Exploring Internal Simulations of Perception in a Mobile Robot using Abstractions

(HS-IKI-MD-04-009)

John Stening
(b00johst@student.his.se)

School of Humanities and Informatics
University of Skövde
Box 408, 541 28 Skövde, Sweden

Supervisors: Henrik Jacobsson, Tom Ziemke

Submitted by John Stening to the University of Skövde as a dissertation towards the degree of M.Sc. by examination and dissertation in the School of Humanities and Informatics.
I certify that all material in this dissertation which is not my own work has been identified and that no material is included for which a degree has already been conferred upon me.

Signed: ______________________________

Abstract

This thesis investigates the possibilities of explaining higher cognition as internal simulations of perception and action at an abstract level. Relatively recent findings in both neuroscience and psychology indicates that both perception and action can be internally simulated by activating sensory and motor areas in the brain in absence of sensory input and without any resulting overt behavior. An investigation was conducted in order to test the hypothesis that perception can be simulated in a mobile robot using abstractions. The result from this investigation showed that this was indeed the case but that the accuracy was limited. The simulations allowed the robot to anticipate long chains of future situations but were not good enough to support any overt behavior. To further improve the results there is a need for better training techniques and/or a more complex architecture.

Keywords: internal simulations, emulations, perception, representations, autonomous agents.

Acknowledgement

First of all I would like to thank Tom Ziemke and Henrik Jacobsson for being excellent supervisors, providing valuable ideas and advice during my work on this thesis.

I would also like to thank Nicklas Bergfeldt for helping me understand the details on the YAKS Khepera simulator. You have saved me a lot of trouble during my experiments.

Finally, I would like to say thank you to my friends and family for being there and having patience with me during this period.

This thesis would not have been possible without you.

Content

1 Introduction

What makes a creature a *cognizer*? Is it the ability to merely survive in a hostile environment or is there something larger and more fundamental? According to Varela, Thompson and Rosch (1991) all living creatures are cognitive. This does however not necessarily mean that they are all cognizers. Clark and Grush (1999, p.7) for instance, argue that what distinguishes cognizers from non-cognizers is the ability to go beyond being "trapped in a (…) web of closed-loop interactions with the very aspects of reality upon which their survival depends" and instead rely on some kind of higher internal (or external) models to aid their behavior. Looking back at history it is possible to see some important leaps in evolution. About 3.5 billion years ago the first single cell organisms made their first appearances. These were (and still are) completely reactive creatures without any ability to consciously influence their own situation. Their inability to go beyond the here and now does not qualify them as real cognizers. About 3 billion years later the first reptiles arrived shortly followed by the dinosaurs. These were more sophisticated creatures with somewhat higher cognitive abilities, barely aspiring to be cognizers. They were, at least to some degree, able to go beyond being completely reactive to the environment. Around 250 million years ago the first mammals appeared. Mammals are relatively complex creatures, able to plan ahead for a short time, and at least to some degree predict the outcome of their actions. This makes them aspirants of being called cognizers in Clark and Grush (1999) terminology. Seen from this angle, it is possible to say that something very important happened around 250 million years ago in evolution, regarding the complexity of the mind. Kenneth Craik (1943) also realized the big benefits of being able to use some kind of model to improve the behavior:

> "If the organism carries a "small-scale model" of external reality and of its own possible actions within its head, it is able to try out various alternatives, concluding which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it."

To be able to evaluate future behaviors before they are set in action would of course represent an enormous evolutionary advantage. It gives the cognizer the ability to "[try] out ideas and let its hypothesis die in its stead." (Dennett, 1995 in Clark & Grush, 1999). Hypotheses or actions, which earlier would have lead to a certain death or loss of food can now be discarded in favor of more promising ones.

Of course, there is still a very big difference between the first mammals and the great apes in general and the human mind in particular. But following ideas from Hesslow (2002) and Clark and Grush (1999), if these fundamental mechanisms, found in early cognizers once were in place, they probably only needed a cheap evolutionary modification to become far more complex.

There is however, a classical dispute regarding the nature of these cognitive mechanisms dividing the research community. On the one side, there is traditional cognitive science trying to explain cognition as information processing and maintenance of accurate internal representations of the external world. On the other side, there is a constructivist[1] bottom-up approach trying to explain cognition as something based on a symbiosis between the mind, body and environment. Lately some researchers have presented novel theories which, to some degree, reach beyond this polemic, and explain higher cognitive functions as internal simulations of action and perception. It lies within the focus of this dissertation to combine these new theories with ideas in the autonomous agent research community found in the bottom-up constructivist approach, and investigate the possibilities of developing autonomous agents which, like the first primitive cognizers, to some degree can anticipate the consequences of their own actions.

1.1 Dissertation outline

The rest of this dissertation will be structured as follows: Chapter 2 will introduce the reader to some common beliefs, disputes and new ideas within today's cognitive science. Furthermore, Chapter 2 discusses how cognition can be explained as simulations or emulations of actions and perception. This Chapter also discusses some previous attempts to implement such a simulation or emulation theory in a robot. Finally, Chapter 2 will cover some new techniques which might improve the results of these implementations. Chapter 2 ends up in a hypothesis which is presented in chapter 3. Chapter 4 gives the setup for a number of experiments which might answer this hypothesis. The raw results of these experiments are presented in chapter 5. Chapter 6 gives a more thorough conclusion about the results and a general discussion of the implication of these results and what should be done next.

[1] This thesis will use the term constructivism as defined by Stewart (1995), e.g. our experience are not passively derived, but actively build up through interaction with the environment. Knowledge are basicaly constructed from such experience. Furthermore, all living creatures are more or less cognitive.

2 Background

This chapter introduces the reader to the current situation in cognitive science and how cognition in general can be explained as simulations or emulations of perception and behavior. Furthermore, this chapter discusses both how simulations have been previously tested in robot studies and how the results from these experiments could be further improved using abstractions.

2.1 A New Perspective on Cognitive Science

This section gives an overview of two common approaches in cognitive science, traditional cognitivism and bottom-up constructivism, and the classical dispute that stands between them. It also discusses what is needed within the constructivist approach to take one step further, and start dealing with 'higher' cognitive abilities. A suggestion is given in a simulation or emulation hypothesis (Hesslow, 2002; Clark & Grush, 1999; Grush, 2002).

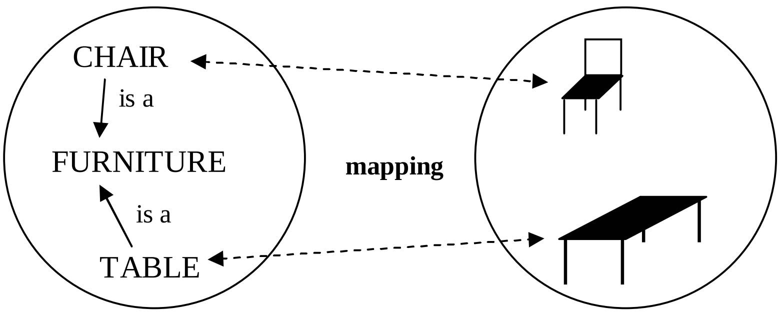
2.1.1 The traditional view of Cognitive Science

Until the last couple of decades traditional cognitive science has unchallenged treated cognitive science as something mainly concerning information processing and maintenance of internal representations as a reflection of an external and independent world (e.g. Pfeifer and Scheier, 1999; Gardner, 1985). The reason for this can be said to have its origin in historical circumstances. At the same time as cognitive science became a standalone research field, the modern computer made its first appearance. Von Neuman was according to Gardner (1985) one of the first to draw striking parallels between the human mind and a computer. Similarly, Turing (1950) compared *digital computers* and *human computers*. This has led some researchers, following in Turing's footsteps, to argue that the human mind is nothing but a computer with mind-like software. Therefore it became natural to think of the mind as an information processing unit which receives input from the environment (perception), processes the information (thinking), and acts upon the decision reached (behavior) (Pfeifer & Scheier, 1999). The comparison between computers and the human mind also made it natural to make a distinction between hardware and software. In order to understand cognition, one only needs to study the latter. The physical machinery is not fundamental for cognition – only the implemented program. These ideas in traditional cognitive science can be summarized in the term *functionalism* which is expressively described by Pfeifer and Scheier (1999, p.43):

> "Functionalism (...) means that thinking and other intelligent functions need not be carried out by means of the same machinery in order to reflect the same kind of processes; in fact, the machinery could be made of Emental cheese, so long as it can perform the functions required. In other words, intelligence or cognition at the level of algorithms or computational processes without having to consider the underlying structure of the device on which the algorithm is performed."

Unfortunately, if one focuses too much on the software, there is a risk of missing important features about the hardware. This has resulted in traditional cognitive science practically having ignored possible bodily aspects of cognition (maybe with an exception for perception). It has also had the consequence that researchers sharing the traditional view have been testing their cognitive models on small sub-domains of cognition, like for instance playing chess. Clark (1997) describes this kind of focus on a small slice of cognition as dealing with *vertical microworlds*.

Traditional cognitive science can also be said to be objectivist and realist in its ontology (Stewart, 1995). It usually presupposes the existence of an objective independent external world which is more or less accessible to us. Furthermore, traditional cognitive science is often said to hold internal representations as a reflection of the independent external world, as fundamental for cognition. One clear formulation of this view of representations comes from Palmer (1978). According to his view a representation consists of five aspects; (1) the represented world, (2) the representing world, (3) what aspects of the represented world are being modeled, (4) what aspects of the representing world are doing the modeling, and (5) what the correspondence between the two worlds are. This *correspondence* view of representation is illustrated in Figure 1.



internal representations
Are built up from symbolic representations which correspond or refer to object and concepts in the external reality.

external reality
contains objects

Figure 1: The correspondence notion of representations. Some aspects of the represented world are modeled in the representing world in such a way that there exist some correspondence between the represented world and the representing world. Adapted from Dorffner (1997) and Ziemke (2001).

This view on representations should be put into the context of symbols and symbol-manipulation which is used in traditional theories in order to explain the basics of intelligence. Or as Newell and Simon (1976, p. 83) express it: "Symbols lie at the root of intelligent action, which is, of course, the primary topic of artificial intelligence". A symbol is something that stands for, or represents something else. These symbols can be further organized in expressions or symbol structures which are used to describe the current state of the world. Besides symbol structures there are processes working on these expressions in order to create new expressions. These ideas are summarized in the physical symbol system hypothesis (PSSH) (Newell & Simon, 1976, p. 87) which states that a "physical symbol system has the necessary and sufficient means for general intelligent action". Physical here means that such a system clearly obeys the laws of physics. General intelligent action means the same scope of action as we see in human action. Put together with Palmer's (1978) ideas the symbols and the processes can be said to constitute the representing world and the parts of the representing world which are doing the modeling.

As previously hinted, many of these theories have been questioned. For the last two decades, some researchers with different approaches to cognitive science have started to point out fundamental flaws in the previous work. Some of the more important critics concern the total focus on the mind and the lack of understanding for the importance of bodily and environmental aspects in cognition (Brooks, 1991a, 1991b; Clark, 1997; Pfeifer & Scheier, 1999; Varela, Thompson & Rosch, 1991). There are those who claim that these aspects are as important as the mind itself if we want to understand cognition. A big part of our behavior, and consequently our cognitive processes, is based on interaction with the environment. Moreover, the environment lets us offload information into it, so that the cognitive workload is reduced. In order for this to work we are relying on our body, which is not a complement to the mind but an extension, which lets us interact with the environment.

Other researchers question the use of symbols as the foundations of both representations and intelligence (Brooks, 1991a, 1999b; Searle, 1980). Brooks (1991a) for instance claims that the need for internal representations is overrated. Much cognitive work can be solved in a more natural way without them, by just interacting with the environment. A more fundamental and deep philosophical argument against the use of symbols as representations comes from Searle (1980). In his *Chinese room argument* he claims that symbols as a reference to an independent external world lacks meaning in a strong sense. A computer with an installed program (even mind-like software) which stores and handles information in symbolic form (all information in a computer is symbols) does not understand the meaning of the symbols – because they are not *grounded* in the outside world. Even if one trace the meaning of one symbol back to the meaning of another, one will always come to a starting symbol which lacks explanation or ground in the outside world. As Dorffner (1997) pointed out, a large part of the problem lies in the fact that most of these software's are designed by humans. Hence, the meaning of the symbol used by the program is not its own but the designers (Figure 2). The claim made by Searle that no computer can have understanding does not mean that there could not be intelligent machines (humans are such machines). It only states that such machines can not be based on a computer metaphor.

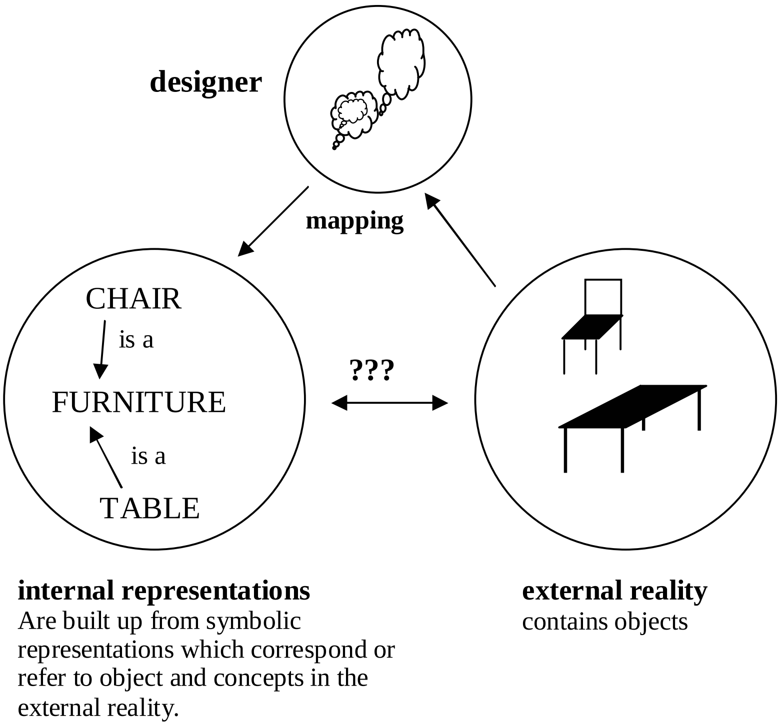


Figure 2: The problem of designed representations. What is the actual correspondence and who has the representations – the designer or the system? Adapted from Dorffner (1997) and Ziemke (2001).

There are also those who question the focus on sub-domains or vertical micro worlds (Brooks, 1991a, 1991b; Clark, 1997; Dreyfus, 1979) and want to aim for a more holistic picture of cognition. Traditional cognitive scientists has been successful in developing a big variety of application and expert systems capable of solving complex tasks – the medical expert system MYCIN and the chess playing Deep Blue, to name but a few. But the big question is if that is what cognition is all about? Many researchers claim that there are more important skills a cognitive creature must possess, most importantly the skills of surviving in a challenging environment and to reproduce.

In order to handle criticism like these there is a need for a different approach to cognitive science in general and artificial intelligence in particular. An approach which takes into account the importance of the body and environment and avoids falling into the trap of using ungrounded symbolic representations. In addition, the approach should not be based on a metaphor between computers and humans, but put its reliance on more biologically plausible solutions. The next section gives an overview of such an approach.

2.1.2 A Constructivist bottom-up approach to Cognitive Science

Brooks (1991a) argues that in order to understand human-level intelligence we must spend a lot of time practicing with simpler models. Human-level intelligence is simply too complex and too little understood to be a good starting point. While cognitivists have dealt with the complexity problem through dividing cognition into sub-domains, Brooks (1991a) and Beer (1990) argue that we must try to build simple but complete systems. This will give us a chance of understanding how the pieces interact and how they form an emergent whole. These systems should also be let loose in a real environment and deal with real perception and real action. In order to make this possible the systems need to be a physical being in some sense. These ideas can be summarized in the two very important concepts, defined by Brooks (1991b, p.571, original emphases) – situatedness and embodiment:

> "[**Situatedness**] The robots are situated in the world – they do not deal with abstracted descriptions, but with the here and now of the world directly influencing the behavior of the system. "
>
> "[**Embodiment**] The robots have bodies and experience the world directly – their actions are part of a dynamic with the world and have immediate feedback on their own sensation."

To be embodied implies that the system continuously answers to physical forces etc. (Pfeifer and Scheier, 1999). On the one hand, this makes the systems situation considerably more complex, but on the other hand, it simplifies a lot of everyday problems. One example is walking behavior in a simple insect-like robot. There is no need for any complex central control system if each leg simply does the right thing according to the laws of physics (Pfeifer and Scheier, 1999). Furthermore, Brooks (1991a) argues that being embodied and situated changes the need for complex representations. In many cases representations simply gets in the way. Instead the system can rely directly on the real world, using the world, as its own best model. A more thorough description of a complete embodied and situated system as mentioned above, is autonomous agent, summarized by Beer (1995, p. 173):

> "By *autonomous agent*, I mean any embodied system designed to satisfy internal and external goals by its own actions while in continuous long-term interaction with the environment in which it is situated. The class of autonomous agents is thus a fairly broad one, encompassing at the very least all animals and autonomous robots."

