Internal Simulation of Behavior has an Adaptive Advantage

Joost Broekens (broekens@liacs.nl)
Leiden Institute of Advanced Computer Science, Leiden University,
Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands

Abstract

In this paper we test the hypothesis that internal simulation of behavior has a robust adaptive (learning) advantage. From an evolutionary perspective, it is plausible to assume that agents that simulate behavior have an additional survival value compared to those that do not. We present experimental results with a computational model of learning and decision-making. Our experiments are based on biasing the agent's action-selection by a simulation of its future interactions. Using our model, we show that this influence of simulation on learning results in a significant learning advantage. Because increased individual adaptation is an evolutionary advantageous feature, this is a relevant result for the evolutionary plausibility of the simulation hypothesis.

Keywords: action-selection; adaptive agent; reinforcement learning; simulation hypothesis; computational model.

Introduction

It is important to understand the nature of reasoning in adaptive agents, both natural and artificial. This understanding is needed, for example, to efficiently solve problems related to the *frame problem* and to understand mechanisms of generalization versus specialization of knowledge. Reasoning itself assumes there is something to reason about, i.e., knowledge. Reasoning in the context of adaptive agents thus implies there are (at least) two parallel and complementary processes: first, knowledge acquisition, i.e., *learning*, and second the actions resulting from reasoning upon this acquired knowledge, i.e., *behavior*. Reasoning essentially is about making an informed choice, a choice informed by the acquired knowledge and made possible by the constraints of the body of the agent.

The *simulation hypothesis* (Hesslow, 2002) states that thinking consists of internal simulation of interaction with the environment. This hypothesis is based upon three main assumptions: simulation of actions—actions can be prepared not necessarily resulting in execution—, simulation of perception—perceptions can be generated by the brain itself and not necessarily need external stimulation—, and anticipation—the existence of associative mechanisms both between real actions and perceptions and between simulated actions and perceptions. Continuous activation of these associative mechanisms constructs chains of simulated prospective interactions, known as covert behavior. These interactions bias actual behavior, known as overt behavior.

Hesslow (2002) equates thinking with conscious thought. We use a broader definition of the simulation hypothesis, namely: reasoning, as defined above, is facilitated by the internal simulation of interaction with the environment. Our definition of reasoning implies at least two relevant processes, i.e., learning and behavior. We have studied the influence of simulation on the learning of an adaptive agent. Our experiments are based on biasing the agent's action-selection by a simulation of its future interactions. We use a computational reinforcement learning (RL) model. Our model allows a small amount of anticipatory simulation concurrent with its reactive mode of operation. Our agent lives in a gridworld in which it must autonomously learn to forage. The agent has the computational model as "brain". We compare to what extent different simulation strategies result in a different learning performance.

This paper is structured as follows. We briefly discuss the theoretical background, our computational model, experimentation method, and agent system. We end with a discussion of our results and a conclusion.

Theoretical Background

Interactivism (Bickhard 1998; 2001) is a crucial element to our approach, besides reinforcement learning (Sutton and Barto, 1998) and the simulation hypothesis. Interactivism explains reasoning as resulting from the continuous interaction between an agent and its environment. Importantly, interactions can be active and prepared. Active interactions prepare a set of next possible interactions, referred to as interaction potentialities. These potentialities become active when interaction with the environment matches, and thus prepare further interactions. The concept of active and prepared interactions is compatible with the concept of simulated action and perception and the associative chaining of interactions, and is used in our computational model.

Interactivism and the simulation hypothesis have several other important assumptions in common:

1). Thinking is not necessarily symbolic or ideally logical, two of the important limitations of earlier models of cognition. Animals do not necessarily think symbolically (see, e.g., Anderson, 2003; Bickhard, 2001), and frequently make mistakes (Cohen and Blum, 2002; Damasio, 1994).

2). Perception and action are two sides of the same coin, and highly related through (at least) sensory-motor control areas. This avoids the input-function-output paradigm of cognition and is an important point—highly related to issues surrounding the frame-problem—for future development of a neuronal version of our model. However, in this paper we do not focus on this point in detail.

