmental exploration may be biased dependent on the current system knowledge, which enables curious behavior. The possibility to impose goal-directed behavioral execution during learning may be further explored, potentially moving from very general, inaccurate representations to progressively finer-coded, more accurate control. Coupled forward-inverse models discussed above come into mind here, where the forward model accuracy may determine motor activity during learning, and thus bias the focus of inverse model learning. Also motivational constraints may be incorporated easily, resulting in the selection of a goal, which is then pursued by means of the inverse model. Thus, motivations may trigger goals in the reachable space, which can be approached without additional computational effort besides the invocation of the direct inverse mapping.

In sum, inverse adaptive control approaches may be considered as the tools that can realize anticipatory, goal-based action decision making and control. Due to their focus on this aspect of motor control and their general lack of predictive capabilities, it seems straight-forward to modularly combine inverse adaptive control systems with suitable forward predictive mechanisms, which may stabilize control in dynamic control problems. This has been done by the discussed forward-inverse model combinations.

5.5.2.2 Advanced Structures

The discussed advanced structures do either form only predictive representations of sensory inputs or couple some of the discussed system modules. Thus a separate discussion of the considered systems is not carried through.

It may be noted, however, that RNN-like mechanisms can be expected to be necessary to tackle POMDP problems, since internal state representations are necessary in this case. To combine multiple sources of information, these systems may need to be further modularized, which has been partially realized by the gating structure of neurons in the LSTM system. However, partial predictions, and predictions on multiple levels in time and space most likely require the incorporation of multiple forward and inverse modules and the successful knowledge exchange between these modules. Coupled forward-inverse model show one approach to successfully combine direct controllers with schema-based forward models. The discussed hierarchical system approaches may be used to form hierarchical controllers and generate predictions at multiple levels of abstraction in time and space. Moreover, additional challenges, such as epistemic actions, may be tackled with such system combinations. The current capabilities of all discussed systems are now further contrasted from a broader perspective, identifying current system shortcomings as well as arising challenges.

5.6 Discussion

The system classifications point towards several immediate and longer term challenges. In this discussion, we contrast the different systems with respect to their predictive and anticipatory capabilities and identify the most important challenges lying ahead. Hereby, the combination of several systems and system capabilities appears highly promising to generate more complex, autonomously learning, highly adaptive, flexibly behaving cognitive systems.

5.6.1 Contrasting Predictive System Capabilities

The system categorizations showed that there are a rather wide variety of predictive learning systems, each of which also have distinct anticipatory processing potentials. Although it is hard to contrast these potentials directly, Table 5.1 shows an overview of the predictive capabilities of the discussed learning architectures. All systems exhibit highly promising but in many cases differing predictive capabilities. The table may serve as an indicator of the most important challenges lying ahead for each investigated system and which aspects are the most immediate challenges that point towards successful system enhancements and improvements.

The table suggest that there is a current lack of system competencies in several seemingly highly relevant aspects of predictive capabilities: (1) the development of competent predictive system that are able to learn predictions on multiple levels of abstraction in time and space; (2) the development of systems that effectively incorporate context information in their predictions. These two points are discussed in the remainder of this section. Albeit also important challenges, the problem of handling environments with only partially available information (POMDP problems) as well as the problem of the self/other distinction is not further discussed due to the diversity of the problem and its strong dependency on representations and diversity in the approaches to this problem.

Although several of the predictive systems have the potential to predict multiple aspects and provide accuracy or confidence estimates of their predictions, it seems to be difficult to provide multiple predictions in parallel, such as the prediction of

Table 5.1 The contrasted predictive capabilities of the considered systems suggest further advancements as well as potential system combinations.

Aspect	DYNA-PI	XACS	NN-b.D	SURE_REACH	Inverse Mod.
Form	Symbolic	Symbolic	Real-Valued	Real-Valued	Real-Valued
TimeScale	Discrete	Discrete	Continuous	Continuous	Continuous
Noise Robust	No	Partially	Yes	Yes	Yes
Payoff/Sensory	Both	Both	Sensory	Sensory	Sensory
Multiple Space	Single	Single	Single	Single	Single
Full/Partial	Full	Partial	Partial	Full	Full
Det./Fuzzy	Deterministic	Partially Fuzzy	Potentially Fuzzy	Fuzzy	Potentially Fuzzy
Time: Imm./Longer Term	Immediate	Immediate	Immediate	Immediate	Immediate
Generalization	No	Yes	Yes	Yes	Yes
Info. Sources	Two	Two	Two	Two	Two
(PO)MDP	MDP	MDP	MDP	MDP	MDP
Self/Other	Self	Self	Self	Self	Self

next sensory input, plus the prediction of the position of an object in the input, or the prediction of other, often pre-processed environmental features. The hierarchical networks starting from Rao and Ballard (1997) might be an approach to realize such multiple abstract capabilities. The hierarchically combined layers, structured appropriately, may each have a different (emerging) type of abstract representation and thus also abstract predictions. It seems that the integration of other mechanisms, such as the clustering-for-prediction capabilities of the XACS system or the long-term dependency detection capability of the LSTM system, into these hierarchical network structures points towards a highly challenging but also highly rewarding future research direction.

