School of Humanities and Informatics
Masters Dissertation 25p
D-level
2005

Unsupervised Imitation in Evolved Robots

Anders Nystrand

HS-IKI-MD-05-004

Unsupervised Imitation in Evolved Robots

Submitted by Anders Nystrand 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.

2005-08-29

I hereby certify that all material in this dissertation which is not my own work has been identified and that no work is included for which a degree has already been conferred on me.

Signature: ____

Supervisor: Andreas Hansson

Unsupervised Imitation in Evolved Robots

Anders Nystrand

Abstract

Imitation learning has been studied from a large range of disciplines, including adaptive robotics. In adaptive robotics the focus is often on how robots can learn tasks by imitating experts. In order to build robots able to imitate a number of problems must be solved, including: How does the robot know when and what to imitate? How does the robot link the recognition of observed actions to the execution of the same actions? This thesis presents an approach using unsupervised imitation where artificial evolution is used to find solutions to the problems. The approach is tested in a number of experiments where robots are being evolved to solve a number of navigation tasks of varying difficulty. Two sets of experiments are made for each task. In the first set the robots are trained without any demonstrator present. The second set is identical to the first one except for the presence of a demonstrator. The demonstrator is present in the beginning of the training and thereafter removed. The robots are not being programmed to imitate the demonstrator but are only instructed to solve the navigation tasks. By comparing the performance of the robots of the two sets the impact of the demonstrator is investigated. The results show that the robots evolved with a demonstrator need less training time than the robots evolved without any demonstrator except when the task is easy to solve in which case the demonstrator seems to have no effect on the performance of the robots. It is concluded that evolved robots are able to imitate demonstrators even if the robots are not explicitly programmed to follow the demonstrators.

Keywords: Artificial Intelligence, Evolutionary Algorithms, Artificial Neural Networks, Robotics, Imitation.

Table of contents

1 Introduction

Building machines that are able to learn how to perform certain tasks is a research area that has received much attention. See (Lungarella et al. 2003) and (Walker et al. 2003) for surveys. When building mobile robots, one common approach is to let an Artificial Neural Network (ANN) act as a controller and map sensor inputs to motor actions. ANNs are suitable as robot controllers because they are robust, noise tolerant, have graceful degradation and are able to generalize and adapt to changes (Nolfi and Floreano 2000). There are many methods available to train an ANN: back propagation, reinforcement learning and evolutionary algorithms to mention a few. Evolutionary algorithms have the advantage that they do not require much feedback, and are able to solve complex problems, but they can take considerable time when evolving robots (Nolfi and Floreano 2000). One possible way to speed up the evolutionary process when training ANNs controlling mobile robots could be to let the robots imitate an expert. There are many examples of robots that have learnt to imitate experts, see for examples Gatsoulis et al. (2002) and Billard (2002). It is therefore worth investigating if evolved ANNs can learn a task faster by imitating experts than by themselves. The expert could be a human, a human controlled robot or a robot that has already been trained for the task. The latter is investigated in this thesis.

Imitation learning is a research area that has received much attention, see for example Billard (2002), Gatsoulis et al. (2002), Saunders et al. (2004), Andry et al. (2000), and Dautenhahn and Nehaniv (2002). The area is studied by researchers from a large range of disciplines, including ethology, neuroscience, psychology, linguistics, artificial intelligence (AI) and adaptive robotics. While much research focus on understanding the cognitive and neural processes behind imitation, in robotics the focus is often on how to use imitation. Both learning to imitate (Mataric, 2000), and learning through imitation (Gatsoulis et al. 2002) are of great interest in robotics.