3). These hypotheses closely relate to Damasio's (1994) concept of thinking as an "as if" loop, involving simulated actions that are evaluated by their *somatic markers*, emotional impact estimators. Somatic markers are attached to outcomes of scenarios through learning. Three systems are critically involved, the prefrontal cortex (PFC), the somato-sensory cortex (SSC) and the body. The two mechanisms behind these markers are the body-loop and the

"as if" loop. When the PFC signals the body to be in a certain state, the SSC organizes itself according to the body, i.e., the body-loop. The "as if" loop consists of the PFC instructing the SSC to organize itself, bypassing the body. The body-loop thus involves action, while the "as if" loop involves simulated action. The "as if" loop produces imagined—future—states, and the somatic markers that are attached to these states equal the predicted emotional outcome (reward/punishment). This signal is used to bias decision-making (Damasio, 1994). Even though we do not model the body of the agent, the somatic marker concept is very useful to understand the relation between reinforcement learning, emotion and decision-making.

Evolutionary Continuity and Adaptive Advantage

An important consequence of the simulation hypothesis is that agents do not need a separate "decision module" that evaluates the simulated interactions. Simulated interactions are grounded in existing sensory-motor systems, elicit previously learned emotional consequences and thereby bias action-selection (Hesslow, 2002), much like Damasio's decision-making based on somatic markers and Bickhard's (2000) action-selection based on preference towards one interaction outcome rather than another. If the system prepares to act in two different ways, but for some reason one of these ways appears "more attractive", then the system will eventually prepare the more attractive action and thereby make a choice. Although this approach will result in sub-optimal decisions at some points, it does circumvent the necessity to logically search through large spaces of actions in order to find the "best" action and it provides an efficient heuristic for action selection.

Cruse (2002) argues along the same lines, stating that the evolution of cognitive properties does not necessarily requires the introduction of new additional modules. In Cruse's case by module he primarily means "internal world model", but the argument contains the same message: evolutionary continuity.

So, two key issues of the simulation hypothesis are: (1) no evolutionary leap between humans and other mammals, and (2) no need for a special mechanism to evaluate the imagined future state (Hesslow, 2002). These imply that:

- Notwithstanding evolutionary continuity, for simulation to be evolutionary plausible, it seems fair to assume that in a population of agents, those agents that simulate behavior (having that feature) have additional survival value compared to those that do not. In the case of an adaptive agent, this survival value results in an important way from the enhanced learning performance of the agent. The better an agent is at adapting to a new task, the more likely it is to survive. We will further refer to this enhanced learning performance as *adaptive advantage,* not to be confused with the long term evolutionary advantage an agent is implied to have as a result of the survival value resulting from the enhanced learning performance.
- The simulation mechanism must be robust; small changes in other parts of the agent's information processing system may not seriously downgrade the adaptive advantage.
- It must be possible to use existing (possibly slightly changed) mechanisms to simulate and evaluate the near future. This is compatible with Svensson and Ziemke (2004) who stress that one of the keys to our understanding of embodied cognition is to understand how sensorymotor processes and higher-order cognition *share* neural mechanisms, and this is compatible with Cruse's (2002) line of argument mentioned earlier.

In this paper we test the hypothesis that internal simulation of behavior has a robust adaptive advantage without the need to make major changes to learning and evaluation mechanisms.

Hierarchical-State Reinforcement Learning

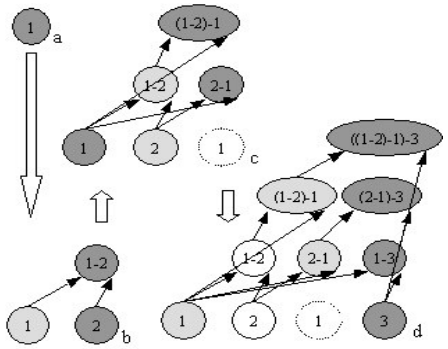

Figure 1a-d. Example instance of a model resulting from the sequence of situations "1-2-1-3" presented to an initially empty model. Dark-gray nodes are active at time *t*, light-gray nodes were active at *t-1*.