As, the phrase "satisfy internal and external goals" suggests, the emphasis here lies on something different than what has been tradition in cognitive science. The goal of an autonomous agent is not primarily to be good at highly intelligent activities seen from a human perspective. What is meant here is simply, in most cases, surviving in a demanding environment and to be able to reproduce in order to ensure the survival of the own species. The goal can also be to carry out a specific task like exploring the surface of another planet or clean the researcher's office. However, Pfeifer and Scheier (1999) point out that *autonomy* is not an all-or-nothing issue, but a matter of degree. They argue that there are no completely autonomous agents – they are all to some degree dependent on external factors, which are out of the agents' control. Hence there can be said to exist, at least two, different kinds of autonomous agents; those which are only dependent on the environment and those which are dependent on other agents as well. For an ordinary organism, the environment is the source of food, drink, oxygen, building material and so on. If the organism is not capable of acquire these resources on its own, it is dependent on other agents. To be dependent only on the environment is of course to be more autonomous, than having to rely on other agents in order to survive or complete a given task. If one, like Brooks, wants to build complete systems, one should also try to design the system as autonomous as possible.

If the traditional view of cognitive science can be described as realist and objectivist in its ontology, this alternative approach could be described as constructivist – following the basic ideas of researchers like Kant, Piaget, von Uexküll (1957, in Ziemke, 2000) and Varela (et al, 1991). According to this view, information from the environment is *mentally presented* to us, instead of *mentally represented*. Hence, there is no need for, like in the traditional approach, to talk about any accurate correspondence between internal model and outside world. What is important here is that an agent's view of the outside world makes sense to the agent itself. Experience and information from the outside world is therefore something highly subjective. Expressed in von Uexküll (1934, in Clark, 1997 and 1957, in Ziemke, 2000) terms; the agent's perceptual world (the agent's subjective sensations) together with its effector world (the agent's possible actions) forms a closed unit, the *Umwelt*. This Umwelt constitutes the agent's *world* and should be considered as real as any other world. Quite naturally, it would be questionable to compare the worldview of a human and a honeybee. A specific situation which, for the human, appears to have a certain meaning, most probably gives a completely different meaning for the honeybee. Hence, it would also be wrong to assume that an artificially constructed autonomous agent should have the same view of the world as the researcher(s) who built it. Maybe one of the most important ideas in the constructivist approach is that the agent's mental world is self-organized and actively built up through interaction with and experience from the environment. Knowledge about the world is not something that is passively mapped with an objective world.

Bickhard and Terveen (1995) and Dorffner (1997) have suggested a form of representation which can handle this highly subjective, self-organized and actively built up, mental world – *interactivist representations*. Dorffner (1997, p. 99) argues that these representations simply are "internal structures on which an agent operate in order to guide its behavior". Furthermore, these internal structures do not need to correspond to any external identifiable structures, what is important is that these representations are "defined only with respect to the agent itself, its drives and needs, and its behavior" (Dorffner, 1997, p. 99). In other words, a specific internal state, representing a dangerous situation, does not need to correspond exactly to some physical features of the external world; it only has to represent a solution for the agent for how to deal with the here and now. It is however, fully possible to have interactive representations encode specific world states in the classical sense, if this is appropriate for achieving certain behavior. But contrary to traditional approaches, it is not necessary. Both Bickhard and Terveen (1995) and Dorffner (1997) emphasize that these representations are actively built up by the agent itself during interaction with the world. This presupposes that the agent is let lose in a real world with real problems, clues and challenges. These kinds of representations lack the problem of grounding its symbols and still have intentionality, if only for the agent itself. Unlike in traditional approaches, there is no outside researcher designing the agent's internal representations and its intentionality[2].

There have been proposed a number of different central control structures for implementing these embodied and situated systems. The pioneer Brooks (1991b) has suggested a subsumption architecture with a number of parallel hardwired behaviors (Figure 3). The positive side of this approach is that it can deal in a fast manner with the here and now. The problem is on the other hand, that it is hardwired, pre-designed and does not adapt. Hence, it can not be said to be self-organized and therefore not fit into a more restrictive constructivist approach.

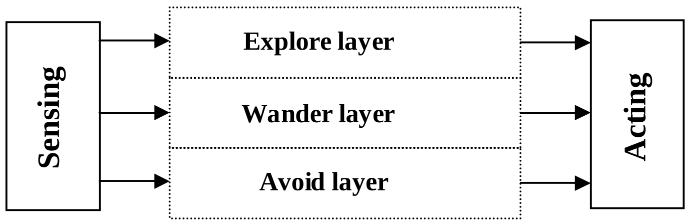


Figure 3: Brooks' subsumption architecture. A number of independent expert layers working in parallel resulting in an overall behavior. Adapted from Brooks' (1986).

Dorffner (1997) on the other hand proposed an approach, *radical connectionism*, under which the appeal of connectionist models can be combined with the ideas of using real embodied and situated agents in real worlds. Artificial neural networks are according to Dorffner well suited for self-organization and

[2] The term intentionality is here only used in the meaning of aboutness.

learning, they can easily handle sensory data and under the right conditions, they offer a solution to the symbol grounding problem. Finally, artificial neural networks offer a more or less biologically plausible solution. These qualities can only come to its rights under the right conditions. In order to be truly radical, connectionist models should focus mainly on self-organization, real sensori-motor interfaces, situated models and autonomous behavior.

There has been some criticism raised against these kinds of approaches to cognition, mainly from stakeholders for a more traditional viewpoint, arguing that simplistic bottom-up models only can quote for low-level cognition and not explain complex cognitive phenomena (Stewart, 1995). Using Clark's (1997) terms, one could say that if traditional approaches have focused on *vertical microworlds* (perception, speech, representations etc.) these new approaches have focused on *horizontal microworlds* (basic but complete behavior like surviving in an environment). Unfortunately, this critique is to some degree valid, most studies following the ideas of researchers like Brooks, Beer and Dorffner (to name a few), have been focusing on very basic aspects of cognition. Strictly speaking, what makes the human so fascinating from a cognitive perspective is not her ability to walk or survive, it is her ability to communicate, form new concepts or use tools etc. If one, like most researchers within traditional cognitive science holds these 'higher' abilities as examples of true cognition, one is probably eager to label the rest 'less important'. One way of dealing with this critique is to study the evolution and see were evolution have spent most of its time. It is only within the last million years that there can be seen any form of higher cognition, the rest of the time evolution has spent on more basic abilities. According to Brooks (1991a), this implies that once the basic and fundamental parts are in place, the rest is trivial. Thus, one could argue that we should spend most of our energy trying to get a good picture of the basic aspects of cognition. Another way of dealing with critique is to improve and develop the own approach, so that it can handle more complex cognitive phenomena. The next section will present a series of theories which explains (higher) cognition as internal simulations of actions and simulations (Clark & Grush, 1999; Grush, 2002; Hesslow, 2002). These theories could very well work as a further extension of a bottom-up constructivist approach, and allow autonomous agents to become more complex.

2.2 Higher Cognition as Internal Simulation of Behavior and Perception

Germund Hesslow (2002) has out laid a theory of how conscious thought, or our introspective experience of an 'inner world', can be explained as simulation of behavior and perception. According to Hesslow (2002), such a theory could rest upon three core assumptions: First, behavior can be simulated by activating motor structures in the brain as during normal overt behavior, but suppressing its execution. Second, perception can be simulated by activation of sensory cortex as in the case of normal perception of stimuli, but without any real sensory input. Third, and perhaps most interesting, there exist associative mechanisms that enable both behavioral and perceptual activity to elicit anticipatory perceptual activity in the sensory areas of the brain. That means, "simulated action can elicit perceptual activity that resembles the activity *that would have occurred* if the action had actual been performed" (Hesslow, 2002, p. 242, original emphasis). This gives the cognizer the ability to explore different options and chose the most beneficial one without having to investigate dangerous and time consuming alternatives in the outside world and perhaps putting its own health at risk (cf. Craik, 1943; Chapter 1).

The following sub-sections will present the simulation hypothesis in more detail and give some scientific evidence that supports its validity (see also Svensson, Lindblom and Ziemke, in press). Furthermore, some similar theories will be briefly discussed. Finally, the simulation hypothesis will be related to cognitive robotics and some previous attempts to partially implement some ideas derived from the simulation hypothesis.

2.2.1 Simulation of Action

There are a number of scientific findings, both neurological and behavioral, that supports the idea that actions can be simulated by activating the motor areas in the brain but suppress the signal leading to execution. One example is that the time it takes to mentally simulate a simple motor task closely corresponds to the time it takes to actually perform the task (Hesslow, 2002). These conclusions were drawn from, among others, a study by Decety, Jeannerod & Prablanc (1989), who showed that their subjects took equal amount of time for finishing a distance walking it blindfolded as imagining walking it. Furthermore, Ingvar and Philipson (1977) showed, using imaging techniques that both simulated and physically performed hand movements activate parts of the pre-motor area of the frontal lobes, whereas only the overt movements activate the primary motor cortex. Normally, the pre-motor areas are considered responsible for indirect motor functions such as coordination etc. while primary motor cortex are responsible for a more direct connection to the rest of the body (Gazzaniga, Ivry & Mangun, 2002)

Another source of evidence, which partially supports the idea that we can activate motor areas in the brain as during normal behavior but without causing any overt behavior, is the discovery of *mirror neurons* in the macaque monkey (Rizzolatti, Fadiga, Gallese & Fogassi, 1996; Svensson et al, in press). In the macaque monkey neurons located in the area F5 (ventro-rostal part of area 6, caudal to the lower arm of the arcuate sulcus) discharge during goal-oriented hand movements such as grasping, holding or manipulating. These neurons also become activated when the monkey observes an experimenter performing the same goal-oriented hand movements. In other words, mirror neurons in the monkey fire both when performing a specific action and when observing someone perform the very same action. This phenomenon has been interpreted by Rizzolatti et al (1996) as vital for understanding motor events, and thus to provide the capacity to "recognize the presence of another individual performing an action, to differentiate the observed action from other actions, and to use this information in order to act appropriately" (p.137).

2.2.2 Simulation of Perception

Similar to simulation of action, there is a large number of scientific findings supporting the idea that we can simulate perception by activating the sensory areas in the brain without actually have any sensory input. A first group of evidence come from behavioral studies; Shepard and Metzler's (1971) *mental rotation task*, which compared actual manipulation of physical objects and the corresponding *mental* manipulation, showed that the time it took for the subject to find a solution, both in the case of physical and mental rotation, closely corresponded to the degree of rotation that had to be done in order for the object to reach orientations at which they could be compared. In both cases it was as if the subject looked internally or directly at the object. According to Hesslow (2002), this led many researchers to draw the conclusion that mental imagery uses the same mechanisms as the visual system. Lee and Thompson (1982) demonstrated in a series of experiments the accuracy with which humans are capable of guiding their behavior based only on internally generated sensory experience. A group of subjects were first allowed to look at their surrounding environment and asked to focus attention to specific objects, such as marks on the floor and different obstacles. The subjects were then blindfolded and asked to perform different tasks in the environment, such as walking to marked location, avoiding obstacles or throwing objects at different targets. The subjects performed almost equally well when they were blindfolded as when they were free to look. One possible explanation of these results are given by Ziemke, Jirenhed and Hesslow (2002), who argue that it is possible that the "subjects actually did 'see', 'walk' and 'throw', but did so internally – in their 'inner world' where sensory experiences and consequences of different behaviors may be anticipated" (p. 3).

Another group of evidence comes from neurological studies and patients with neurological damages. Interestingly, patients with the diagnosis *cortical blindness* have not only lost their visual perception but also the capability to form any kind of visual images. This indicates that we use the same system for both visual perception and mental imagery. The most compelling support for the simulation of perception assumption however, according to Hesslow (2002), comes from contemporary functional imaging techniques. Imaging a visual stimulus or performing a task that requires visualization is followed by increasing activity in the primary visual cortex. The same seems to be true for some specialized visual areas. The primary visual cortex is traditionally considered to handle direct visual input while the secondary visual cortex is said to be responsible for more associative perceptual (primarily visual) functions (Gazzaniga et al, 2002).

2.2.3 Anticipation

What we perceive is often determined by our own behavior. If we move our head or eyes in a new direction that will result in a new visual input, if we reach out to touch something that will result in a tactile input, and so forth. Hesslow (2002) therefore argues that sensory consequences to a large degree are predictable. As a result, it would be possible to assume the existence of some kind of associative mechanisms in the brain, making it possible for both perceptual and behavioral activity to elicit other perceptual activity in the sensory areas of the brain (see Figure 4 a, b). Such elicited perceptual consequence could then be the source of new perceptual activity, creating chains of simulated perceptual consequences and responses (see Figure 4 c).

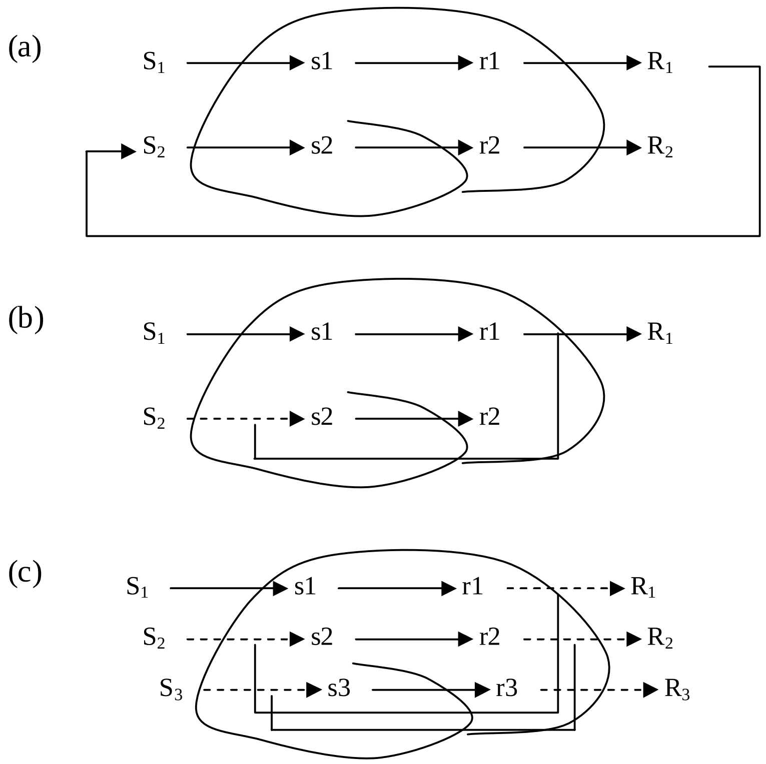


Figure 4: Internal simulations. (a) A situation S_1 elicits activity s1 in the sensory cortex, which in turn leads to a response preparation r1. The response preparation r1 results in the overt behavior R_1 which courses a new situation S_2. (b) A predictable relation between a response and the resulting stimuli allows associations to be formed such that the response preparation r1 directly elicits the activity s2 in the sensory cortex. (c) If internally generated stimuli can elicit a response preparation, it should be possible to simulate long sequences of responses and sensory consequences. Adapted from Hesslow (2002).

One obvious advantage with such anticipation mechanism is that a cognizer would have the ability to interrupt, or not even initiate, an activity which threatens to have dangerous consequences (Hesslow, 2002). This would of course constitute an enormous evolutionary leap, separating animals with higher cognitive abilities from more simple life forms. Such an anticipation mechanism could also be one explanation to our introspective experience of an 'inner world' or possibly consciousness. Moreover, as expressed by Hesslow (2002, p. 246), "[i]f the preparation of a verbal response can generate activity in the auditory cortex or in Wernicke's area, it should be possible to 'hear' it before it results in overt speech and we should be able to speak internally". Thus, chains of internal simulations could be the underlying mechanisms which results in what we experience as an internal dialogue or what we might call 'verbal thinking'.

Finally, as pointed out by Hesslow (2002), there are some interesting features with the simulation hypothesis in general. First of all, it does not require any assumptions about the existence of representations or other mental entities. The simulation hypothesis is solely based on basic behavioral and neural processes. Thus, it does not stand in any fundamental conflict with a constructivist approach to cognitive science. Second, the simulation hypothesis does not assume any big evolutionary leaps. The mechanism necessary for higher cognitive abilities are shared by all mammals, even though they might not be fully developed.