> "This interest arises from the fact that imitation learning is a powerful means of acquiring new skills within a social environment, found in humans and other animal species. It could lead into a more user friendly way of programming robots as a human could program a robot by just demonstrating the task." (Gatsoulis et al., 2002, p. 485)

There are many definitions of imitation, and many different types of imitation. For instance, immediate imitation is distinguished from deferred. In the former imitation takes place only a short time after a demonstration while in the latter it takes place after a significant period of time. When studying animals, true imitation is distinguished from mimicry. True imitation is the ability to replicate and, by so doing, learn skills that are not part of the animal’s prior repertoire by observation of those performed by others, while mimicry is the ability to replicate a behaviour that is usually part of the usual animal repertoire according to Billard (2002). Tomasello (1999) defines imitation as the reproduction of the behaviour or behavioural strategy of the demonstrator, for the same goal as the demonstrator. Tomasello (1999) also presents a number of ways of learning from a demonstrator which he does not consider to be imitation, including exposure and stimulus enhancement. Youngsters may learn by exposure if they are exposed to new learning experiences because they stay near conspecifics without learning anything from the behaviour of the conspecifics directly, as when a youngster follows its mother and so stumbles upon water and thereby learn the location of the water. Youngsters may learn by stimulus enhancement if they are attracted to objects with which others are interacting, and then learn things on their own about those objects.

Saunders et al. (2004) identified two social learning paradigms that are widely used in robotics research: (i) following or matched-dependent behaviour and (ii) static observational learning, where the robots learn the behaviour either through perception from a static location or while following.

In all of the previous work on imitating robots, with only one exception to the author's knowledge, the learner and demonstrator are forced to synchronize with each other by the experimenter. That means the learner always knows when to pay attention and when not to. The only exception where the learner and demonstrator are not forced to synchronize is in the paper by Andry et al. (2000) where it on page 359 is argued that:

> "If the robot could have the ability of perceiving, or predicting, when the experience ends, it would be more autonomous, starting and ending spontaneous learnings by 'mimicking' the others whatever they are humans or not. Such an ability could be a great improvement inside a group of robots."

Andry et al. (2000) carried out an experiment where a simple system learnt to imitate the way a teacher pushes three keys, without any explicit reward from the teacher. There has to the author's knowledge not been any research investigating the ability of a more complex and evolved robot to learn through unsupervised imitation of a demonstrator. In this thesis it is investigated how the presence of a demonstrator affects the learning ability in evolved robots when the robots are not explicitly instructed to follow the demonstrator.

The rest of this thesis is organised as follows: In chapter two brief descriptions of ANNs and evolutionary algorithms are given. Chapter three describes the problem under investigation as well as the aims and objectives of this thesis. Chapter four presents a definition of unsupervised imitation. Chapter five presents an experiment showing that unsupervised imitation is possible. Chapter six presents a number of experiments where it is investigated how the degree of difficulty of the task the robot is learning affects the robot's tendency to imitate. Chapter seven presents the conclusions and identifies future work.

2 Background

In this chapter brief descriptions of ANNs and evolutionary algorithms are given as well as an introduction to evolving ANNs for mobile robots. This chapter also gives brief descriptions of imitation studies and related work. Readers who are already familiar with these concepts can skip this chapter.

2.1 ANN

ANNs consist of a number of nodes connected by weights. Each node has an activation value that varies over time. The activation value is sent through all outgoing weights from the node and is received by the nodes on the other side of the weights. Each weight can affect the strength of the signal, usually by multiplying the strength by a value. The activation value for a node is a function of the sum of the strength of all incoming signals to the node. This function is usually a threshold function, or a function similar to a threshold function. Figure 2.1 shows some common variations of this function. Figure 2.2 illustrates how the sum of all incoming signals are passed on to this function.

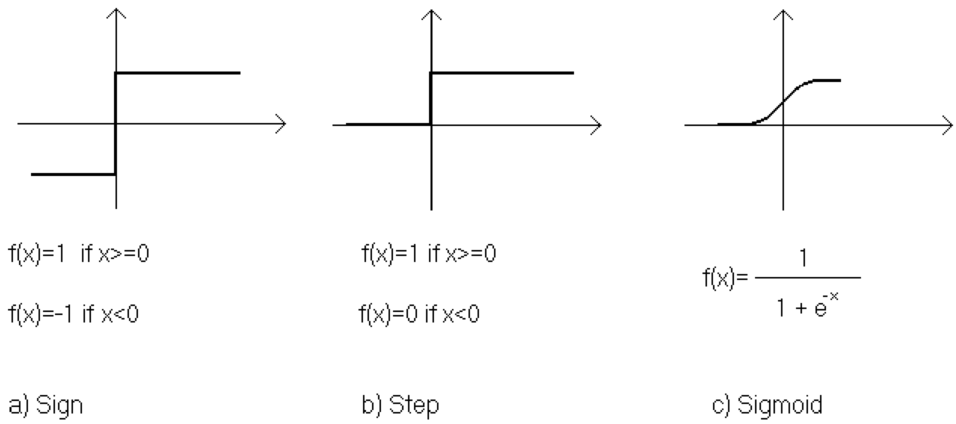

Figure 2.1 Some common functions used when calculating the activation value of a node.