To be explicit about the changes that enable our computational model to use a small bit of anticipatory simulation, we first explain the basic model (Broekens and DeGroot, 2004) that does not implement the simulation hypothesis. It is a predictive, connectionist, interactivist-based computational model of learning and decision-making, *though it is not neural*. It has the following characteristics. (1) Information is stored as a directed graph in which nodes encode interactions. (2) Interactions have a stability property, learned through continuous interaction with the environment, analogous to Bickhard's (2000) concept of stabilization and destabilization. (3) Interactions have a value property, learned through reinforcement. (4) Interactions can form between other interactions resulting in a hierarchy of more and more complex interactions (Figure 1). By doing so the model builds hierarchical predictions of future states. (5) Its initial state is empty, and it grows by interaction with the environment: i.e., the model is only used in an online learning setting. Every interaction takes the same fixed amount of time. (6) When a new situation presents itself at time *t*, the model automatically creates a new node representing that situation. (7) Interactions that were active at time *t-1* are connected to this new node. These connections are also represented as new nodes that can subsequently function as more complex interactions (e.g., creation of node "1-2" in Figure 1b). (8) Nodes are created only if they do not yet exist (e.g., no second node "1" in Figure 1c, but a new interaction node "(1-2)-1" that

connects node "1-2" with node "1"). (9) Typically, at any moment in time, many interactions are active, but at each level of complexity only one interaction is active (e.g., Figure 1d, dark gray nodes). Active interactions include those that have just taken place at time t, but also those between the interactions at time t-1 and the interactions at time t (e.g., node "1-3" and node "(2-1)-3" in figure 1d), etc. (10) This process of interaction-chaining continues until a maximum level, k, defining the maximum amount of knowledge about situations in the past that is present in the current state of the model, as well the maximum number of interactions that can be active at one time (c.f. Figure 1). So a node is a "gate" between a sequence of previous interactions and a set of potential next interactions. We call such a node an "interactron".

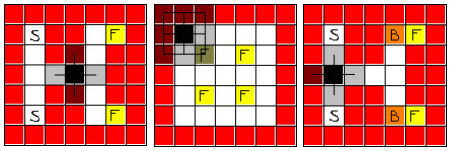

Figure 2a-c. Three different experiments. Agent is black, lava is dark (red), food='F', roadblock='B', start location='S' Tasks from left to right: *find food*, *forage*, *invest*.

Our experiments are performed in gridworlds. A gridworld is a two dimensional grid containing positively and negatively reinforced locations and objects, in our case, lava (negative reinforcement of –1), roadblocks (–0.5), food (+1.0) and empty—neutral—cells (Figure 2). The agent is able to walk on any type of cell, but is discouraged to walk on the lava (by the negative reinforcement). The agent selects an action from its set of potential actions $A=\{up, down, left, right\}$, executes the action in the gridworld and perceives the result of that action. One single interaction with the environment (also referred to as *situation*) is defined as an action-perception pair. The agent's perceptual field has either a chessboard (Figure 2b) or a cityblock (Figure 2ac) metric. For example, in Figure 2c, the agent would perceive something like "plppp" representing the (l)ava left of the agent and the (p)ath above, right, below, and under the agent. If the agent came to this cell by moving to (d)own, the interaction the model receives would be "dplppp" (replacing, e.g., every node "1" in Figure 1).

Reinforcement Learning, Probabilities and Action Selection

In our model we have implemented *stabilization* and *destabilization* of interactions based on the insights of Bickhard (2000). If a node x is activated, the usage υ_x of that node is increased by 1. The function $p^t{}_y(x)$ calculates the *conditional usage* of node x under the assumption that y is active at time t and is defined by:

$$p^t{}_y(x) = \upsilon_x \Big/ \sum_{i=1}^{n(y)} \upsilon_{x_i}$$

Where $x_1,\ldots,x_{n(y)}$ the potential interactions predicted by node y, $n(y)$ the number of potential interactions predicted by y and $x \in \{x_1,\ldots,x_{n(y)}\}$. For example, in Figure 1d, if we assume that y="1" and active at time t, that x= "1-3", and that node "1-2" has been active twice as often as node "1-3", then $\upsilon_{x1} = \upsilon_{1\text{-}2}$ (i.e., the usage of node "1-2") and $\upsilon_{x2} = \upsilon_{1\text{-}3} = \tfrac{1}{2}\upsilon_{1\text{-}2}$ (i.e., the usage of node "1-3"). Therefore, in this case $p^t{}_y(x)$ equals 1/3, which is the naïve probability that x_1 occurs under the assumption y at time t. If $p^t{}_y(x)$ drops below the threshold θ—the destabilization rate—x is deleted including all its dependencies. Consequently, consistent interaction with any part of the environment results in a stable sub-graph of nodes. Inconsistent interaction results in the destabilization of the involved nodes and eventually in the deletion of these nodes. We use θ in our experiments to simulate different rates of forgetting. Note that $p^t{}_y$ is the local probability function for y and that every node has its own function $p^t{}_y$ (learned by interaction with the environment). Therefore, at any time t at most (and during normal operation exactly) k of these functions are active, for there are k active nodes.