Related but not identical to the capability of predicting at multiple levels of abstract representations lies the capability of predicting at multiple time scales. Again, hierarchical networks seem to have the most potential in this respect. However, even more important than representational abstraction is the question of how to abstract in time. To generate flexible longer time-bridging capabilities during learning, it needs to be clarified when predictive responsibility should be delegated to the next higher level. Early work in this direction suggests that learning at a higher level should be activated if the current level is well-predicting on average but currently encounters highly ill-predicted input (Schmidhuber, 1992a,b). Interestingly, it was recently shown that a very similar principle can serve for the effective detection and generation of options, that is, higher level motor programs, in reinforcement learning (Butz et al., 2004b; Simsek and Barto, 2004). In general, the information content received from the sensory inputs must be significantly higher and persistently high in order to delegate predictive responsibility to the higher prediction layer. Further research in this respect seems very important.

Another approach for multiple levels of abstraction in time is the consideration of delay in sensory feedback. Hierarchical control structures partially take these feedback constraints into account, such as the work of Kawato et al. (1987), in which a lowest-level PD controller serves as backup in case the higher level inverse model-based controllers and forward models are incorrect or inaccurate. The combination of these principles with more competent network structures, points towards another big future research challenge.

The incorporation of multiple sources of information for prediction, apart from the distinction of sensory inputs and action input, is also only partially realized in most of the predictive systems. Hereby, it can be expected that context information should not be simply included as an additional lower level input, but rather should be exploited as a different type of input that serves as a focusing and predisposition mechanism in the system. Thus, in the rule-based XACS system, context may pre-select currently relevant rules, or, in the LSTM system, context information may be used to open and close certain input, forget, and output gates in order to stream information flow in a context-dependent way. The usage of context information from Balkenius' context dependent attention-processing and reinforcement learning systems (cf. Chapter 4) may serve as an inspiration of how to incorporate such mechanisms in a more flexible way into predictive learning systems.

Besides these possible advances, it should be kept in mind that predictive system capabilities are only useful if they serve a purpose, that is, if they affect motor control favorably. To generate competent anticipatory cognitive systems, predictions need to be learned in order to improve learning and behavior. Thus, the general challenge is to develop more competent anticipatory decision making and control systems and possibly also bias the learning of predictions on the resulting anticipatory behavior capabilities. To achieve this endeavor, it will be necessary to combine several predictive systems and couple predictive and inverse systems for the problem structures at hand. Moreover, it will be necessary to exploit their respective competencies modularly to generate more effective anticipatory processing mechanisms. How this might be achieved is outlined in the following section.

5.6.2 Contrasting Anticipatory System Capabilities

Before contrasting the systems' anticipatory behavioral capabilities, we want to point out that the model learning components themselves are not as much influenced by their own predictive capabilities as might be advantageous. Although most considered systems use error-based learning principles, targeting learning resources towards task-specific, motivational goals poses an interesting additional challenge. That is, while "learning for control" may be the first principle, "learning for the achievement of ecological relevant goals" may be an even more focused principle that points out that learning should focus on those control aspects that are really relevant to the learning system.

Table 5.2 shows the current anticipatory capabilities of the discussed learning systems. Schema-based approaches all have similar properties, although the various implementations differ in certain respects depending on their generalization capabilities and utilized representations. Inverse modeling systems are additional mechanisms that may shape behavioral learning directly. In addition, RNN approaches are expected to be useful to learn the achievement of longer-term goals. Hierarchically structured top-down, bottom-up systems may serve well to form abstracted repre-

Table 5.2 The contrasted anticipatory capabilities also show several current drawbacks as well as potential system combinations and integrations.

Aspect	DYNA-PI	XACS	NN-b. DYNA	SURE_REACH	Inverse Mod.
inv.model	no	no	no	yes	yes
reward based	yes	yes	partially	yes	no
general planning	yes	yes	yes	yes	no
focused planning	no	yes	no	no	no
offline simulation	yes	yes	yes	yes	yes
flexibility	low	medium	medium	high	medium
flexible goals	no	no	limited	yes	limited
curious behavior	yes	yes	yes	yes	yes
epistemic actions	no	no	no	no	no
surprise mechanisms	no	no	no	no	no
motivational goals	no	limited	limited	yes	yes

sentations in space and time to be able to plan and act goal-directedly on a more conceptual level.

Besides the potential learning improvements by the means of anticipatory mechanisms, the table shows that several other capabilities require future research. First, faster behavioral adjustments due to unexpected sensory inputs have hardly been investigated. That is, surprise mechanisms could be exploited further for (1) fast self-stabilization mechanisms and (2) the activation of additional cognitive resources for more focused learning and adaptation. Kalman filtering-based updates and other error and information gain estimations may help to improve control and stabilization capabilities in this respect.