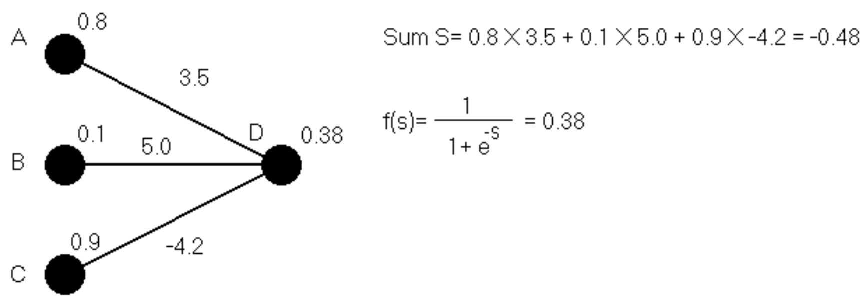

Figure 2.2 Weights are connected from node A, B and C to D. The values of these weights are 3.5, 5.0 and -4.2. The current activation values of node A, B and C are 0.8, 0.1 and 0.9 respectively. The activation value of node D is calculated by first calculating the sum of the strength of all incoming signals and then use this sum in the function f(s). The resulting activation value for D is 0.38 in this example.

Some of the nodes in the ANN act as input nodes and receive a signal from an external device such as a sensor, while some other nodes act as output nodes that send the signal to an external device such as a motor. Figure 2.3 shows some common architectures for ANNs. Nodes in feed-forward networks are organised in layers. All of the input nodes are located in the first layer which is called the input layer. The output nodes are located in the output layer which is the last layer of the network. A number of hidden layers can be located between the input and output layer. All of the nodes in a layer are connected to all of the nodes in the next consecutive layer, and can not be connected to any other nodes. This limitation implies that the activation values of the output nodes in a feed-forward network at time *t* is a function of the activations of the input nodes at time *t*. Nodes in recurrent networks on the other hand, can be connected to any of the nodes in the network. This implies that the activations of the output nodes in a recurrent network at time *t* is a function of the activations of the input nodes at time *t* and the activations of all the other nodes at time *t* - 1. Recurrent networks are therefore able to remember previous states, while feed-forward networks are reactive in the sense that they are not affected by their previous inputs.

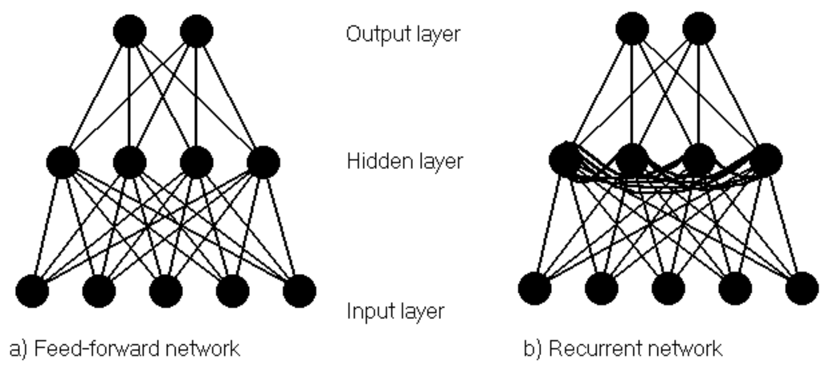

Figure 2.3. Two network architectures where the nodes are organised in layers. All nodes in one layer are connected to all the nodes in the next layer. In the recurrent network all the nodes in the hidden layer are connected to all of the nodes in the same layer. These recurrent connections give the network a memory of its previous states.