Our reinforcement mechanism has two parts. First, when the agent acts, all active nodes y receive a reinforcement signal, r^t, at time t that changes the *direct reinforcement* value, λ_y, of these nodes with a learning rate ρ:

$$\lambda^{t+1}{}_y = \lambda^t{}_y + (r^t - \lambda^t{}_y) * \rho$$

Second, every node has an indirect inherited reinforcement ν—the result of the back-propagated markers of hierarchically higher nodes. λ and ν are summed into the final value $\mu=\lambda+\nu$, reflecting Damasio's assumption that the somatic marker of a predicted situation equals its own value added to the sum of all the cumulative values of the interactions it predicts. When a node y is active, the marker $\mu^t{}_y(x)$ of any hierarchically higher node x prepared by node y, is used to update the *indirect reinforcement*, ν_y, of node y:

$$\nu^{t+1}{}_y = \sum_{i=1}^{n(y)} \mu^t{}_y(x_i) * p^t{}_y(x_i)$$

Markers are thus propagated back through the interaction hierarchy only when the interactions to which they are attached are prepared. This lazy propagation reflects the probabilistic properties of the interactions with the environment. This mechanism follows standard TD learning mechanisms (Sutton and Barto, 1998) except, e.g., the probabilistic value-function defined per node.

Action-selection is based on a winner-take-all (WTA) mechanism and biased by the μ of all prepared nodes. Note that any interaction is composed of both an action and a perception. All *prepared* interactions inhibit (negative μ) or exhibit (positive μ) the level of activation $l^t{}_{a\_h}$ at time t of the agent's possible actions $a_h=a\_h \in A$ in the following way:

$$l^t{}_{a_h} = \sum_{i=1}^{k} \sum_{j=1}^{n(y_i)} {}^{*} \mu^t{}_{y_i}(x^i{}_j)$$

with active nodes $y_{i,}$ and nodes x^i_j where i denotes dependency on y_i and x^i_j prepares a_h (indicated by *).

Additionally, if there are any good actions (any $l^t{}_{a\_h}>0$) the best action a_h, i.e., $l^t{}_{a\_h}=max(l_{a\_1},\ldots,l_{a\_m})$, is selected. If there are only bad actions (all $l^t{}_{a\_h}<0$) a stochastic selection is made based on $l_{a\_1},\ldots,l_{a\_m}$; the action with the highest activation therefore has the highest chance of being chosen resulting in a probabilistic WTA action selection. So, action-selection is simultaneously based on generic and

specific knowledge, allowing it to learn and use generic aspects of the environment as well as more specific ones.

Experimental results have shown that this model is able to learn, unlearn and reuse information, and to solve a T-maze like selection tasks where the agent learns to conditionally use a crossing (Broekens and DeGroot, 2004).

Internal Simulation and Action-Selection Bias

Our basic predictive model does not include internal simulation of behavior. To study the influence of simulation on learning we add the following capability: after every real interaction with the environment, the model simulates one time-step ahead. Analogous to what Hesslow (2002) describes, the model always is one step ahead of the actual situation. To enable simulation we changed the model in the following way. Instead of selecting an action based on past interactions the following process is executed:

1. Select: at time t select a set of to-be-simulated interactions from the interactions predicted by all k active nodes.
2. Simulate: send the selected interactions to the model as if they were real interactions. The model advances to time $t+1$.
3. Reset-state: to be able to select an appropriate action, reset the model's state (the active interactions) to the previous timestep, i.e., time t.
4. Action-selection: select the next action using the standard selection mechanism (explained later). The propagated markers of the simulated interactions have biased this action-selection.
5. Reset-markers: reset μ, λ and ν of the interactions that were changed at step 2 (simulation) to the values of μ, λ and ν of these interactions before step 2.