Second, task-dependent planning mechanisms may be investigated further. The combination of different predictive methods to enable prediction for action decisions on multiple levels of abstraction seems inevitable. It also remains an interesting question, how exact planning needs to be in order to be sufficiently effective. Davidsson (1997) showed that one-track predictions (those that predict only the usual behavior-dependent future and do not consider alternatives) are often sufficient to improve behavior by inducing preventive mechanisms if the usual behavior leads to undesired states.

Third, while curious behavior has been implemented in a few architectures, epistemic actions were not successfully shown in any of the considered architectures. Epistemic actions may, however, be realized in several systems. However, it remains unclear which predictive representations can most effectively trigger epistemic actions. It seems necessary that a system would need to generate hypotheses about the environment and trigger actions to verify uncertain but relevant hypothesis. For example, in a search task, a robot may look behind an obstacle to see if the ball might be there. Kiryazov et al. (2007) have generated a first realization of such a system on a real robot platform. The system is able to generate hypotheses based on analogy making, consequently triggering goal-directed verification activity.

Hierarchical NN-based system architectures may offer another solution for the realization of epistemic actions. Once higher levels are able to pre-activate lower level neurons, these pre-activations may not only lead to the faster detection of such inputs but also to the activation of suitable motor activity to search for the hypothesized inputs. In general, while systems might have a general curious action selection mechanism, for example, for improving predictive model learning, epistemic actions may be based on the same principle of predicted information gain, only that in this case, plasticity needs to be more dynamic in that the entropy of current important available information needs to be considered and selectively improved. Such mechanisms may lead to truly curious behavior and the automatic activation of epistemic actions.

Fourth, the coupling of motivational mechanisms and potentially even emotional mechanisms with the behavioral decision and control modules poses additional challenges. Context may be handled as a special input to the predictive and to the control system and it may reflect current system motivations. The activated contexts—activated, for example, by a neural activity pattern in the hierarchical neural architecture—should trigger matching motor programs and action decisions that usually lead to the activated context. As discussed, coupled forward-inverse models are a good candidate in this respect, selecting those coupled models that are maximally suitable given the current context.

5.6.3 Integration

The contrasted factors show that the challenges ahead in the design of competent, flexible, and highly adaptive cognitive system architectures comprise system improvements of predictive and anticipatory capabilities. However, possibly even more important, they require the effective combination and integration of various learning and representational mechanisms.

Research currently still focuses on the improvement of particular predictive system capabilities. In the future, though, we expect that successful combinations of different predictive system capabilities will become increasingly important. We expect the following enhancements to be particularly fruitful:

1. The development of predictive systems that process and combine different sources of information (such as context information, sensory information, and action information).
2. The implementation of predictive hierarchies that can generate predictions at different levels of abstraction in time and space.
3. The coupling of predictive representations with action control representations.

Especially the last point poses a great challenge but might be the key to the generation of actual cognitive systems. Perceptions need to be linked with appropriate action codes (causing affordances and bottom-up action predispositions). And, vice versa, action codes need to be linked to corresponding sensory effect codes that are expected to change after action execution. With such a cognitive structure at hand, many anticipatory capabilities might even emerge naturally from the system structure itself. However, even with a less sophisticated representation, several advanced anticipatory capabilities will need to be investigated:

1. Anticipatory representational shaping needs to be further developed, that is, the learning of representations directly for effective behavioral decision making and control.
2. The further development of curious behavior capabilities and epistemic action capabilities: to realize a cognitive system that automatically activates epistemic actions. It seems important that such behavior is triggered by the anticipated information gain that seems most relevant for the achievement of current goals.
3. Anticipatory top-down mechanisms need to be further developed, which influence bottom-up sensory processing. This includes attentional mechanisms (cf. Chapter 4) but also action decision making and control mechanisms since action decision making can be considered as yet another attentional process.
4. A motivational and potentially emotional module may be coupled with the predictive system in order to induce even better action decision making capabilities, enabling the execution of opportunistic actions and actions that are anticipated to satisfy expected motivations (such as taking food and water on a hike).

The anticipatory enhancements are certainly not stand-alone but are very interdependent and also highly dependent on the predictive representations used. Thus, the discussed enhancements of the predictive capabilities of the system should not (only) be pursued in isolation but rather should be designed from the beginning to serve the realization of effective anticipatory action decision and control mechanisms. It is expected that interactive, emergent, and unforeseen properties will be detected along the way of this research endeavor and will as well lead to novel insights in information processing, adaptive behavior, embodiment, and cognition as a whole.

5.7 Conclusions

This chapter has shown that there are various challenges ahead. In order to create competent, anticipatory, adaptive learning systems, the systems do not only need to be competent in learning accurate predictive models of their environment but also need to be able to effectively exploit the learned models for adaptive behavior. This process is expected to be interactive rather than iterative in that the developing predictive representations should immediately cause anticipatory mechanisms that, vice versa, immediately influence the further development of the predictive representations. The categorizations and contrasting discussions in this chapter may serve as guidelines for the development of such more effective anticipatory mechanisms and competent cognitive systems. It is hoped that this chapter does not only provide a useful overview of the discussed systems but that the chapter also encourages further assessments of learning systems with respect to their predictive and anticipatory capabilities and the creation of combinations of these systems to tackle the challenges ahead.