An ANN can be viewed as a machine that maps an input pattern to an output. By changing the values of the weights this mapping can be altered. There are many training methods for ANNs that finds a set of weights for the network suitable for a certain problem. See Russel and Norvig (1995) for a detailed description of ANNs.

2.2 Evolutionary algorithms

An evolutionary algorithm is a search method that can be used to find a solution for a given problem. As was mentioned in chapter 1, evolutionary algorithms have the advantage that they do not require much feedback, and are able to solve complex problems. They use a 'population' of candidate solutions to a given problem. By using a fitness function each candidate solution is evaluated and assigned a fitness value. The better the candidate solution performs when trying to solve the problem, the higher fitness it receives. Once the candidate solutions have been assigned fitness values a selection method is used to choose which solutions to keep and which to discard. There are many selection methods, but they all have in common that candidate solutions with high fitness are more likely to be kept than solutions with low fitness. The evolutionary algorithm replaces the discarded solutions with new ones created by modifying and combining the kept candidate solutions. A candidate solution is modified by a mutation operator that randomly changes some parameters, and two candidate solutions are combined into a new one by a crossover operator. Figure 2.4 describes how a crossover operator can work. This process of evaluation and reproduction is repeated for several 'generations', until time runs out or an adequate solution has been found. The designer has to find a way to represent the solutions, choose a selection method, specify how new candidate solutions will be reproduced and define a fitness function. By making a general fitness function, the designer does not have to worry about how the candidate solutions should solve the problem, but only has to specify what a solution should achieve. This way the evolutionary algorithm can come up with solutions that are not biased by the designer.

The advantages of evolutionary algorithms come with a cost. Since every candidate solution have to be evaluated for every generation the algorithm can take a very long time to finish.

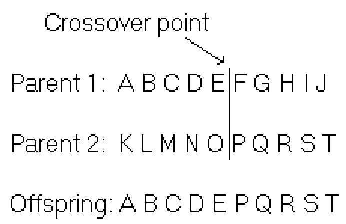

Figure 2.4 A crossover operation. The offspring is a combination of its two parents.

Evolutionary algorithms can be divided into a number of categories. The categorisation in figure 2.5 is taken from Streichert (2002) and divides evolutionary algorithms into five sub categories.

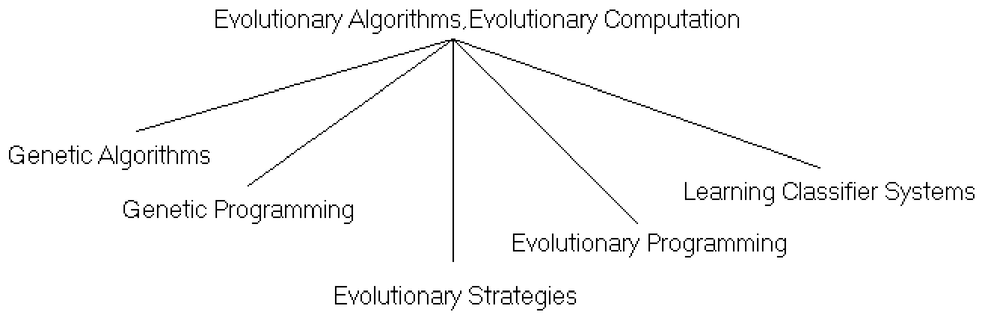

Figure 2.5 Classification of EA methods

The five sub categories in figure 2.5 have different ways to represent solutions and different selection, crossover and mutation methods. See Streichert (2002) for a detailed description of evolutionary algorithms.

2.3 Evolving ANNs for mobile robots

When evolving ANNs using an evolutionary algorithm, the candidate solutions can be represented as a list of floating point values, each value corresponding to one weight of the ANN. It is also possible to let the topology and activation functions be evolved.

It was stated above that evolutionary algorithms can take a long time to finish. This is specially true when evolving ANNs for mobile robots. It takes a long time since the behaviour of the robot and its interactions with the environment must be simulated in order to evaluate the ANN. In the case where a real robot and environment is used instead of a simulator the evaluation is even more time consuming.