Step 1 selects predicted interactions to be simulated. In our experiment we have used four different selection mechanisms (also referred to as *simulation strategy*).

- First, no simulation (NON). The actions are selected as described in the previous section and the 5-step simulation procedure is not executed.
- Second, simulation of the predicted best interaction (BEST). The winning interaction of the WTA selection resulting from step 1 is sent to the model for simulation (step 2). Every real interaction is accompanied by a reinforcement signal. As this is a simulation we lack such a signal. Instead, this signal is simulated using the μ of the winning interaction as reinforcement, so we simulate the predicted interaction and its associated marker, analogous to Damasio's (1994) "as if" loop.
- Third, a selection of not just *the* best, but the predicted 50% best interactions, a more balanced selection, (BEST50). Again we simulate the reinforcement signal using the μ's of the simulated interactions.
- Fourth, all of the predicted interactions (ALL).

In essence, BEST, BEST50 and ALL simulate three different values for the *selection threshold* of the WTA interaction selection (ranging from high to low respectively) that is used to select the interactions for the simulation step.

In the simulation step (2) the stabilization-destabilization process is deactivated. Earlier experiments showed that, when active, the agent's behavior is inconsistent with the environment, probably because simulating certain interactions alters the knowledge of the environment because the conditional probabilities of the nodes change by simulating an interaction, distorting the real probabilities.

After resetting the state to one that is appropriate to the current situation (step 3), the simulation mechanism results in: (1) a propagation of the markers μ of the predicted interactions at time $t+1$ to the ν of the simulated interactions at time t according to the reinforcement learning principle used, and (2) an update of the direct reinforcement value λ of the simulated interactions at time t based on their own μ. This means that, without further changes, simulation *by itself* not only propagates predicted reward and punishment but also changes the direct reinforcement from the environment. These learning effects are interesting to study, however, here we want to study the effect of simulation on learning *by biasing action-selection*. Therefore, after action-selection, step 5 is needed.

The changes to the actual architecture of information processing are minimal, an important fact in light of the evolutionary continuity argument of the simulation hypothesis. In a more dynamic model step 2, 3 and 5 would have to be reconsidered.

Experimental Setup

Every simulation strategy is tested in three different tasks that involve finding food, each task in its own unique environment. The darker cells around the agent in Figure 2 show the agent's perceptual area for every task. Since we also want to know how robust this advantage is, we vary the learning rate ρ and the destabilization threshold θ (see Table 1). For the first task the agent has to learn its way from a randomly changing starting location (S in Figure 2a) to a randomly changing food location. When successful, the agent is replaced at a starting location and tries again. Repeating this process enables the agent to learn how to get from both starting locations to both food locations.

For the second task the agent has to learn how to optimize foraging (Figure 2b). Now, the agent is initially placed in the environment, after which it should just explore and find food. The food locations are randomly selected, and the challenge for the agent is to forage.

For the third task, the agent has to learn to overcome an initial negative interaction (road-block, B in Figure 2c, reinforcement of −0.5) in order to get to a larger positive one (food, "F" in Figure 2c). We changed the reinforcement of the food to +1.75 in order to compensate for the negative reinforcement of the road-block. With this experiment we wanted to test how the simulation strategies handle an "investment". By setting the reinforcement of the food equal to 1.75 the average reinforcement of the food remains 1.0.

Every experimental setup (c.f. Table 1) is run 15 times. For every run the agent has 255 trials to find the food. It has to learn the properties of the task within these 255 trials. For every trial the agent has a maximum of 1000 steps to actually get to the food. If the agent reaches this maximum, the agent advances to the next trial.

Table 1: Venn diagrams of statistical difference between simulation strategies. Every diagram (cell in table) is a representation of these differences in one setting. Overlap means there is no statistically significant difference (one tailed *t*-test, α=0.05, n=15). Higher is worst (more steps), lower is better (less steps). Light is NON and dark is ALL and interpolated for BEST and BEST50.