2.2.4 Related Theories

A number of researchers have expressed ideas similar to those of Hesslow (2002); Clark and Grush (1999), Grush (2002), Shanahan (unpublished manuscript) and Svensson et al (in press) to name a few. Grush (2002) and Clark and Grush (1999) have suggested the use of emulators as the basis for cognition in the absence of real-time information and feedback from the world. Clark and Grush (1999) give one basic example of an emulator in the simple process of reaching for an object. When reaching for an object one is dependent on continuously proprioceptive feedback, especially when visual feedback is not available. The problem is however, that such information often is required faster than it is available. The solution is to use an emulator, which provides some kind of quicker mock feedback. The emulator takes as input the current input state of a system (in this case a human being) and tries to predict the next upcoming state of the system. According to Clark and Grush (1999) the use of such emulators makes it possible to "exploit mock feedback ahead of real-world feedback, and hence allows rapid error-correction and control" (p.6). It would also support reasonable sensible behavior even in the absence of real-world feedback. Finally, it allows the cognizer to improve its motor skills off-line without engaging the real-world system at all. These ideas have indeed won support from experimental implementations of emulator's trough so called forward models (chains of feed-forward networks). Hoffman and Möller (2004) showed in a study that a chain of forward models (emulators) trained on visual and motor data obtained from a mobile robot were capable of anticipate or emulate the appropriate sequence of motor commands to find its way from start to goal in an environment. In addition, Clark and Grush (1999) argue that with such simple emulators at hand, they would probably only require a simple evolutionary modification in order for them to be able to run emulation's completely off-line. Such emulations would then be of great importance when it comes to planning, mental imagery and such.

Clark and Grush's (1999) description seems very much like the anticipation mechanism presented by Hesslow (2002). Their emulators are, like Hesslow's, based on the idea that what we perceive (in this case mostly through proprioception) and experience to a high degree are possible to predict. Both Clark and Grush (1999) and Hesslow (2002) provide examples of mechanisms which make it possible for the cognizer to behave in a reasonable sensitive way, even in the absence of real-world feedback. Furthermore, both mechanisms provide to basis for higher cognitive functions, and do so based on simple, evolutionary plausible functions.

Clark and Grush (1999, p.7) also express the idea that emulations would "constitute the most evolutionary basic scenario in which it becomes useful to think of inner states as full-blooded representations of extra-neural (…) states of affairs". In these terms, emulations or simulations (even though Hesslow does not express any need for such mental entities) could be seen as one kind of representations. Such representation would then stand in great contrast to the kind of mental representations favored by traditional cognitive scientists, as they do not correspond to any current situation or single object in the world, but actually precede the current state of affairs. Such representations would however fit the previous discussed description of interactivist representations given by Dorffner (1997, p. 99), "internal structures on which an agent operates in order to guide its behavior". It is thus possible to argue that both the simulation hypothesis and the ideas presented by Grush and Clark and Grush are compatible with a constructivist approach.

2.2.5 Open Questions

Although the simulation hypothesis offers a biologically and evolutionary plausible theory which is supported by a wide range of scientific findings there are still a number of open question which remains unanswered. Some of these are nicely summarized by Svensson (2002); First of all, what are the differences between simulations of perception and real perception? As pointed out by Thomas (1999, in Svensson, 2002) there must be a difference between mental imagery and perception, otherwise, a person would not be able to distinguish their perceptions from their imagination resulting in hallucinations. Second, how do simulation and emulations relate to memory mechanism? It is reasonable to assume that long-term memory and working memory plays an important part in anticipation. Finally, and most important for this dissertation, at what level of abstraction does these simulations take place? Does it deal directly with sensory information or do simulations work with higher abstract concept? The following section will present an experiment in which Jirenhed (2001) and later Jirenhed, Ziemke & Hesslow (2001) tried to partially implement simulations of perception in a mobile robot. In this case the simulations were carried out at sensory level. As this turned out to be problematic, it directs interest to the question mentioned above.

2.2.6 Simulations in a Cognitive Agent?

There have been some attempts to partially implement Hesslow's (2002) simulation hypothesis. Jirenhed (2001) and later Jirenhed et al, (2001) tried to develop an autonomous agent which was capable of simulating its own perception, using a very minimalistic model. An ANN-controlled Khepera robot was trained to predict its own sensory activation in the next time step. Jirenhed et al's (2001, p.108) hypothesis where that "If the robot's sensory predictions [were] sufficiently accurate (...) then the robot should be able to use them instead of actual sensory input, and thus behave appropriately, at least for some time, in the absence of external stimuli".

The study was divided into three phases where the first phase concerned rudimentary navigation, the second phase sensory prediction and the last phase perceptual simulation. In the first phase a simple recurrent network with 8 input nodes, 3 hidden nodes and 2 output nodes where used. The agent was simply trained, with positive result, to navigate in two different environments, one 'h' shaped and one 'T' shaped world. In the second phase, 8 additional output nodes were added (see Figure 5). The agents were now not only trained to navigate in the environment, but also to predict the sensory activation in the next time step. Also in this case the results were positive.

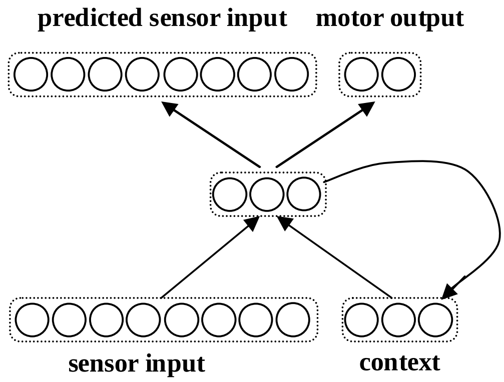


Figure 5: The simple recurrent network used by Jirenhed (2001). The network takes the 8 sensor values as input each time step and tries to predict the next upcoming sensory readings and the appropriate motor activations. Adapted from Jirenhed (2001).

The ANN was to a high degree capable of predicting the next sensory state. However, the network seemed to focus on predicting the most frequent input and ignored less common and weaker input. In the third phase, the network's own sensory predictions were fed back into the network in the upcoming time step (see Figure 6). The network where simply cut of from the world and asked to navigate only based on its own 'inner world'.

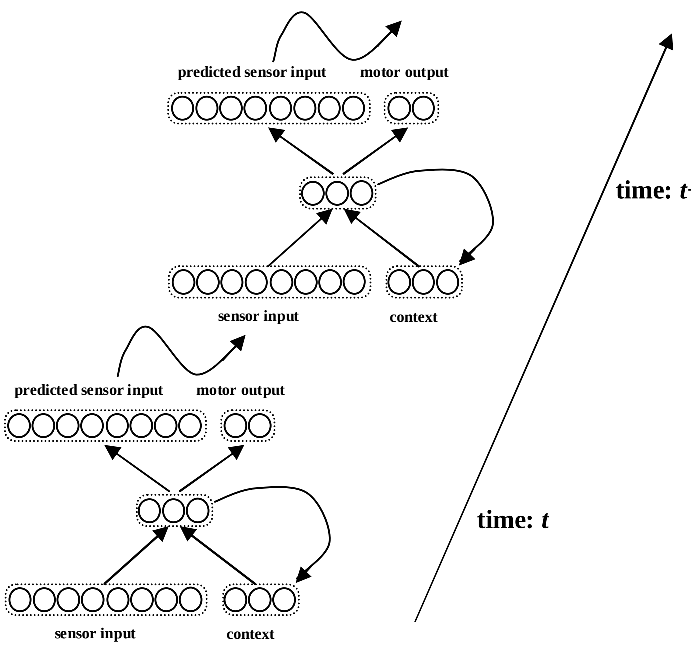


Figure 6: Jirenhed's architecture trying to simulate perception. At each time step *t+1* the predicted sensor values from time step *t* are given as input. Adapted from Jirenhed (2001)

In this case the result was negative. Although the robot seemed to have developed some important internal dynamics the robot were not capable of simulating its own perception. The robot succeeded in avoiding collisions but did simply not navigate in any reasonable way. There are many possible reasons for this result. Jirenhed et al. (2001) give one explanation being the robots limited sensor range. When the robot detects an obstacle there is not enough time left to act in an appropriate way. This hypothesis was indeed tested with a positive result in a later study by Ziemke, Hesslow and Jirenhed (in press). The robot's infrared sensors were replaced by long-range sensors, which allowed the agent to in a better way keep track of the distance to walls and objects. In this case however, a simpler square shaped environment was used, reducing the complexity of the experiment to an absolute minimum. This might also reduce the importance of the result, and raise the question if it is possible to scale up the complexity. Another explanation given by Jirenhed et al. (2001) is a that the ANN architecture was too simple and in need of more modules or more hidden nodes, "the somewhat minimalist architecture and training regime used in [this] experiments (…) obviously have their limitations, nevertheless they might serve as a useful starting point for experiments with other architectures". Yet another explanation, which are of importance for this dissertation, is that the network used by Jirenhed et al. (2001) only learned to predict the most frequent inputs and ignored changes. This comes quite naturally, since it is much easier for the network to get a good reward during training if it focuses on inputs which are easy to predict and occurs often. This effect is also increased by the fact that the network are updated every 100 *ms*. In one tenth of a second the robot's input does not change much. This will result in long series of input with almost the same characteristics and make changes even harder to predict. This will once again raise the question mentioned by Svensson (2002); on what level of abstraction do these simulations occur? Maybe sensory input is to complex and contains too much information for it to be possible to simulate internally. Perhaps it would be a better approach to abstract the sensory information into higher concepts corresponding to situations, and let these be simulated. The next section will discuss some techniques and architectures for extracting concepts from the environment and how to use them to predict the next state of a system.

2.3 Knowing your environment through abstractions

The experiments performed by Jirenhed at al (2001) indicated that the sensory level might be the wrong level of abstraction for simulating perception. Therefore, this section will present some techniques for how to extract concepts or events from a sensory flow. Moreover, this section will discuss how such extracted concepts or events can be used at a higher level to anticipate a future situation. Finally, a technique for how abstract representations can be easily analyzed by inverting stored chains of concepts will be presented.

2.3.1 A cascade of Prediction Networks

Nolfi and Tani (1999) investigated in a study how prediction learning could be used to extract regularities from the external environment in the case of a mobile robot navigating in a simple environment divided into two rooms (see Figure 7). A neural network divided into several *levels* were trained to at each level predict the internal state of the previous level when such states change significantly (the first level were trained to predict the activation of the distance sensors on the robot in the next time step). The basic idea is that this architecture will a) progressively re-code sensory information and enhance useful regularities and filter out useless information, and b) reduce sequence length, which in turn will have higher layers extract higher level regularities. In other words, lower layers extracts low level regularities like walls, corners and corridors and higher layers extracts high level regularities like “the left side wall of the large room” (Nolfi & Tani, 1999). It should be pointed out that the networks’ task were not to control the behavior of the robot, only make predictions of the next internal state.

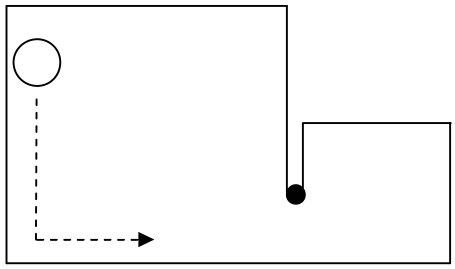


Figure 7: The environment used by Nolfi and Tani (1999). Two rooms, 40 x 40 and 20 x 20, respectively are joined by a short corridor. The big empty circle symbolizes the robots current position and the dashed arrow its direction. Adapted from Nolfi and Tani (1999).

A schematic illustration of the layered network is shown in Figure 8. The network used by Nolfi and Tani (1999) only made use of two levels of prediction networks, but it is fully possible to extend the architecture with more levels. Each level also contained a segmentation layer which compresses sub-sequences of homologous states into a single state. The result of this is shorter and shorter sequences which will, as previously mentioned, let the higher level extract high level or long term regularities. The segmentation layer works as a self-organizing map, but with the difference that only the actual node is updated during training, and not the whole neighborhood (Nolfi & Tani, 1999). The number of possible segmented states is pre-defined by the choice of number of output nodes in a winner-take all network. In this particular case there were three output nodes in the main study.

The network was implemented in a robot with a pre-programmed wall-following behavior and then trained on the simple domain shown in Figure 7 using backpropagation. The training was divided into three phases. In the first phase the first level were trained for 100,000 time steps. In the second phase the first levels segmentation layer were trained for 100,000 time steps. In the final phase the second level were trained for 10,000,000 time steps. But since the second level took as input the segmented state of the previous level and only should predict when the state change significantly, the effective training lasted for about 530,000 cycles. The result showed that both levels had learned their prediction task well (prediction error around 0.05 and 0.04, respectively). The result also showed that the segmentation layer had produced the segmented states; (a) patterns experienced while the robot is doing wall following, (b) patterns experienced while the robot is turning along a corner and (c) patterns experienced while the robot is traveling along a corridor.

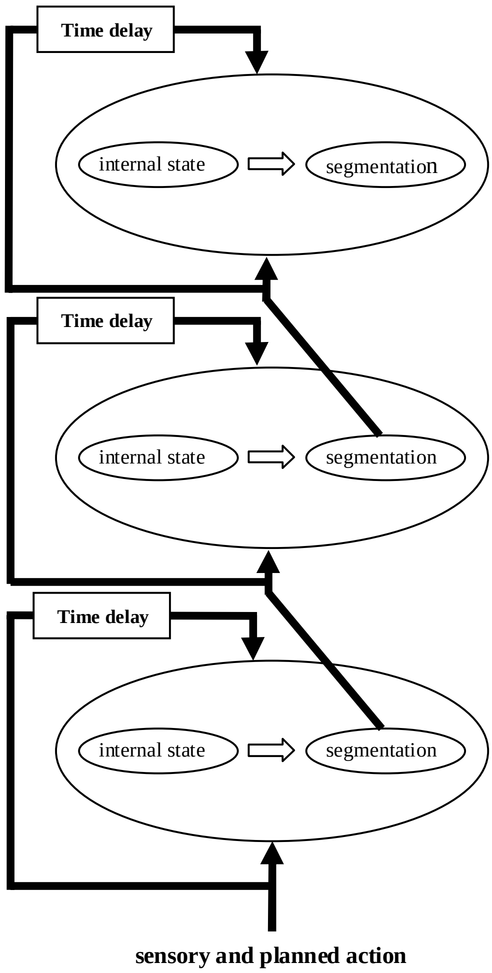


Figure 8: Nolfi and Tani's (1999) hierarchy of prediction networks. At each prediction level the information in the internal state are segmented and fed as input the next prediction layer in the hierarchy. Adapted from Nolfi and Tani (1999).

The robot was also tested in three different variations of the training domain. These result showed that the prediction error for the first level didn't change significantly. According to Nolfi and Tani (1999) this implies that the first level doesn't need to extract any knowledge from the environment in order to complete its task. The second level on the other hand showed big variation in the prediction error between different environments. This result implied that the second level had extracted internal representations which included the topological structure of the environment (Nolfi & Tani, 1999). A further hypothesis was that the robot, with help from the internal representation, should be able to localize its own position in the environment. In one additional experiment the robot were randomly placed in the original environment each 1000 time step. At first the prediction error increased significantly, but after a few time steps the prediction error dropped again as the robot had located its own position. From this Nolfi and Tani (1999) draw the conclusion that the robot not only had extracted an internal representation which included the topological structure of the environment, but also a dynamical representation which included the robot’s own relative position to the environment.

What maybe most interesting to note here is that the internal representation is a high level abstraction from the environment and yet it contains enough information for the robot to complete its tasks. This subject will be further discussed in later chapters.

Although not explicitly mentioned in Nolfi and Tani (1999), it is possible to argue that these predictions of the next internal state partially correspond to Hesslow's (2002) ideas of simulation of perception. The task of the network is to predict what the world might look like in the next time step or situation. If the second level received its own predictions as input in the next time step, it might be possible to simulate perception. It is also possible to argue that these simulations of perception bares representational burden as Clark and Grush (1999) argues. Although the robots internal state actually precedes its actions it still helps the robot to locate its own relative position to the environment.

2.3.2 How to Extract Concepts from the Environment

Linåker and Niklasson (2000a) presented a novel architecture for unsupervised learning which is based on change detection rather than traditional error minimization. They also put their architecture, called *adaptive resource allocating vector quantization* (ARAVQ), in contrast to a similar system for classification presented in Nolfi and Tani (1999) (see section 2.3.1, this chapter). Linåker and Niklasson (2000a) argue that error minimization, which were used in Nolfi and Tani's (1999) architecture, is not well suited for a sensory flow segmentation task (like for instance the sensori-motor flow in a robot). The major problem with error minimization is the inherent sensitivity to input pattern densities. It is most commonly the case that the network focuses on the most frequent input patterns and handles less common, but maybe important, patterns as noise. According to Linåker and Niklasson (2000a), these effects were indeed experienced by Nolfi and Tani (1999) but these problems were somewhat remedied by the fact that they used a prediction network.

The ARAVQ focuses on detecting changes in the input signal characteristic. At each time step t, the ARAVQ takes as input the robots sensory readings from the environment and the current motor activation. These inputs $x(t)$ are at each time step buffered in a finite moving average $\bar{x}(t)$ of length n (equation 1)(Figure 9).