Nolfi and Floreano (2000) presented some guidelines when designing fitness functions for evolved robots. They identified three dimensions describing fitness functions. The following descriptions of the three dimension are based on Nolfi's and Floreano's own descriptions.

- The dimension *functional-behavioral* indicates whether the fitness function rates specific functioning modes of the controller or whether it rates the behavioral outcome of the controller. For example, when evolving a legged robot to walk a functional fitness function would rate the movement patterns of the legs, while a behavioral fitness function would rate the distance the robot covered within a certain time period.
- The dimension *explicit-implicit* defines the amount of variables and constraints included in the function. For example, when evolving a robot to survive in a dangerous environment, an explicit function would include many variables like distance to nearest dangerous area, while an implicit fitness function would only rate how long the robot manages to survive.
- The dimension *external-internal* indicates whether the variables and constraints included in the fitness function are computed using information available to the evolving agent.

If the goal of the evolution is to optimise a set of parameters for a very complex, but well defined, control problem in a well controlled environment, then Nolfi and Floreano (2000) recommend a Functional, Explicit and External fitness function. If, on the other hand, the goal is to evolve a robot capable of autonomous operation in partially unknown and unpredictable environments without human intervention, they recommend a Behavioral, Implicit, and Internal fitness function.

2.4 Imitation studies

There has been many attempts to define imitation. As mentioned in the introduction, true imitation is distinguished from mimicry. According to Schaal (1999) true imitation is present only if i) the imitated behaviour is new for the imitator, ii) the same task strategy as that of the demonstrator is employed, and iii) the same task goal is accomplished. Other definitions are proposed by Baldwin (1901), Thorndike (1911), Bakker and Kuniyoshi (1996), and Tomasello et al. (1999). There has been many studies of animals' and children's ability to imitate, see for example Meltzoff et al. (1983) and Tomasello et al. (1993).

Whiten (2000) presents a taxonomy of 'mimetic' processes by which the act of one individual, B, may be caused to resemble those of individual A. In this taxonomy imitation is one of four mimetic processes in the category "social learnings". The other three are: *Stimulus Enhancement* where B learns from A to what object or location to orient behaviour, *Observational Conditioning* where B learns to what circumstances a behaviour should be a response, and *Goal Emulation* where B learns from A the goal to pursue.

Novelty and other issues regarding the general nature of imitation as discussed in animal imitation research usually do not play a significant role in robotics imitation research (Dautenhahn and Nehaniv 2002).

> "In the robotics community the term ‘imitation’ is often used in a much broader sense than in the animal literature. The primary goal of most computer scientists and roboticists is to create artifacts that show certain skills or behaviors (e.g. that can learn by demonstration from a human), which is very different from the possible interest of a biologist in understanding the exact nature and the ecological or evolutionary context of a certain behavior in relation to other naturally occurring behaviors in a given species or related species... Although ‘imitation’ is often used in robotics without very precise definitions, robotics addresses the challenge to precisely define, synthesize and experimentally test mechanisms and procedures of perception, action and learning involved in imitative learning" (Alissandrakis et al. 2004).

One of the problems designers are facing when building imitating robots is how the robot can map the observation of an action to the execution of the same action. The next section presents a number of approaches to building imitating robots.

2.5 Related work

There has been much research about imitation in robotics, see Schaal (1999) for a review. Imitation learning in robotics often focuses on efficient motor learning, the connection between action and perception, and modular motor control in form of motor primitives. Imitation can have another important role in robotics, besides skill acquisition, namely it might serve as a stepping stone towards the development of the kinds of social cognition found in humans, and possibly other animals (Alissandrakis et al. 2004).

This section presents a number of approaches to the problem of designing imitating robots proposed by different researchers. They all use different training methods and different robots with different control architectures of varying complexity. Some of them focus on learning a robot to imitate (mimic) a demonstrator, while others focus on letting a robot learn a task through imitation. One of them investigates unsupervised imitation while the others use explicit imitation. The work in this thesis and the work presented in this section all have in common that they involve a robot that is supposed to mimic or imitate an agent. The work in this thesis focuses on letting an evolved robot solve a task through unsupervised imitation.