Task					
Find food		θ=0	θ=0.01	θ=0.03	θ=0.05
	ρ=1				
	ρ=0.8	E			
	ρ=0.5				
Forage	ρ=1				
	ρ=0.8				
	ρ=0.5				
Invest	ρ=1				
	ρ=0.8			E	
	ρ=0.5				

Results and Discussion

Because our goal is to compare the relative effect in terms of overall performance between different simulation strategies, not to optimize parameters, ρ and θ have been chosen based on workable values and we have not done an exhaustive search for "the best" parameters. This also explains our quantitative comparison approach (Table 1). We compare the average of the total number of steps needed to finish one run (one run equals 255 trials, we average over 15 runs). If every step is assumed to cost some effort, this average is a measure for the performance of a specific setting (a tuple of strategy, task, ρ and θ). Comparison of these averages gives an overall idea of the adaptive advantage of the different simulation strategies. The lower the average, the better the strategy is. Also, by comparing these averages, we can identify the relation between strategy, task, ρ and θ. Individual learning curves (Figure 3) are not needed to compare the overall performance of strategies and will only be considered if needed to explain a certain relation in more detail.

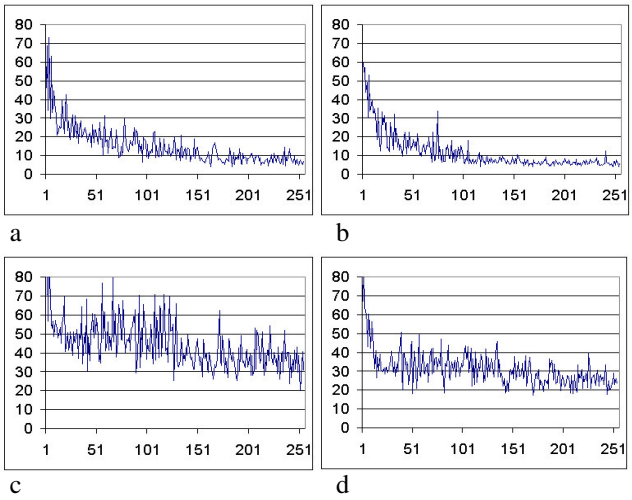

Figure 3. Prototypical simulation effect (ab), a=NON, b=ALL, and specific simulation effect (cd), c=NON, d=ALL. Trials on *x*, steps to complete a trial on *y*

In general, the following results have been observed. The ALL simulation strategy has a robust adaptive advantage compared to the other strategies, specifically at the *forage* task. ALL is either among the best-performance strategies or there is no difference between strategies at all. This suggests that internal simulation of interactions, even if it is just one step ahead, helps an agent to learn a task by providing an extra action-selection bias, and thereby provides an adaptive advantage for the agent. In general this is because ALL either converges faster or better (or both). As a prototypical example, observe the difference in learning curves between (NON, *forage*, ρ=*0.5* and θ=*0.01*) and (ALL, *forage*, ρ=*0.5* and θ=*0.01*) in Figure 3ab. More specific, this is because the forgetting rate and the difficulty of the task disrupt learning almost entirely in the NON case (*invest*, ρ=*0.5* and θ=*0.01,* Figure 3cd). Two notable exceptions ('E', Table 1) are at the *find food* task with ρ=*0.8,* θ=*0* and at the *invest* task with ρ=*0.8* and θ=*0.05*. The first shows that BEST performs significantly better than NON, the second shows that BEST50 performs significantly better than the rest.

To explain these effects we consider the following three reasons. First, to solve the *forage* task the agent requires an explorative, broad view. There is no best path to learn, and instead the agent has to learn where food can be found on

average. If the agent always tries the local "best" solution (i.e., uses NON or BEST), it runs a larger risk of ending up in a local minimum. ALL forces the agent to simulate all of its prepared interactions, including those that appear bad but have a good result at *t+2*. Simulation of an apparent bad interaction can still bias action-selection in a way that favors that interaction if the resulting interaction at *t+1* is good. This results in a broader view. In contrast, to solve the find food task, the agent needs to quickly propagate back the values, it has to find the best path. A broad view is not necessary here, for *find food* is a simple and very goal directed task. The simulation strategy that quickly propagates the positive reinforcement back to the beginning performs best. Both BEST and NON only try the best prepared interaction. So the advantage of the broad view of ALL compared to NON and BEST is less important in the *find food* task. This is supported by the fact that at the *find food* task ALL is significantly better than BEST for *only 2 settings*, while in the *forage* task ALL is significantly better than BEST *for all settings*.