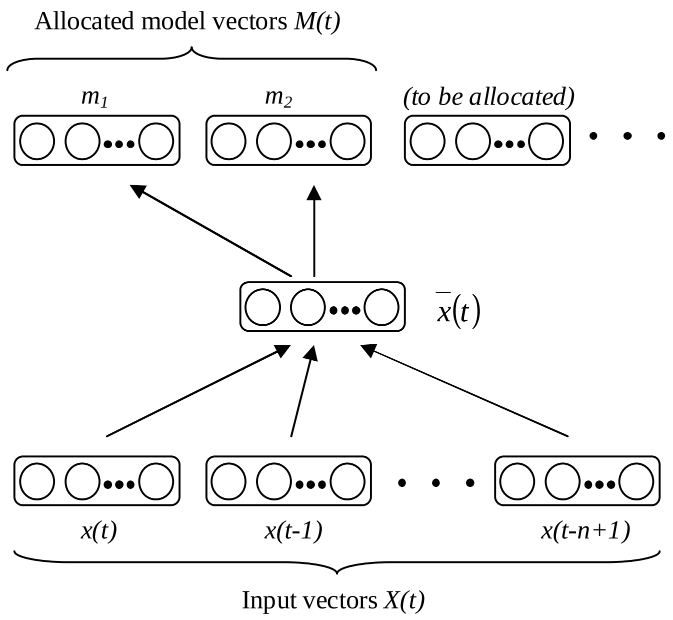


Figure 9: The ARAVQ. The *n* last input patterns are stored in the finite moving average. At each time step *t* the finite moving average are compared to the stored model vectors. If there is a close match that model vector is selected, otherwise a new vector can be stored if both the stability and the threshold criteria is fulfilled. Adapted from Linåker and Niklasson (2000b).

In this way the finite moving average represents the state that the system is in right now – capturing both some of the systems history and filters out unwanted noise.

Equation 1: calculation of the finite moving average.

$$\bar{x}(t) = \frac{1}{n}\sum_{i=0}^{n-1} x(t-i)$$

Essential for the ARAVQ is a set of stored model vectors *M(t)*, each one corresponding to or representing a higher concept or situation in the world (Figure 9). At first the set *M(t)* and the corresponding model vectors $m \in M(t)$ are empty but as the sensory information starts to flow new model vectors can be added. For this to happen, a number of criteria's has to be fulfilled. First of all, the mismatch between any stored model vector and the finite moving average has to be above a threshold *δ*. Second of all, the mean distance between the last *n* signal values and the finite moving average must be stable, i.e. stay below a stability criteria *ε*. If both these condition are fulfilled a new concept can be added. From time step *n-1* the ARAVQ starts to compare the finite moving average with all stored model vectors (if any). This is done by creating a distance metric *d(V, X)* which defines the distance between stored model vectors $v_j \in V$ and the last set of input vectors $x_i \in X$ (equation 2), where || . || denotes the Euclidean distance.

Equation 2: Euclidian distance.

$$d(V,X) = \frac{1}{|X|} \sum_{i-1}^{|X|} \min_{1 \le j \le |v|} \left\{ \left\| x_i - v_j \right\| \right\} \; ; \; x_i \in X,\; v_j \in V$$

This distance metric can be used to calculate the mean distance $d_{\bar{x}(t)}$ from each of the last *n* signal values to the moving average $\bar{x}(t)$ (equation 3).

Equation 3: calculate mean distance from finite moving average.

$$d_{\bar{x}(t)} = d\left(\left\{\bar{x}(t)\right\}, \left\{x(t), \mathrm{K}\ , x(t-n+1)\right\}\right)$$

It can also be used to calculate the distance $d_{M(t)}$ between each of the last *n* signal values and the best matching model vector at each time step (equation 4).

Equation 4: calculate distances between *M(t)* and the *n* last signal values.

$$d_{M(t)} = \begin{cases} d\left(M(t),\ \{x(t)\ldots,x(t-n+1)\}\right) & |M(t)|>0 \\ \varepsilon+\delta & |M(t)|=0 \end{cases}$$

If the ARAVQ just has started working and *M(t)* is empty, the distance $d_{M(t)}$ is set to be sufficient for incorporating the moving average into this set of model vectors. Here this is done by setting the mean distance between then last signal values and the moving average to be below ε and the mismatch between the finite mowing average and the non existing model vectors to be above the threshold δ. If there already is some model vectors in *M(t)* the criteria's work as usual (equation 5).

Equation 5: compare finite moving average and signal values to stored model vectors.

$$M(t+1) = \begin{cases} M(t)\cup\bar{x}(t) & d_{\bar{x}(t)} \le \min\left(\varepsilon, d_{M(t)}-\delta\right) \\ M(t) & \text{otherwise} \end{cases}$$

Finally, at each time step, the model vector which is the closes match to the current finite moving average is selected as a winner *win(t)* (equation 6). This model vector hopefully corresponds to the situation the robot is in right now.

Equation 6: select the winning model vector.

$$win(t) = \arg \min_{1\le j\le |M(t)|} \left\{\left\|\bar{x}(t)-m_j\right\|\right\}\ ;\ m_j \in M(t)$$

Since a new model vector only is initiated in roughly the right neighborhood of an objective concept, they can be further adopted. This is done when a particularly good match comes around. The definition of a good hit is if the stability between the n last signal inputs and the finite moving average is below 0.5ε. If it is a good hit the distance between the winning model vector and the finite moving average is reduced by the learning rate α (equation 7).

Equation 7: adopt the winning model vector in the case of a good hit.

$$\Delta m_{win(t)} = \begin{cases} \alpha\left[\bar{x}(t) - m_{win(t)}\right] & \left\|\bar{x}(t) - m_{win(t)}\right\| < \dfrac{\varepsilon}{2} \\ 0 & \text{otherwise} \end{cases}$$

By adjusting the different variables ε, δ, n and α it is possible to control the sensitivity of the ARAVQ. Very liberal values on ε and δ will generate a lot of model vectors and vice versa. A high value on n will filter out a lot of unwanted noise but perhaps lose some important information along the way. It is simply the case of trail and error to find a good configuration.

In the case of classifying sensory input pattern in a robot these matching vectors closely corresponds to Nolfi and Tani's (1999) segmented states or concept. However, in contrast to Nolfi and Tani's (1999) segmentation architecture the number of possible concepts does not need to be pre-specified, instead new matching vectors or concepts is dynamically allocated (Linåker & Niklasson, 2000).

Experimental results from a comparative study in which the ARAVQ were implemented in a robot under similar environmental settings as in Nolfi and Tani's (1999) study showed that the network succeed to find all obvious concepts (from an observers point of view) and more importantly it did so in just one epoch. Nolfi and Tani (1999) trained their segmentation layer for 100,000 time steps. It should here explicitly be pointed out that Linåker and Niklasson's study had nothing to do with sensory prediction and that these results only concerns classification and abstraction of higher concepts. However the ARAVQ network and its classification abilities in combination with a prediction network similar to Nolfi and Tani's might indeed be of great interest. To make this idea even more appealing the next section will present a later study by Linåker and Niklasson (2000b) were they show how abstracted model vectors in an ARAVQ and some measurement of time can be used to invert abstract internal representation back to an accurate model of the external environment.

2.3.3 Analysis of higher Concepts through Inversion

As previous studies by both Nolfi and Tani (1999) and Linåker and Niklasson (2000a) have shown, unsupervised techniques are very useful to extract regularities from input sensory flow and to, in Nolfi and Tani's case, re-code information so that long term regularities can be extracted at higher levels. Linåker and Niklasson (2000b) present a technique for inverting such extracted abstractions in an ARAVQ network. The use of such technique might be interesting when it comes to analyzing the internal states of a neural network. Nolfi and Tani for instance showed in their study that their architecture based on a hierarchy of prediction networks must have extracted a dynamical representation which included the robots own relative position to the environment. But they could not show exactly what the internal representations looked like, only that it must have done it. If it would be possible to invert such a representation it would be great advantage for further development in the same research field.

To invert the extracted abstractions Linåker and Niklasson (2000b) used the reference model vectors extracted by the ARAVQ (c.f. Figure 3) implemented in an autonomous robot. Each model vector can be associated with a specific signal characteristic which gave rise to the model vector in the first place (trough the occurrence of novel and stable input characteristics in the sensory flow). Since this signal characteristic includes both sensory flow and motor activation it is fully possible to calculate both the typical sensor and motor activation for each model vector (see Figure 10 and 11). Linåker and Niklasson (2000b) simply used a counter to keep track on the number of time steps in a row a specific model vector was active. When another model vector became active the counter re-started. In that fashion it is possible to extract a sequence of model vector winners (see Figure 12).

	sensor (1-8) and motor activation (LM, RM)									
Vector	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**LM**	**RM**
a	*0.00*	*00.0*	*0.00*	*0.00*	*0.09*	*0.75*	*0.00*	*0.00*	*0.75*	*0.75*
b	*0.00*	*0.00*	*0.00*	*0.00*	*0.01*	*0.26*	*0.00*	*0.00*	*0.75*	*0.61*
c	*0.00*	*0.04*	*0.99*	*1.00*	*0.99*	*0.81*	*0.06*	*0.00*	*0.40*	*0.75*
d	*0.00*	*0.00*	*0.01*	*0.11*	*0.94*	*1.00*	*0.23*	*0.20*	*0.40*	*0.75*
e	*0.99*	*0.55*	*0.00*	*0.00*	*0.09*	*0.78*	*0.00*	*0.00*	*0.75*	*0.75*

Figure 10: Model vectors allocated by the ARAVQ. Adapted from Linåker and Niklasson (2000b).

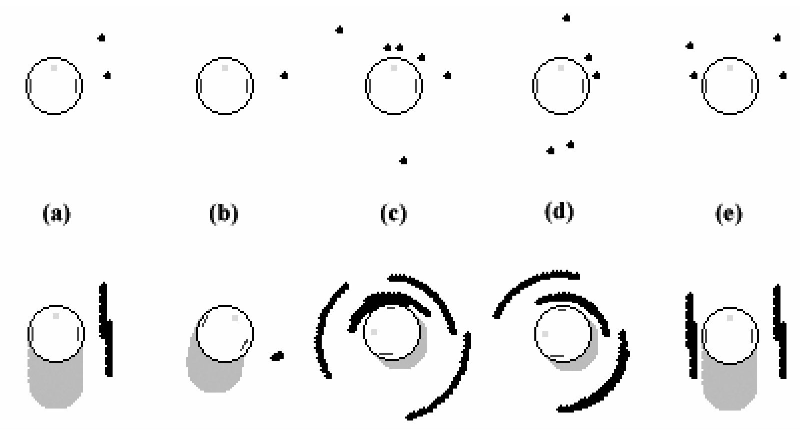


Figure 11: Typical sensor readings and motor activation for each model vector. Reprinted with permission from Linåker and Niklasson (2000b).

The sequence of model vector winners and the typical sensory readings and motor activation for each model vector makes it possible to invert the robots internal states. The robot is simply put in “motion” in a way that each inversion of the model vectors indicates for the number of time steps specified in the extracted sequence of model vector winners. The result in Linåker and Niklasson’s case is a detailed model of the external environment (see Figure 13) which shows to which striking degree the robots internal representation has captured the structure of the domain on which it were trained.

$$a_{233}c_{15}d_{12}a_{18}b_{19}a_{7}b_{53}a_{52}c_{16}d_{13}a_{17}a_{237}c_{16}d_{12}a_{15}b_{12}a_{75}e_{131}a_{106}c_{15}d_{13}a_{14}b_{15}$$
$$a_{176}c_{16}d_{12}a_{14}b_{18}a_{17}c_{17}d_{12}a_{17}b_{17}a_{3}b_{110}a_{64}b_{36}a_{102}d_{10}a_{36}c_{15}d_{12}a_{16}b_{16}$$

Figure 12: Sequence of model vector winners. Adapted from Linåker and Niklasson (2000b).

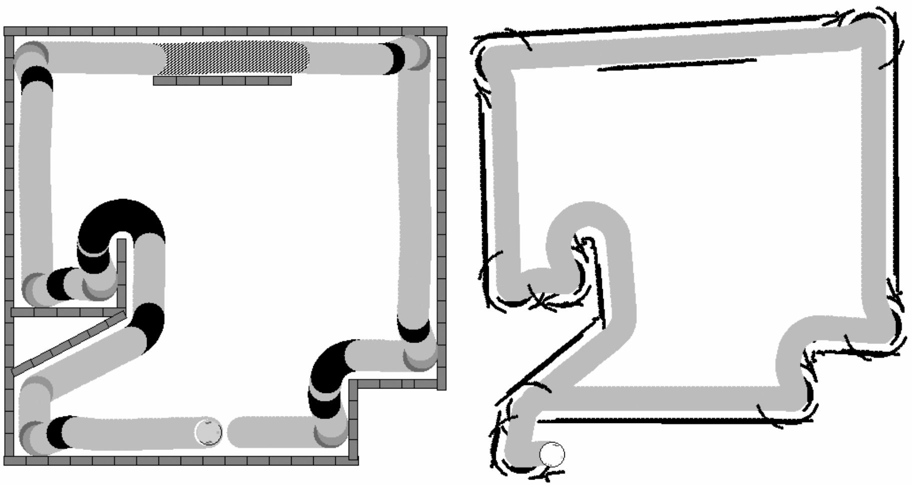


Figure 13: (left) the original environment and the distribution of the 5 different concepts found by the ARAVQ. (right) an inversion created from the sequence of model vector winners. Reprinted with permission from Linåker and Niklasson (2000b)

What is interesting to note here is that abstract representations contains so much information. As shown in Nolfi and Tani, their robot managed to solve quite complex task only with help from such extractions. The results in Linåker and

Niklasson show a compression rate of information around 1:41, from sensor flow to extracted model vectors and the sequence of model vector winners, which is needed to perform the inversion. Newer approaches in AI do in many cases regard realistic and direct sensory information as crucial for cognition. But the informational bombardment might also give rise to a representational bottleneck. This might indeed be the case in the experiments of Jirenhed (2001).

3 Problem Description

Jirenhed (2001) tried to develop an autonomous agent with the task to simulate its own perception at sensory-level. This quite straightforward approach turned out to be problematic. The robot was not able to rely on the simulated sensory values in absence of real sensor value, in order to navigate in the environment. One possible reason for this failure is that simulation of sensor values is the wrong level of abstraction. First of all, real world dependent sensory values in general are full of noise and contains to much irrelevant information and second of all, the sensor values – which are updated every 100 ms – does not change much from time step to time step. This might have the effect that the robot is good at simulating sequences of common sensor input but bad at simulating changes. As mentioned by Linåker and Niklasson (2000a), this is particularly the case when the robot is trained using error minimization since focusing on similarities early in the development will give high fitness. Hence, solutions which are good at detecting changes will be lost. Therefore is this thesis going to use a different approach to implement the simulation hypothesis. In general terms, this thesis will try to simulate perception at a more abstract level. The two upcoming sections will specify the aims in more precise terms and how these aims should, if possible, be fulfilled.

3.1 Aim

The concrete aim of this thesis is to investigate if it is possible to simulate perception in an autonomous agent at a more abstract level using a de-coupled hierarchy of prediction networks similar to the one used by Nolfi and Tani (1999). This thesis will however, use the ARAVQ architecture presented by

Linåker and Niklasson (2000a) at the lowest level in order to handle sensorimotor information flow and to make abstractions from the environment. The higher level of the hierarchy is thus intended to work on the abstracted information in order to simulate perception (Figure 14). As one way of analyzing the performance of such architecture the inversion technique presented by Linåker and Niklasson (2000b) will be used. Hopefully this will give the opportunity to study the robots internal representations in detail. Nolfi and Tani (1999) faced the very same type of representation but without such a tool. Hence, they were only capable of saying that the robot *must have* developed quite complex representations, not what they might look like.

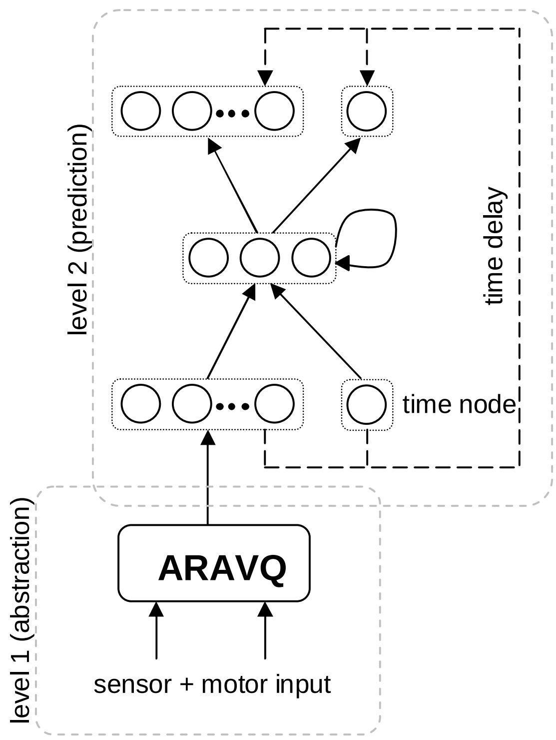


Figure 14: Linåker and Niklasson's (2000a) ARAVQ combined with Nolfi and Tani's (1999) hierarchy of prediction networks. Level one is intended to extract concepts from the environment while level 2 tries to predict the next active concept and the number of time steps during which it is activated. The number of input and output nodes in level 2 corresponds to the number of concepts found by the ARAVQ plus one additional for time measurement.

3.2 Objectives

In order to test if it is possible to simulate perception using this kind of architecture a number of experiments should be done. The general experimental setup is based on Nolfi and Tani's (1999) study (e.g. environment and type of robot). More details on the experiments will be presented in the next chapter.

- As a first step should Nolfi and Tani's (1999) experiment be partially replicated in order to validate some basic functionality of the architecture. This will answer the question if this novel architecture is equally good as Nolfi and Tani's at solving the prediction task, and it will indicate if this architecture has developed the same kind of complex dynamical representations, which allowed their robot to locate its own relative position to the environment.

- If the architecture should prove to be equally successful as Nolfi and Tani's, it should thereafter be de-coupled from the world and try to simulate its own perception at an abstract level. This will be done by feeding the second level predictions back as input at the next time step (Figure 15).

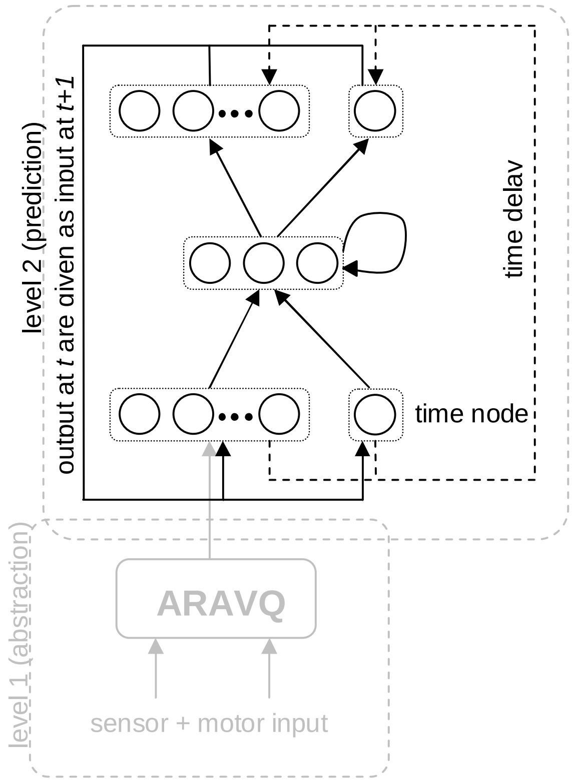


Figure 15: Simulation of perception. At each time step *t* the networks own predictions from the previous time step *t-1* are feed back as input to level 2.

- It should there after be possible to invert the abstracted representations from the ARAVQ, in both the case of simple predictions and simulations, using the time measurements in the second level. This will give an idea of the accuracy of the robots predictions and simulations.

- Finally, if the previous experiments turns out to be successful and the internal simulations accurate, the robot should be 'blindfolded' and let lose in the environment, only relying on its own simulated perception. Although this is a somewhat unrealistic condition for a situated agent, it will be a good benchmark for how well simulations can be performed at an abstract level.

4 Experimental Setup

This chapter describes the setup for the experimental investigations which were performed in this dissertation in order to validate the hypothesis presented in the previous chapter. Section 4.1 presents the materials and environments used in the experiments. Section 4.2 discusses the implementation of the architecture and the algorithms used in the experiments. Finally, 4.3 give an overview of the strategy and goals of the experiments.

4.1 Materials and Environments

The robot used in these experiments is a simulated version of the miniature mobile Khepera robot manufactured by the K-Team (www.k-team.com). The Khepera robot has a cylindrical shape with a diameter of 55mm, a height of 30mm and a weight of 70g (Figure 16). It is equipped with eight short-range sensors, distributed around the body, six around the front in a semicircular way and two at the back (Figure 16). The short-range sensors are capable of detecting an object at a distance of approximately 50mm. Furthermore, the robot is equipped with two motors which can drive the two wheels forward or backwards independent of each other.

The Khepera robot is very popular in the autonomous agent research community thanks to its convenient size and the possibility to add extra modules featuring grippers, extra sensors and cameras. These are however not of interest in this study and will not be further presented here.

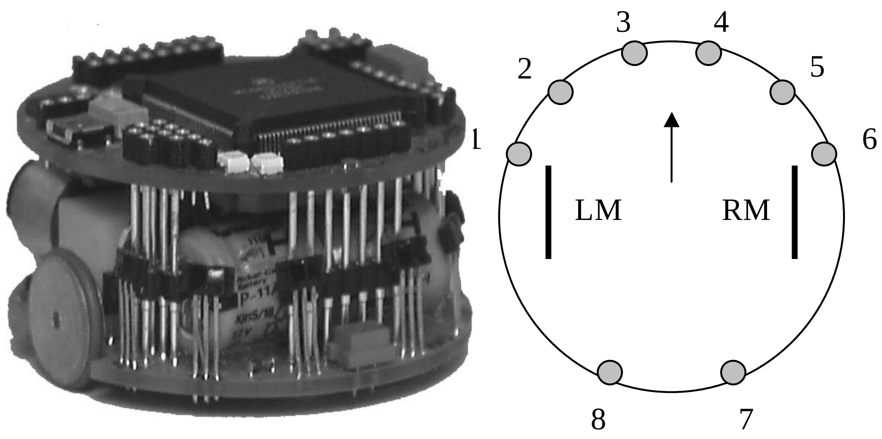


Figure 16: (left) a photo of the Khepera robot manufactured by the K-team (www.k-team.com). (right) a schematic illustration of the Khepera robot. The robot is equipped with 8 short-range sensors, distributed around the body, and 2 motors. The small arrow point towards the front of the robot.

A common problem in robot experiments is that it can be very time consuming. To perform experiments in real-time may take days. One solution which is widely used is to run the experiments on a computer-based simulator. In this way the time it takes to run an experiment are limited by computational power instead of natural constraints such as maximum speed of a robot or limited materials (physical robots, environments etc.). Thus, an experiment which in real-time takes days only takes a couple of hours in a simulator. Due to time limitations and the fact that Nolfi and Tani (1999) used a simulator in their work, all experiments in this thesis will be performed in a simulator. The specific simulator used here is YAKS (Yet Another Khepera Simulator) which were developed at the University of Skövde (Carlsson & Ziemke, 2001). YAKS support several different training techniques (evolutionary algorithms, back-propagation etc.) and allow easy development of environments and robot-control mechanisms.

Four different environments are used in the experiments (Figure 17). One main environment (A) and tree additional experimental environments (B-D) which are intended to be used in order to test the robots internal representations in one of the experiments. The main environment consists of two rooms, one 50 × 50cm and one 25 × 25cm, and is joined together by a short corridor.

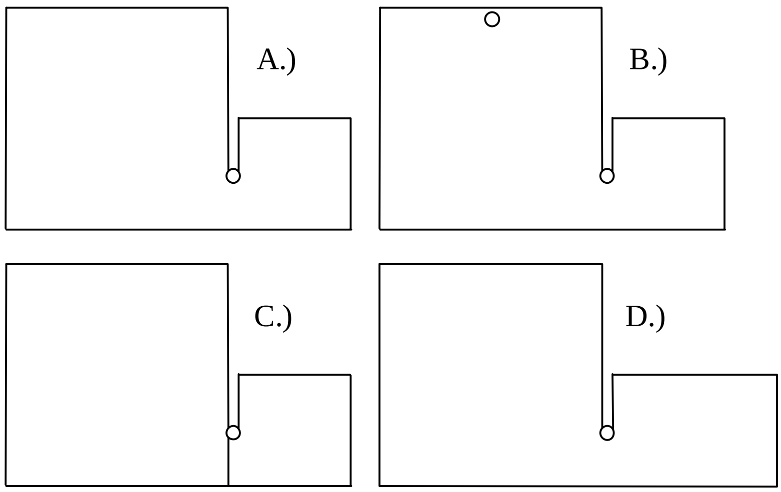


Figure 17: The environment used in the experiments. (A) The standard environment, (B) the standard environment with an additional obstacle, (C) an extra wall sealing the corridor to the small room and (D) extra long small room. Environments are adapted from Nolfi and Tani (1999).

4.2 Algorithms and Architectures

In order to navigate in the environment, the robot is equipped with a pre-trained feed-forward ANN providing a wall-following behavior (Figure 18). The ANN takes as input the activation from the eight short range sensors and gives as output the appropriate motor activation for the right and left motor (Figure 18).

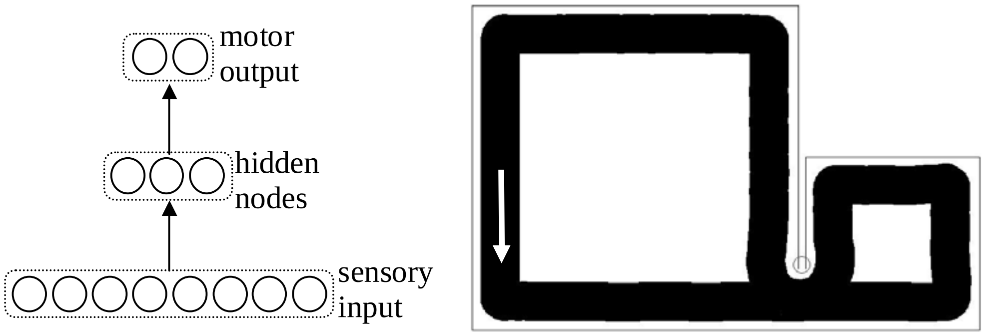


Figure 18: (left) the pre-trained feed-forward ANN used to control the robots behavior. (right) the behavior of the ANN-controlled robot in the standard environment. The robots direction is indicated by the white arrow.

As discussed in Chapter 3, the architecture used in this thesis is a combination between Nolfi and Tani's (1999) hierarchical architecture and Linåker and Niklasson's (2000a) ARAVQ. The complete first level (including segmentation) in Nolfi and Tani's architecture are simply replaced with an ARAVQ which extracts novel and stable concepts from the environment. The ARAVQ which were developed for this thesis is completely based on the algorithms provided by Linåker and Niklasson (2000a). Since there are eight short-range sensors and two motors the ARAVQ will take a total of ten inputs. The second level of the architecture will have a number of input and output nodes equal to the number of model vectors found plus one for encoding time. Since the idea is that the second level should be able to predict the next upcoming concept each of the input and output nodes minus one correspond to a model vector.

At each time step *t* when the ARAVQ does not change concept, the time variable is increased with 0.1 (Figure 19). The value 0.1 are chosen since it corresponds to the length of each time step in the simulator (100ms). If the ARAVQ changes concept, the second level output is compared to the real target values and the new concept and time are feed as input to the second level. Thereafter the time variable is reset to 0.

```
/* initialize network */
time=0;
lastPredictedConcept=currentConcept;
setInput(currentConcept, time) to ANN;
/* run ANN for nrOfTimeSteps */
for (for all time steps){
  /* if the ARAVQ have changed concept */
  if (currentConcept != lastPredictedConcept){
    time=time+0.1;
    compare output from ANN with
      target concept and time;
    give fitness to individual;
    setInput(currentConcept, time) to ANN;
    time=0;
    lastPredictedConcept=currentConcept;
  /* if no change of concept */
  } else {
    time=time+0.1;
  }
}
```

Figure 19: The general strategy for the second level in the architecture. At each time step the output from the ARAVQ are compared to the last predicted concept. If there is no change the time variable is increased with 0.1. If there is a change the networks output are compared to the target values and a new set of inputs are given based on the new concept and the time variable.

Since all nodes have a logarithmic activation function, the time output is multiplied with 10 in order to be able to encode sufficiently large numbers. At each concept change, the highest activation on the outputs encoding the model vectors is regarded as the architectures prediction of the next concept. The number of nodes in the hidden layer will be altered by trial and error to give a good solution.

4.3 General Strategy

The architecture will be trained in two phases. In the first phase the ARAVQ will be trained on the standard environment for 3000 time steps (each one corresponding to 100ms). This approximately corresponds to seven laps in the main environment. The model vectors found by the ARAVQ will then be stored for use in the training of the second level. From now on all the ARAVQ will be doing is to select the current model vector, no further training or discoveries of concepts will occur. During the second phase, the second level will be trained. At each time step, where one time step corresponds to the length of one active concept, *t* the network will be trained to predict the active concept and activation time of that concept at time step *t+1*. The second layer will be trained using a number of different training techniques, evolutionary algorithms[3] (EAs) and backpropagation. The reason for this is simply that it is a complicated task for the network to learn and there are really no good way of knowing which technique is best. Nolfi and Tani (1999) used simple backpropagation in their experiments and will therefore be the first choice. If however this will turn out to be unsatisfying this thesis will not be limited by this. If and hopefully when the architecture successfully has learned the prediction task the rest of the objectives in chapter 3 can be tested.

[3] Several different fitness functions, parameter settings and selection methods will be tested to find a suitable evolutionary algorithm.

5 Experimental Results

This chapter presents the results of all experiments performed in this thesis. Each of the following sections corresponds to one of the objectives mentioned in chapter 3. The results will be discussed in chapter 6.

5.1 Results Objective 1

The aim of the first objective was to partially replicate Nolfi and Tani (1999) using a combination of their hierarchical architecture and Linåker and Niklasson's (2000a) ARAVQ. The main concern is to see if this novel architecture can develop the same kind of complex representations as those mentioned by Nolfi and Tani (1999), i.e. that their robot was able to keep track of its own relative position to the environment. And perhaps most fundamentally, these representations made it possible for the architecture to anticipate the next upcoming event.

At first, the ARAVQ was trained for 3000 time steps using the parameters: threshold $\delta = 0.6$, stability criteria $\varepsilon = 0.3$, learning rate $\alpha = 0.03$ and $n = 3$. To have the ARAVQ better acknowledge the behavior of the robot (turning behavior in particular) the magnitude of the motor input where doubled. With this configuration the ARAVQ found five concepts (Figure 20), corresponding to; following wall to the right (a), turning left (d), moving straight-ahead through corridor (e), turning right (b) and turning right in a corridor (c).

	sensor (1-8) and motor activation (LM, RM)									
Vector	1	2	3	4	5	6	7	8	LM	RM
a	0.0022	0.0034	0.0061	0.0031	0.3176	0.9990	0.0044	0.0075	1.8414	1.8323
b	0.0145	0.0315	0.0066	0.0055	0.0482	0.9775	0.0065	0.0346	1.9800	1.4003
c	0.9958	0.3456	0.0044	0.0050	0.0062	0.8940	0.0059	0.0041	1.9811	1.4301
d	0.0079	0.0046	0.0068	0.0445	0.8843	0.9995	0.9471	0.6969	0.5577	1.9970
e	0.8934	0.0090	0.0049	0.0066	0.2670	0.9994	0.0078	0.0005	1.8685	1.8647

Figure 20: The five model vectors allocated by the ARAVQ. Each vector consists of 10 values; 1-8 corresponds to the 8 short range sensors found on the Khepera robot (cf. Figure 16) and LM and RM corresponds to the left and right motor respectively.

An illustration of how these five different concepts are used in the environment can be seen in Figure 21. During one typical lap in the environment the ARAVQ changes concept 20 times, creating a quite complex sequence. At first, when the robots sensor input contained a few percentages of noise, it sometimes happened that the ARAVQ was too sensitive and changed active concept even though there were no ‘obvious’ reason. This typically happened when the robot got very close to a wall and the wall-following behavior got mixed up with the turn right behavior. Since this would make the prediction task for the second level almost impossible the noise in the input had to be reduced to 0 and the sensitivity parameter n in the ARAVQ increased to 4. These two small adjustments did take care of the problem. Worth mentioning is that Nolfi and Tani (1999) only found three concepts (using 3 hidden nodes). None of these concepts included a right turning behavior. As a consequence, their solution would not be sensitive enough to support navigation.

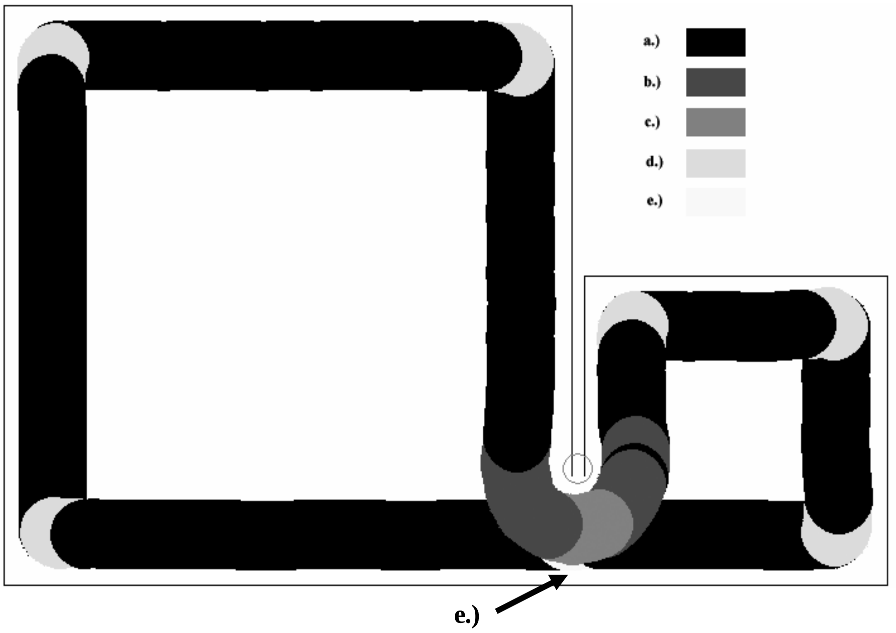


Figure 21: The distribution of the five concepts found by the ARAVQ. (a) following wall to the right, (d) turning left, (e) moving straight-ahead through corridor, (b) turning right and (c) turning right in a corridor.