Second, compared to NON, ALL/BEST50 is robust to different rates of forgetting θ, but this effect is specifically noticeable in the *invest* task, where even a small θ (*=0.01*) disrupts learning for NON but not for ALL. This task is difficult, so the agent needs more time to learn. This means that there is more time to forget parts of the already built world model. Because ALL and BEST50 can simulate interactions that appear bad but are good at *t+2*, they also have a higher chance at influencing the action-selection process such that a good backup action is chosen when a part of the model has been forgotten. ALL and BEST50 provide a more balanced heuristic to select the next interaction. This is supported by the more hockey stick shaped learning curve of (ALL, *invest*, ρ*=0.5* and θ*=0.01*) (Figure 3d) compared to NON (Figure 3c). It seems that NON forgets knowledge at such a rate that performance can actually get worse (around *t=60*), while the performance of ALL first increases quickly after which it keeps increasing slowly.

Third, learning rate seems to affect NON the most (not shown in figures). This is due to the fact that learning depends on how quickly the model propagates back the positive reinforcement of the food. Since all simulation strategies simulate 1 step ahead, the positive reinforcement is visible earlier, thus affecting simulation strategies less than NON. This is also the reason why simulation is dramatically better than NON on the *invest* task (c.f., Figure 3cd). The roadblock investment is a problem for NON, but simulation can overlook the investment to the food reward.

Last, at this point it is unclear why BEST50 performs better than ALL at the *invest* task, ρ*=0.8* and θ*=0.05*.

Cognition, Planning, and Simulation of Behavior

We currently connect nodes—and encode and compare information in these nodes—in a way that works in simple gridworlds but introduces problems for real-world navigation. Specifically, the number of nodes exponentially increases if the complexity of the world (and actions of the agent) increases. In light of the hypothesis that cognitive systems are those that have the ability to plan (in contrast to reactive systems, see e.g. Cruse, 2002), this is an important shortcoming of our computational model. If the world is complex, many interactions can be prepared (planned), resulting in an explosion of simulation effort (specifically for ALL). This is highly related to both (a) the fact that nodes are distinct even if they share many of the features of other nodes and (b) the fact that our model does not extract relevant features from the environment, but instead features are encoded in the nodes and assumed equally important. We are working on a neuronal implementation of the nodes in order to start to address this issue. However, in this paper we have focussed on the overall mechanism of simulation. We believe that the positive adaptive effect of this mechanism exists even if our use of nodes changes.

Conclusion

Using our computational model, we have shown that the influence of simulation on learning has a significant learning advantage. This positive effect occurs in three different learning tasks, and for a variety of learning rates as well as rates of forgetting. Since increased individual adaptation is an evolutionary advantageous feature, this is a relevant result for the evolutionary plausibility of the simulation hypothesis. We realize that such conclusions based on computational models should be made carefully.

Acknowledgements

I would like to thank Fons Verbeek and Walter Kosters for helpful comments and ideas.

References

Anderson M. L. (2003). Embodied cognition: A field guide. *Artificial Intelligence, 149.* 91-130.

Bickhard. M.H. (1998). Levels of representationality. *JETAI, 10*, pp. 179-215.

Bickhard. M.H. (2000) Motivation and emotion: An interactive process model. In: R.D. Ellis and N. Newton (eds.), *The Caldron of Consciousness.* New York: J. Benjamins.

Broekens J. and DeGroot D. (2004). Emergent representations and reasoning in adaptive agents. *In Proc. 3rd International Conference on Machine Learning and Applications* (pp. 207-214).

Cohen J.D. and Blum K.I. (2002). Reward and decision. *Neuron, 36.* 193-198.

Cruse H. (2002). The evolution of cognition—a hypothesis. *Cognitive Science, 27.* 135-155

Damasio. A.R. (1994). Descartes' error: Emotion, reason, and the human brain. New York: G.P. Putnam.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *TICS, 6, 2002,* 242-247.

Sutton R. S. and Barto A. G. (1998). *Reinforcement learning: An introduction.* Cambridge, Massachusetts: The MIT Press.

Svensson H. and Ziemke, T (2004). Making sense of embodiment: Simulation theories and the sharing of neural circuitry between sensorimotor and cognitive processes. *Proc 26th Ann. Conf. of the Cogn. Sci. Soc.* Mahwah, NJ: Lawrence Erlbaum.