However, the downside of using many concepts is, as mentioned earlier, that they create a rather complex sequence for the second level to learn. That was indeed what was found in the second training phase. The second level turned out to be impossible to train using five concepts. Both back propagation and different forms of EA were used, but none where sufficiently successful. The worst results were obtained using backpropagation and the best using an EA. The reason for this failure is probably that out of the 20 changes that take place during one lap two of the concepts (*c* and *e*) only occurred once. Quite naturally, the architecture ignores less common concepts in favor of more frequent concepts. When using an EA about 91% of all concept where classified correctly. These 91% closely corresponded to the 18 concept which occurred more frequently. To try to get around these problems, several different techniques where used. First, weighted fitness was introduced. The fitness given to an individual for correctly predicting a concept was based on the frequency of that concept. In that way, it would be equally good to predict the concept *d* one time as the concept *a* nine times. Unfortunately, this did not lead to any improvement. Second, the EA was modified so that the robot was only allowed to continue its way trough the environment, collecting fitness, for as long as it predicted correctly. This turned out to be reasonably successful. The robot was able to finish roughly four laps in the environment, when time prediction where ignored. However, when time prediction was introduced learning was much poorer. The reason for this might be that evolution focuses too much on these new parameters. If for instance a correct prediction of a concept gives 5 fitness points and the fitness for time prediction ranges from 0-5 and the individual currently is able to correctly predict 48 concepts in a row. Then it is much more privileged to optimize the previous 48 time predictions than to learn to predict another concept and only gain 5 more fitness points. It is also very difficult for evolution to insert successful mutations during such a long sequence of predictions. Even the slightest change of weights in the network architecture might have large consequences. Instead of predicting one additional concept correctly, the whole previous sequence might be spoiled.

Since none of the attempts to train the second level using five concepts were good enough, the only real solution to the problem was to reduce the number of concepts. This was done by re-training the ARAVQ for 3000 time steps with slightly stricter parameters: threshold $\delta = 0.7$, stability criteria $\varepsilon = 0.2$, learning rate $\alpha = 0.03$ and $n = 3$. This time the magnitude of the motor activation was not doubled. With these settings the ARAVQ found the exact same concepts as Nolfi and Tani (1999), following wall to the right (a), turning left (b) and moving straight-ahead through corridor (c).

	sensor (1-8) and motor activation (LM, RM)									
Vector	1	2	3	4	5	6	7	8	LM	RM
a	0.0041	0.0109	0.0085	0.0035	0.2703	0.9893	0.0054	0.0067	0.9404	0.8744
b	0.0079	0.0085	0.0023	0.0265	0.9961	0.9704	0.9360	0.5497	0.2860	0.9984
c	0.9512	0.5172	0.0072	0.0049	0.0396	0.9282	0.0073	0.0058	0.9858	0.7226

Figure 22: The three model vectors allocated by the ARAVQ. Each vector consists of 10 values; 1-8 corresponds to the 8 short range sensors found on the Khepera robot (cf. Figure 15) and LM and RM corresponds to the left and right motor respectively.

The distribution of concepts during one lap in the environment can be seen in Figure 23. This time the least common concept occurs twice in 16 changes. Hopefully this, in addition to the fact that there are only three concepts to choose from, will make it much easier for the second level to predict the next upcoming concept.

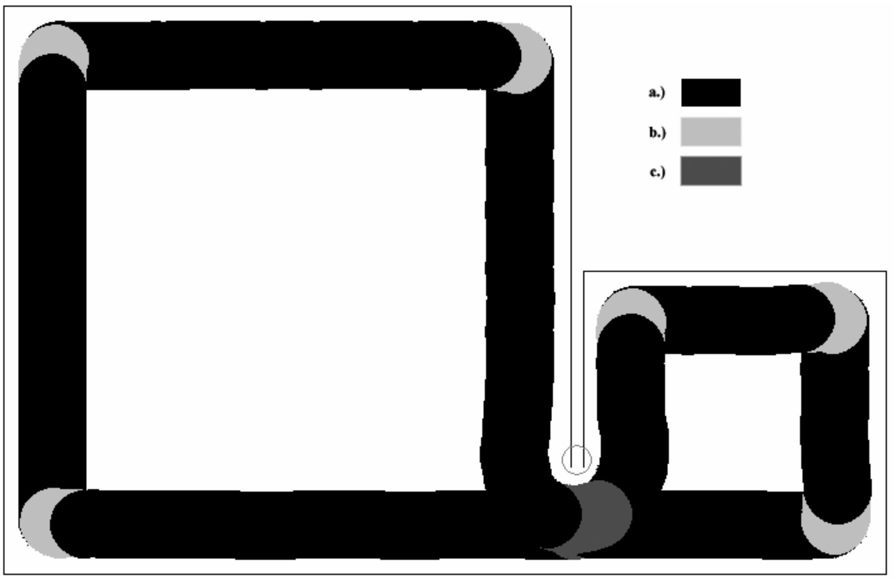


Figure 23: The distribution of the three concepts found by the ARAVQ. (a) following wall to the right, (b) turning left and (c) moving straight-ahead through corridor.

The second level was again trained using both backpropagation and several different EAs. Somewhat surprisingly, it turned out to be very difficult to train the second level with backpropagation, although the exact same parameters were used as those mentioned in Nolfi and Tani (1999) (*learning rate*, *momentum* etc.). Several attempts where performed using different number of hidden nodes, ranging from 3 to 40 (Nolfi and Tani (1999) used 30) but none were completely successful. In all cases the architecture failed in predicting at least one concept. Better results were obtained using an EA. A population of 100 individuals was trained for 5000 generations using the fitness function shown in equation 8, were *f(t)* is the fitness received at time step *t*, T_{pred} the time predicted by the architecture and T_{target} the correct value. The parameter *n* corresponds to the number of inputs to the second level and C_{pred} and C_{target} corresponds to the output for a possible active concept and the desired output for that possible active concept, respectively. If the candidate output is indeed the desired output, the target is 1, otherwise 0. This function is intended to be some kind of evolutionary guidance. *End epoch* symbolizes the fact that the individual's epoch is ended if it predicts wrong. The parents for the next generation of individuals were selected using a combination of elitism and tournament.

Equation 8: fitness *f(t)* received at each time step *t*

$$f(t)=\begin{cases} 5+\left(\dfrac{5}{1+\left|t_{pred}-t_{target}\right|}\right) & \text{correct prediction} \\ 5-\left(\dfrac{1}{\mathrm{n}}\displaystyle\sum_{i=0}^{n-1}\left|c_{pred}(i)-c_{target}(i)\right|\right)\ ;\ \textit{end epoch} & \text{otherwise} \end{cases}$$

The fitness function is only used in the case of a concept change in the ARAVQ. Thus, one time step *t* here is not to be confused with one time step 100*ms* for the robot moving around in the environment. In sum, an individual can receive $5+5$ fitness points at each successful time step t. Each individual where given a maximum of 3000 time steps (100*ms*) to navigate the environment. This was enough to finish roughly 7 laps, each one containing 16 concepts switches, giving a total maximum fitness around 1100. Initially a gaussrand mutate variance of 0.5 was used. This mutation rate where however reduced each 200 generation down to a lowest value of 0.05. The training result (Figure 24) shows that the architecture develops slowly until 900 generations. At that point the network passes some kind of limit and is capable to run for all 3000 time steps, enabling a maximum of 116 concept predictions.

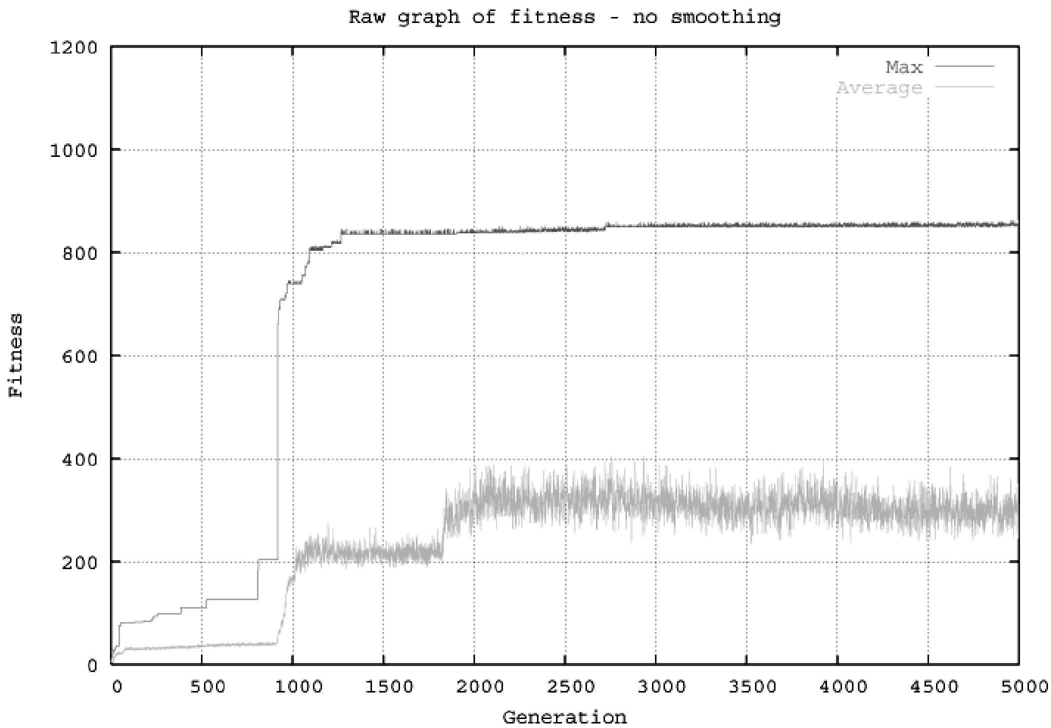


Figure 24: Fitness received by the best (Max) and average (Average) individual at each generation.

The maximum fitness reached during 5000 generations (roughly 840) distributed on the 116 changes gives an average of approximately 7.1, which should be compared to the maximum 10. This indicates that the time prediction is not optimal. A table over the predicted concepts and their time values and their real counterpart (Figure 25) shows that the second level are pretty good at predicting the time of corners (b) and corridors (c), but worse at predicting the time of wall following behavior (a).

time *t*	real	predicted	*cont.*		
1	b 0.6	b 0.5965	26	a 2.6	a 2.5095
2	a 6.1	a 3.9221	27	b 0.6	b 0.6000
3	b 0.5	b 0.6045	28	a 2.7	a 0.4559
4	a 6.1	a 1.5040	29	c 0.6	c 0.0533
5	c 0.2	c 0.2055	30	a 6.8	a 8.8648
6	a 3.5	a 8.4159	31	b 0.6	b 0.5986
7	b 0.6	b 0.5968	32	a 6.1	a 6.1049
8	a 2.5	a 5.5843	33	b 0.5	b 0.6014
9	b 0.6	b 0.5970	34	a 6.0	a 3.6157
10	a 2.5	a 2.5737	35	b 0.6	b 0.6050
11	b 0.6	b 0.5996	36	a 6.3	a 1.2813
12	a 2.9	a 0.4739	37	c 0.2	c 0.2091
13	c 0.8	c 0.0563	38	a 3.8	a 8.3935
14	a 6.7	a 8.8655	39	b 0.6	b 0.5972
15	b 0.6	b 0.5986	40	a 2.5	a 5.5851
16	a 6.0	a 6.0963	41	b 0.5	b 0.5970
17	b 0.6	b 0.6013	42	a 2.6	a 2.5402
18	a 6.1	a 3.6309	43	b 0.6	b 0.6000
19	b 0.5	b 0.6050	44	a 2.6	a 0.4688
20	a 6.5	a 1.2756	45	c 0.9	c 0.0524
21	c 0.1	c 0.2221	46	a 6.6	a 8.8817
22	a 3.4	a 8.3370	47	b 0.6	b 0.5986
23	b 0.6	b 0.5967	48	a 6.1	a 6.1010
24	a 2.5	a 5.5210	49	b 0.6	b 0.6014
25	b 0.6	b 0.5971	50	a 5.8	a 3.6507

Figure 25: Table over predicted concepts and activation time (seconds) and their corresponding targets.

A plot of both the real and the predicted values (Figure 26) shows that there are a stable difference in the predictions of time for wall following behavior in a small room and a big room. It also seems to be the case that the length of the time predictions for wall following is longer directly after a corridor (c) and then shrinks and becomes shorter and shorter.

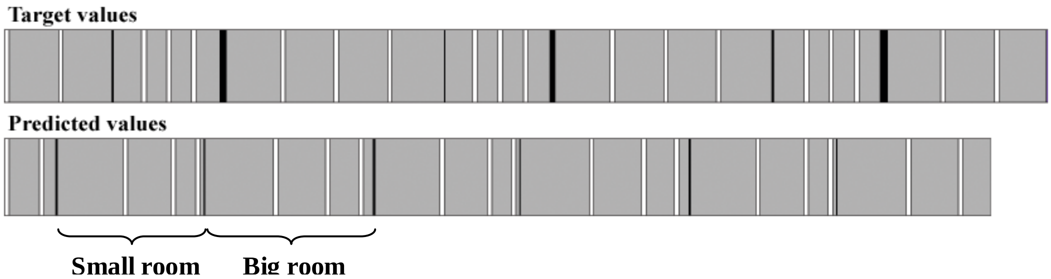


Figure 26: The real and predicted concept and time plotted over a timeline. Gray corresponds to wall following (a), white to corners (b) and black to moving through a corridor (c). The plot shows that the time predictions for corners and corridors are relatively good while wall following are much poorer.

Nolfi and Tani (1999) used a number of slightly different environments to show that their architecture must have captured some topological features from the environment in its internal representations. A similar study was performed here, using the environments B-D in Figure 16. The best individual was tested for 10 000 concept changes in each environment. The result (Figure 27) shows that the architecture predicted almost perfect (error <1%) both in environment *A* and *D*. Both the concept prediction and time prediction in *A* and *D* were slightly better than in *B*. In the *B* environment a couple of additional concept changes occurred (as a consequence of the extra obstacle) which increased the error for concept prediction to around 15%. The worst results where obtained in the *C* environment. The lack of corridor concepts made the architecture very confused. After a short while in the environment the second level started to predict all the upcoming concepts as corridors. Thus, the results were as bad as >99% (99.64) incorrect.

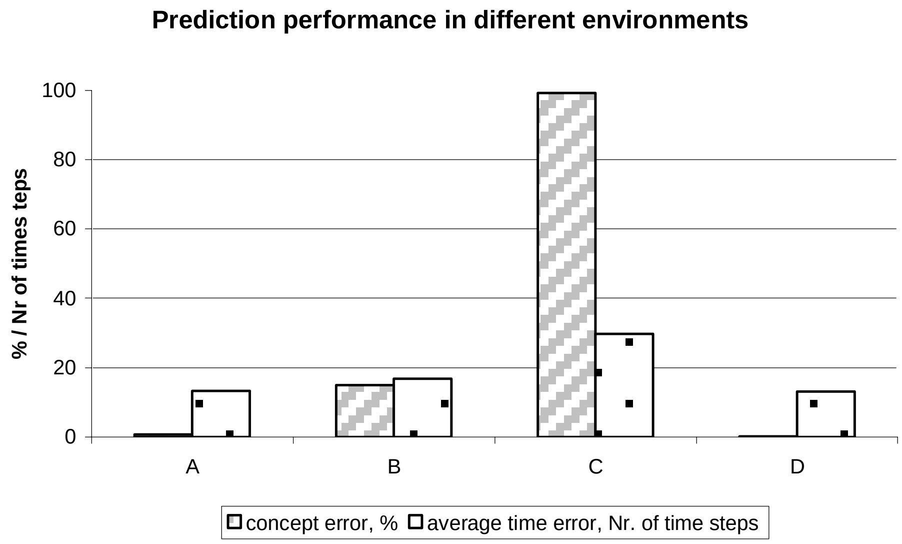


Figure 27: Prediction performance in different environments, (A) is the standard environment and (B-D) the experimental environments found in Figure 16. The first bar shows the percentage error for concept prediction and the second the average number of time steps erroneously predicted by the ANN.

Although these results do not completely correspond to Nolfi and Tani's (1999) finding, it is still possible to draw the same conclusion: that the second level of the architecture must have extracted some topological regularity from the environment.

Nolfi and Tani (1999) did one more experiment to examine the capabilities of the internal representations developed by the second level. Within some intervals (1000 time steps) they changed the location of the robot to see what happened with the second level prediction. They found that the prediction error drastically increased for a number of time steps but soon dropped as the robot was able to re-locate its own relative position to the environment. The same experiment was performed here. Each 25th change of concept (approximately 1.5 laps) the robot were moved for 190 time steps (100ms) in the environment. This corresponds to almost a half lap, placing the robot somewhere in the other end of the environment. The result, illustrated in Figure 28, shows that the architecture starts to predict wrong for a short while after it has been re-located, but soon finds its own relative position to the environment and starts to predict correct again. Hence, the architecture used here seems to have developed the same kind of complex representations as those found in Nolfi and Tani (1999).

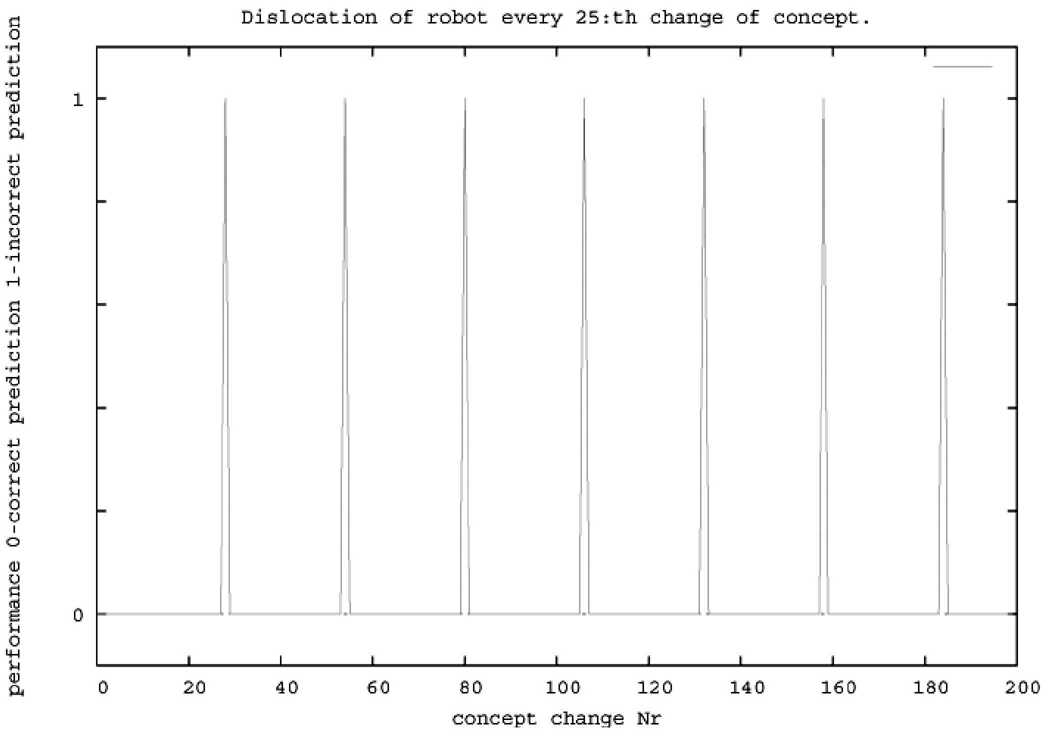


Figure 28: Prediction results when the robot is dislocated every 25th change of concept. At first the network starts to make erroneous predictions, but soon it is able to locate its own position in the environment and predict correct again.

5.2 Results Objective 2

The second objective aimed at de-coupling the second level in the architecture from the world and let it run on its own predictions. In that way, it might be possible to actually – in a primitive way – simulate perception. At each time step *t + 1* the predictions from the previous time step *t* are given as input to the second level. The black circle in Figure 30 shows the robots starting location and the gray circle shows the position where the second level starts to simulate. For the first ten concepts changes the recurrence in the ANN are able to build up some history and locate itself in the environment. The results, from the simulation which lasted fore 40 concept changes, are presented in Figure 30 and plotted in Figure 31. The table shows that the second level is capable to simulate long series of concepts perfectly. Not surprisingly the simulations of time are not as accurate. But as in the case of time prediction, the ANN seems to have captured the time of the turning left concept. Similarly, it also here seems to have captured the general idea that there are a difference in length of a wall following, a turning left and a corridor concept.

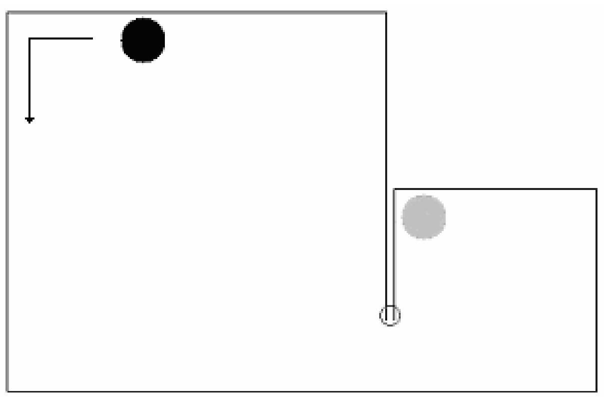


Figure 29: The robot starts to move around in the environment at the black spot and begins to simulate perception at the gray spot.

Figure 31 shows how the simulated time for wall following behavior slowly decreases each time step until the point when the network simulates a corridor. It is reasonable to believe that this works as some sort of calculation mechanism which allows the network to keep track of the number of passed concepts and when to simulate a corridor.

time *t*	**real**	**simulated**	***cont.***		
1	b 0.6		**26**	a 2.6	a 3.8592
2	a 6.1		**27**	b 0.6	b 0.6020
3	b 0.5		**28**	a 2.7	a 1.2293
4	a 6.1		**29**	c 0.6	c 0.0285
5	c 0.2		**30**	a 6.8	a 8.9167
6	a 3.5		**31**	b 0.6	b 0.5991
7	b 0.6		**32**	a 6.1	a 6.2807
8	a 2.5		**33**	b 0.5	b 0.6012
9	b 0.6		**34**	a 6.0	a 3.8592
10	a 2.5		**35**	b 0.6	b 0.6020
11	b 0.6	b 0.6020	**36**	a 6.3	a 1.2293
12	a 2.9	a 1.2294	**37**	c 0.2	c 0.0285
13	c 0.8	c 0.0285	**38**	a 3.8	a 8.9167
14	a 6.7	a 8.9167	**39**	b 0.6	b 0.5991
15	b 0.6	b 0.5991	**40**	a 2.5	a 6.2807
16	a 6.0	a 6.2807	**41**	b 0.5	b 0.6012
17	b 0.6	b 0.6012	**42**	a 2.6	a 3.8592
18	a 6.1	a 3.8592	**43**	b 0.6	b 0.6020
19	b 0.5	b 0.6020	**44**	a 2.6	a 1.2293
20	a 6.5	a 1.2293	**45**	c 0.9	c 0.0285
21	c 0.1	c 0.0285	**46**	a 6.6	a 8.9167
22	a 3.4	a 8.9167	**47**	b 0.6	b 0.5991
23	b 0.6	b 0.5991	**48**	a 6.1	a 6.2807
24	a 2.5	a 6.2807	**49**	b 0.6	b 0.6012
25	b 0.6	b 0.6012	**50**	a 5.8	a 3.8592

Figure 30: Simulated chain of concepts and activation times compared to the real targets.

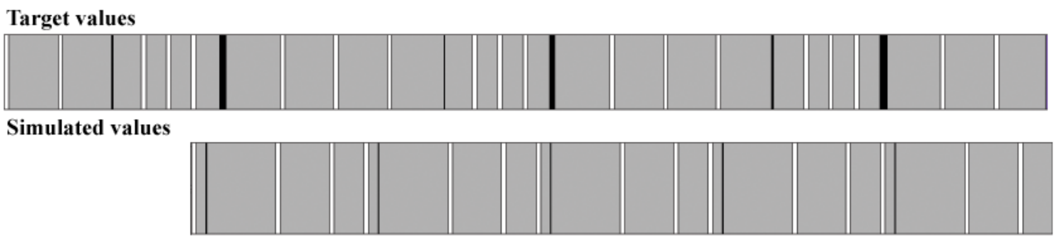


Figure 31 The real and simulated concept and time plotted over a timeline. Gray corresponds to wall following (a), white to corners (b) and black to moving through a corridor (c). Again the plot shows that the time predictions for corners and corridors are relatively good while wall following are poorer. It is also possible to see how the simulated time for wall following slowly decreases for each step until a certain point when the network simulates a corridor.

Further experiments showed that it was possible to simulate even longer sequences than 40 changes. Simulations featuring around 100 changes without any problem indicate that it would be possible to run unlimited simulations.

5.3 Results Objective 3

The third objective aimed at inverting the stored model vectors found by the ARAVQ and in such way get a further idea of the representations developed by the architecture. The first three inversions are based on the case when the ARAVQ found three concepts. In the first case the real target values are used to create the inversion. The idea is that this will work as a kind of reference for the other inversions. In the second case the simulated concepts and times are used to create the inversion and in the third case the simulated values are used to create the inversion. Finally, a comparison is done with the inversion created from the five concepts first found by the ARAVQ.

In each case a number of concepts and times corresponding to one complete lap are selected. These concepts and times are ordered into a sequence of model vector winners (Figure 32). Then a 'dummy' robot is put in motion in an empty world in the way indicated by the sequence and the stored model vectors.

context	**sequences of model vector winners**
real	$a_{61}b_5a_{61}c_2a_{35}b_6a_{25}b_6a_{25}b_6a_{29}c_8a_{67}b_6a_{60}b_6$
predicted	$a_{39}b_6a_{15}c_2a_{84}b_6a_{56}b_6a_{26}b_6a_5c_1a_{88}b_6a_{60}b_6$
simulated	$a_{38}b_6a_{12}c_1a_{89}b_6a_{63}b_6a_{38}b_6a_{12}c_1a_{89}b_6a_{63}b_6$

Figure 32: Sequences of model vector winners created from real, predicted and simulated concepts and activation times

At each time step all sensor values with activation higher than 0.3, in the model vectors, are plotted at their approximate distance from the robot (Figure 33). The value 0.3 has simply been chosen by trial and error.

	sensor (1-8) and motor activation (LM, RM)									
Vector	1	2	3	4	5	6	7	8	LM	RM
a	0.0041	0.0109	0.0085	0.0035	0.2703	0.9893	0.0054	0.0067	0.9404	0.8744
b	0.0079	0.0085	0.0023	0.0265	0.9961	0.9704	0.9360	0.5497	0.2860	0.9984
c	0.9512	0.5172	0.0072	0.0049	0.0396	0.9282	0.0073	0.0058	0.9858	0.7226

Figure 33: All sensor characteristics above 0.3 in the stored model vectors are plotted during the inversion.

Figure 34 shows the fictive starting position for all inversions.

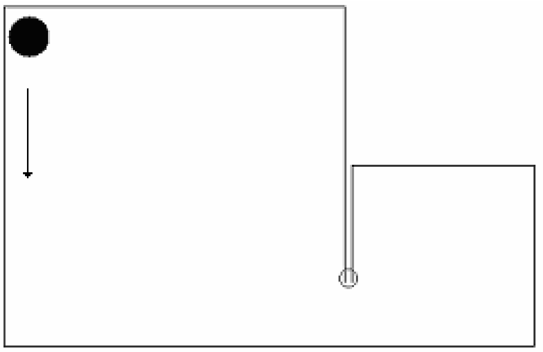


Figure 34: the fictive starting position for all simulations.

The resulting inversions (Figure 35), in all cases (a-c) show that the stored model vectors in combination with the time measurement only in a limited way correspond to the real world. The *a* concept unfortunately features a slight right turning behavior. The reason for this is probably that the *a* concept, when created by the ARAVQ, also partially encoded the long right turn (e.g. Figure 22).

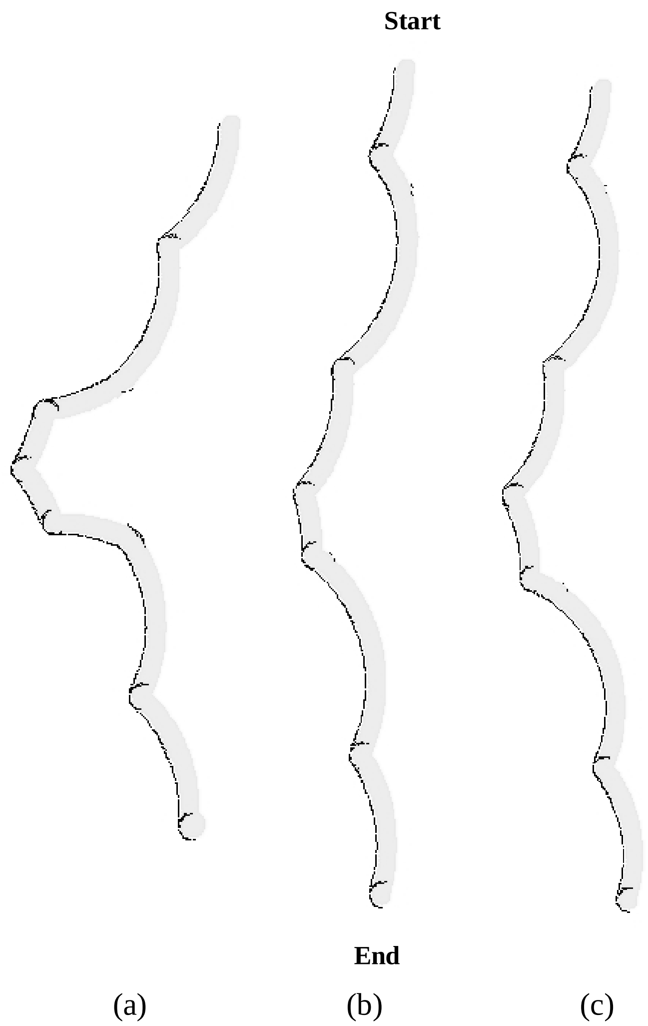


Figure 35: The inversions created from the sequences of model vector winners. (a) Real values, (b) predicted values and (c) simulated values.

This, in combination with the fact that there are no concepts (only) encoding the long right turn limits the accuracy of the inversions. It is however; still possible to see how both the predictions (b) and the simulations (c) have captured most of the important topological characteristics of the environment. In both cases all corners and passes trough the corridors are captured. The number of time steps for corners, and to some degree corridors, is also well captured. Similarly to earlier results, time for wall following are problematic. In both the case of predictions and simulations, the architecture has only captured that wall following concepts last longer than turning left concept and passing trough corridor concept. Unfortunately, there seems to be a general problem also in the inversion from the real values to capture the right amount of turning needed to complete a 90-degree corner. This might be explained by the *n* value in the ARAVQ settings. A large *n* value will have the finite moving average change slower, which in turn will make to switches between concepts rather inexact.

The inversions created from three concepts could to some degree be compared to an inversion created from the five concept originally extracted by the ARAVQ. The same starting position was used for the following sequence of model vector winners created from target values:

$a_{60}d_{6}a_{60}e_{4}a_{34}d_{6}a_{25}d_{6}a_{25}d_{6}a_{19}b_{10}c_{8}b_{15}a_{51}d_{7}a_{59}d_{7}$

Again all sensor values with activation higher than 0.3 in the model vectors were plotted (Figure 36).

sensor (1-8) and motor activation (LM, RM)										
Vector	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**LM**	**RM**
a	0.0022	0.0034	0.0061	0.0031	0.3176	0.9990	0.0044	0.0075	1.8414	1.8323
b	0.0145	0.0315	0.0066	0.0055	0.0482	0.9775	0.0065	0.0346	1.9800	1.4003
c	0.9958	0.3456	0.0044	0.0050	0.0062	0.8940	0.0059	0.0041	1.9811	1.4301
d	0.0079	0.0046	0.0068	0.0445	0.8843	0.9995	0.9471	0.6969	0.5577	1.9970
e	0.8934	0.0090	0.0049	0.0066	0.2670	0.9994	0.0078	0.0005	1.8685	1.8647

Figure 36: All sensor characteristics above 0.3 in the stored model vectors are plotted during the inversion.

The results illustrated in Figure 37 shows how much more the five concepts are able to encode in the environment. Again the number of time steps for the different model vectors are somewhat distorted. In most cases the robot stops turning after about 70 or 80 degrees. But, the inversion still shows to what a striking level the ARAVQ are capable of capturing the dynamics of the environment.

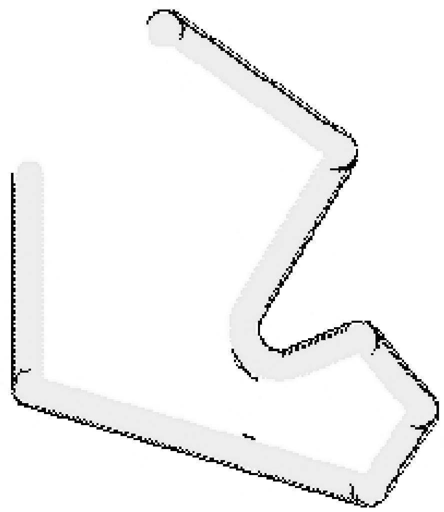


Figure 37: The five concept inversion, created from the sequences of model vector winners.

5.4 Results Objective 4

The final objective, were intended to examine if the robot were able to navigate blindly, only relying in its own simulations. But, as the previous results have shown that is not possible for a number of reasons. First, and most important, the three concepts found by the ARAVQ do not capture the world well enough. There are no concept only encoding the long right turn, and the right turn passing trough the corridor. Moreover, the wall following concept are not moving straight forward, since it partially, but not completely, encodes the right turn. Second, the accuracy of the time prediction and simulation are not good enough. Navigating blindly is a very delicate task which puts very high demands on the sensitivity of internal dynamics and overt behavior. To succeed with blind navigation a much simpler environment would be needed or the training methods need to be improved.

6 Discussion and Conclusion

A number of objectives were set up to answer the hypothesis that an architecture which made use of abstracted concepts would be able to, in a primitive way, simulate perception. The results from the first objective showed that the architecture used in this thesis is comparable to the one used in Nolfi and Tani (1999), both when it comes to predictions and internal dynamics. The architecture did however experience some problems when it came to time predictions of long wall-following behavior. The reason for this could probably be found in the training algorithms. Training ANNs with EAs is not an exact science, and it is fully possible that the prediction problems would be possible to overcome with a slightly different training parameters and fitness function. The results in Nolfi and Tani (1999) were presented with mean squared error (MSE) as measurement which might be problematic when it comes to comparing two results. It is fully possible to acknowledge the changes or development in such results, but since they do not mention what their network typically outputs or how they measure time. It makes a very big difference in MSE between a network that encodes each time step as 1 and one that encodes it as 0.01. It should however be enough to say that both architectures are capable to predict very well.

The results of the second objective show that it indeed was possible to simulate long sequences of concepts and (at least to some degree) time. Simulated sequences as long as 100 concepts in a row were tested without any problem, indicating that the internal dynamics of the network must have captured the topology of the environment very well. It is a guess that the second level of the architecture has memorized the sequence of concepts found in the environment and thus did not have to care too much about the time input. That would explain the very small difference between the predicted and the simulated output.

The results from the third objective did reveal some of the internal dynamics which were the basics for the successful simulations. Although the three concepts found by the ARAVQ were far from ideal, they were good enough to capture the most fundamental part of the environments. The inversions illustrated how well the second level of the architecture had learned to predict and simulate upcoming corners and corridors. Unfortunately, it did also indicate that it would be impossible to navigate blindfolded with these models. The imperfect time predictions combined with the limited concepts developed by the ARAVQ were simply not sensitive enough to be translated into sensory-motor input. The major problems with the concepts were simply that no one encoded the long right turn and that the straight-ahead wall-following behavior featured a slight right-turning movement.

Since, the results of objective three indicated that the accuracy of the architectures internal dynamic were not good enough, no blind navigation was performed in objective four. Although this would have been a very nice benchmark for how suited abstractions are for simulations, one should remember that it is a slightly unrealistic task. Even if an animal or a human would be blindfolded and unable to see, there would still be a number of operating senses providing useful information. Hearing might give some directions towards a familiar sound, touch and propreception might reveal different textures around you or beneath your feet etc. A more realistic task would be if the robot was equipped with both a camera and short-range sensors and just the camera were blindfolded. Then the robot would still have one extra sense providing a limited amount of information. This would however, not change anything here since the problem lies within the limited concepts. In order for blind navigation to work, all parts of the topology in the environment need to be captured (like in the first training session when the ARAVQ found five concepts) and the time predictions must become much more precise.

With these results at hand, it is possible to argue that the hypothesis in this dissertation has been partially confirmed. It was possible to internally simulate perception at an abstract level using a combination of Linåker and Niklasson's (2000a) ARAVQ and Nolfi and Tani's (1999) hierarchy of prediction networks. These internal simulations were however, not accurate enough to be translated back into raw sensor and motor activations again. Put in relation to Hesslow's (2002) simulation hypothesis, it is only possible to say that the robot has developed some kind of simple associations or anticipation mechanism which enables it to, based on history and the present situation, anticipate the next upcoming situation or series of situations (Figure 38). It is possible for the robot to in its '*mind*' find the abstract way from A to B in the environment. This can be compared with me trying to find my way from my workstation to the coffee machine downstairs. I know roughly the series of actions and situations which will take me there, first I take a right out through the door, then follow the corridor, go down the stairs, take a left, pass through the door, take a right and I am there, but I do not know the exact path. The robot is not very good at simulating the active time of each concept or situation, but neither am I. I know that a move straight ahead situation usually last longer than a turn and that a turn usually are followed by a new move straight ahead situation, but so does the robot.

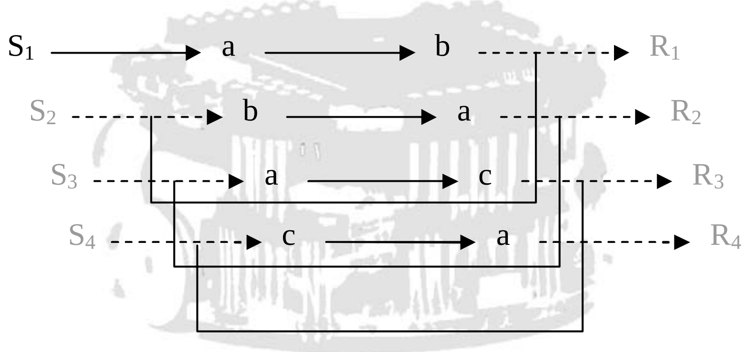


Figure 38: Based on history and the current situation, is the robot capable of, anticipate or internally simulate a long series of upcoming situations.

This also raises some questions about the role of abstractions in Hesslow's (2002) simulation theory. As mentioned by Svensson (2002) is it still an open question at which level these cognitive simulations occur. The results in Jirenhed (2001) indicated that it could be problematic to simulate perception at a raw sensory level, leading this dissertation to try to simulate perception at a more abstract level. The results showed that that was indeed possible, but that the simulated concepts were too inaccurate to be translated back to sensor and motor information. This unfortunately, limits the importance of the simulations, but still demonstrates that abstractions can be very useful, in simulations or emulations, when it comes to refining and handling large amounts of information.

The results in this thesis should also be put in relation to the suggested constructivist bottom-up approach presented in Chapter 2. First and foremost, the robot used in this thesis is both embodied and situated, which allows it to interact with the environment through a continuous sensory motor flow. As suggested by the constructivist approach, both the abstracted concepts and the anticipation mechanism developed in the robot have been actively built up by the robot itself, through such interaction with the environment. These internal mechanisms are also created only with respect to the agent and are therefore highly subjective. Second, the internal simulations of the second level in the architecture can not be said to be any form of corresponding representations, because they, as expressed by Clark and Grush (1999), do actually precede the current state of affairs. They do however, fit the description of interactivist representations given by Dorffner (1997); "internal structures on which an agent operates in order to guide its behavior" (cf. Chapter 2). That is, the robot's internal simulations can be seen as a form of functional representations. The model vectors extracted by the ARAVQ can, on the other hand, be said to be a clear case of corresponding representations. This does not necessarily constitute a problem since there is nothing in a constructivist approach which is directly conflicting with the possibilities of internal structures corresponding to external objects or concepts. In this case there is not the common problem of an external designer deciding the exact correspondence between representation and situation, but a self-organized reference. In sum, it is possible to argue that the robot which is developed in this thesis fits within the framework of a constructivist bottom-up approach (as described here).

Finally there are a number of other interesting issues regarding the results of the experiments in this thesis. First of all, that it turned out to be so hard to train the second layer of the architecture. When the ANN was trained on five concepts it was almost impossible to develop a solution that acknowledged the two least commonly used concepts. If it however, would be possible to develop a better training algorithm which made it possible for the ANN to learn all concepts, it would probably result in even more interesting overall results. The second layer was not unproblematic to train using three concepts either. It is quite surprising that it turned out to be impossible to replicate Nolfi and Tani's (1999) study when using the very same parameters. There might however, be several reasons for this. It might be possible differences in implementation or differences in encoding of concepts and time. Another interesting question is if it would be possible to achieve the same kind of internal simulations of perception using Nolfi and Tani's (1999) architecture. Based on the results from objective one, it is reasonable to assume that the second layer of both architectures work in a similar way. Hence, one can draw the conclusion that the same results would be possible using Nolfi and Tani's (1999) architecture. Finally, the experiments in objective one showed that the Linåker and Niklasson's (2000a) ARAVQ was excellent at extracting appropriate concepts from the environment. It was easy to adjust in sensitivity and found the concepts in a very short time. Less pleasing was that the ARAVQ, in the case of five allocated concepts, sometimes was too sensitive and changed the active concept even though there were no 'obvious' reason. This typically happened when the robot got very close to a wall and the wall-following behavior got mixed up with the turn right behavior. This could however be handled by reducing all noise to the sensors and increase the value of the n parameter.

6.1 Future work

In a possible future set of experiments in the same area, it would be a good idea to try to improve the training algorithm for the second level of the architecture, so that all parts of the environment can be encoded. Alternatively a simpler set of environments should be used. This in combination with the (hopefully) more accurate time prediction might make it possible to use the internal simulations for navigating 'blindfolded'. Additional things that would be interesting to investigate are if it is possible to further extend the architecture such that there exist a more dynamical influence between the predicted or simulated concept and the behavior of the robot. As mentioned earlier, ‘blind’ navigation is a quite unrealistic task for a cognitive creature. It would be interesting to develop a robot that in a test condition got some limited information from the environment and could use that in combination with the internal simulations to guide its behavior.

References

Beer, R. D. (1990) *Intelligence as Adaptive Behavior, an Experiment in Computational Neuroethology*. San Diego: Academic press, Inc.

Beer R. (1995) A Dynamical Systems Perspective on Agent-Environment Interaction. *Artificial Intelligence, 72*, 173-215.

Bickhard, M. H. & Terveen, L. (1995) *Foundational Issues in Artificial Intelligence and Cognitive science – Impasse and Solution*. New York, NY: Elsevier.

Brooks, R. A. (1991a) Intelligence without representations I: J. Haugeland (ed.), *Mind design II: philosophy, psychology, artificial intelligence* (p. 395-420). Cambridge: MIT press.

Brooks, R. A. (1991b) Intelligence without reason. I: R. Chrisley (ed.), *Artificial intelligence: critical concepts* Vol. 3. London: Routledge.

Carlsson, J. & Ziemke, T. (2001). YAKS - Yet Another Khepera Simulator. In: Ruckert, Sitte & Witkowski (eds.), *Autonomous Minirobots for Research and Entertainment - Proceedings of the Fifth international Heinz Nixdorf Symposium* (p. 235-241). Paderborn, Germany: HNI-Verlagsschriftenreihe.

Clark, A. (1997) *Being there: putting mind, body and world together again*. Cambridge, MA: MIT Press.

Clark, A. & Grush, R. (1999) Towards a cognitive robotics. *Adaptive Behavior,* 7(1), 5-16.

Craik, K. (1943) *The nature of explanation.* London: Cambridge University Press.

Decety, J., Jeannerod, M., Prablanc, C. (1989) The timing of mentally represented actions. *Behavioral Brain Research, 34*, 35-42.

Dorffner, G. (1997) Radical connectionism – a neural bottom-up approach to AI. In: G. Dorffner (ed.) *Neural networks and a new artificial intelligence*. London: International Computer Press.

Dreyfus, H. L. (1979) From Micro-Worlds to Knowledge Representation: AI at an Impasse. I: J. Haugeland (ed.), *Mind design II: philosophy, psychology, artificial intelligence* (p. 143-182). Cambridge: MIT press.

Gardner, H. (1985) *The mind's new science: A history of the cognitive revolution*. New York: Basic books

Gazzaniga, M. S., Ivry, R. B., & Mangum, G. R. (2002) *Cognitive neuroscience: The biology of the mind.* (2nd ed.). New York: W. W. Norton & Co.

Grush, R. (2002) An introduction to the main principles of emulation: motor control, imagery, and perception. Unpublished manuscript. Philosophy Department, University of California, San Diego.

Hesslow, G. (2002) Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences,* 6 (6), 242-24.

Hoffmann, H., & Möller, R. (2004). Action Selection and Mental Transformation Based on a Chain of Forward Models. In S. Schaal, A. Ijspeert, S. Vijayakumar, J. C. T. Hallam, & J. A. Meyer (Eds.), *Proceedings of the 8th International Conference on the Simulation of Adaptive Behavior* (pp. 213-222). Cambridge, MA: MIT Press

Ingvar, D. H. & Philipsson, L. (1977) Distribution of the cerebral blood flow in the dominant hemisphere during motor ideation and motor performance. *Annals of Neurology,* 2, 230-237.

Jirenhed, D.-A. (2001). *Exploring Internal Simulation of Perception in a Mobile Robot*. Master's Dissertation LIU-KOGVET-D-0047-SE. Department of Computer Science, University of Linköping, Sweden.

Jirenhed, D.-A., Hesslow, G., and Ziemke, T. (2001). Exploring Internal Simulation of Perception in Mobile Robots. In: Arras, Baerveldt, Balkenius, Burgard, Siegwart (eds.). *2001 Fourth European Workshop on Advanced Mobile Robots - Proceedings*. Lund University Cognitive Studies, vol. 86, pp. 107-113.

Lee, D. N., and Thompson, J. A. I. (1982). Vision in action: The control of locomotion. In D. Ingle, M. A. Goodale, and R. J. W. Mansfield (Eds.), *Analysis of visual behavior* (pp.411-436). MIT Press. Cambridge, MA.

Linåker, F. and Niklasson, L. (2000a). Time series segmentation using an adaptive resource allocating vector quantization network based on change detection. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000)*, IEEE Computer Society, 323-328.

Linåker, F. and Niklasson, L. (2000b). Extraction and Inversion of Abstract Sensory Flow Representations. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior: From Animals to Animats 6 (SAB2000)*, MIT Press, pp. 199-208.

Newell, A. & Simon, H. A. (1976) Computer science as empirical inquiry: symbols and search. In: J. Haugeland (ed.), *Mind design II: philosophy, psychology, artificial intelligence* (p. 81-110). Cambridge, MA: MIT Press.

Nolfi, S & Tani, J. (1999) Extracting regularities in space and time through a cascade of prediction of prediction networks: The case of a mobile robot navigating in a structured environment. *Connection Science, 11, 2*, 125-148.

Palmer, S. E. (1978) Fundamental aspects of cognitive representation. In: E. Rosch & B. B. Loyd (Ed:s.), *Cognition and Categorization* (p. 259-302). Hillsdale, NJ: Lawrence Earlbaum Ass.

Pfeifer, R. & Scheier, C. (1999) *Understanding intelligence*. Cambridge, MA: MIT Press.

Rizzolatti, G., Fadiga, L., Gallese, V. & Fogassi, L. (1996) Premotor cortex and the recognition of motor actions. *Cognitive Brain Research, 3,* 131-141.

Searle, J. R. (1980) Mind, brains and programs. In: J. Haugeland (ed.), *Mind design II: philosophy, psychology, artificial intelligence* (p. 183-204). Cambridge, MA: MIT Press.

Shanahan, M. (Draft) *The Imaginative Mind Rehearsing Trajectories Through an Abstraction of Sensorimotor Space*. Submitted for journal publication.

Shepard, R. N. & Metzler, J. (1971) Mental rotation of three-dimensional objects. *Science, 171*, 701-703.

Stewart, J. (1996) Cognition = life: Implications for higher-level cognition. *Behavioral Processes, 35,* 311-326

Svensson, H. (2002) *Embodied cognition as internal simulation of perception and action: towards a cognitive robotics.* Master's Dissertation HS-IDA-MD-02-011. Department of Computer Science (School of Humanities and Informations), University of Skövde, Sweden.

Svensson, H., Lindblom, J. & Ziemke, T. (in press) *Making Sense of Embodied Cognition: Simulation Theories of shared Neural Mechanism for Sensorimotor and Cognitive Processes.* Submitted for publication

Turing, A. (1950) Computing Machinery and Intelligence. In: John Haugeland (ed). 1997 *Mind design II: philosophy, psychology, artificial intelligence* (p. 29-56). Cambridge: MIT press.

Varela, F. J., Thompson, E. & Rosch, E. (1991) *The embodied mind: cognitive science and human experience.* Cambridge, MA: MIT Press.

Ziemke, T. (2000). *Situated Neuro-Robotics and Interactive Cognition.* Doctoral Dissertation, Department of Computer Science, University of Sheffield, UK.

Ziemke, T. (2001) The construction of 'reality' in the robot: constructivist perspectives on artificial intelligence and adaptive robotics. *Foundations of Science, 6*, 163-233.

Ziemke, T., Jirenhed, D.-A., & Hesslow, G. (2002) Blind adaptive behavior based on internal simulation of perception. Technical Report HS-IDA-TR-02-001. Department of Computer Science (School of Humanities and Informations), University of Skövde.

Ziemke, T. Jirenhed, D-A. & Hesslow, G. (subm.). Internal Simulations of Perception: A Minimal Neuro-Robotic Model. *Neurocomputing*, to appear.