Gatsoulis et al.(2002) carried out some experiments where a Khepera robot learned to locate "food" and bring it back to "home" in an environment with no obstacles. The robots got input from a number of sensors including an overhead camera that provided the positions and orientations of both the learner and the demonstrator. During training the demonstrator would perform a step, send information about its current state to the learner and then wait until the learner had completed the same step before proceeding to the next step. The learner usually learned to collect most of the food in the environment and was able to generalize beyond its training environment.

Billard (2000) studied how simulated humanoid robots could learn to mimic different types of movement sequences using biologically inspired control modules which were high-level abstractions of the spinal cord, the primary and premotor cortexes, the cerebellum, and the temporal cortex. The experimenter decided when the learner should learn and when it should start the rehearsal. Results showed that learning of the sequences was correct to the extent that each step of the sequence was reproduced although the movements were not repeated at exactly the same durations.

Andry et al. (2000) carried out a number of experiments with mimicking robots. In the first experiment a controller of a robot arm learned to imitate movements made by another arm. A camera was placed above, and facing the arm. In the first phase the student robot learned correspondence between its arms internal state and its position in its visual field. When the camera in the second phase was facing another arm, the controller tried to reduce the difference between the current state of its arm and the state it associated with the current visual input, and thereby mimicked the movements of the other arm. In the last experiment a simple network tried to predict the received input. The input of the network was three nodes which were activated whenever a human pressed any of three corresponding keys. The output was three nodes representing the predicted activation of the input nodes for the next timestep. The learning process was unsupervised, and the network learned to spontaneously imitate the keystrokes made by the human as long as they were regular.

None of the previous work investigates the ability of evolved robot to solve a task through unsupervised imitation of a demonstrator, which is the focus of this thesis. In the next chapter the goals of this thesis are presented.

3 Problem statement

This thesis investigates unsupervised imitation in evolved robots. In this chapter the aims and objectives of this thesis are formally stated.

Work on imitation in robotics is motivated by the need to find easy ways to program robots. A robot that is able to imitate could be programmed by just demonstrating a task. Work on unsupervised imitation is motivated by the fact that robots that are able to spontaneously imitate a demonstrator without supervision are more autonomous (Andry et al. 2000). Robots that are able to imitate without supervision have the advantage that they do not need explicit instructions about when, what and how to imitate, which make them more easy to design.

There has been previous work where robots learn to imitate and where robots learn tasks through imitation. Andry et al. (2000) investigated unsupervised imitation in a very simple system which learned to predict keystrokes. There has to the author's knowledge not been any previous work where mobile robots learn a task through unsupervised imitation.

The overall aim of this thesis is to investigate whether evolved robots can solve a task by imitating a demonstrator without being explicitly instructed to follow the demonstrator. If they can, then the presence of the demonstrator should accelerate the evolutionary process, i.e. robots that have the opportunity to observe the demonstrator should learn the task faster than robots that do not have this opportunity.

There are many ways to train robots, in this thesis the robots are trained with artificial evolution since it is a method that has been successfully applied to a wide range of problems in robotics (Nolfi and Floreano 2000). Since there is no definition of unsupervised imitation, it will be necessary to clarify the concept. In this thesis unsupervised imitation refers to when a robot learns a task faster when it has the opportunity to observe how a demonstrator performs the task, while not being explicitly instructed to imitate the demonstrator. A more detailed definition of unsupervised imitation can be found in chapter 4.

One way to reach the aim is to perform a number experiments with two different setups where robots are being evolved to solve certain tasks. In the first set of experiments the robots are alone and must solve the task on their own. The second setup is identical to the first one with the only difference that a demonstrator is present. The demonstrator performs the task the robot is supposed to solve. The training time needed to solve the task for robots who can watch a demonstrator is compared to the training time needed for robots without demonstrators. If the robots that have the opportunity to observe the demonstrator are not explicitly instructed to imitate the demonstrator, and if the training time is reduced compared to the training time of the robots that can not observe a demonstrator, then it can be concluded that the robots solve the task through unsupervised imitation of the demonstrator. This thesis presents a set of experiments in a simulator where the robots must learn to find and go through a number of gates.

Even if unsupervised imitation takes place in one experimental setup, it does not necessarily mean that evolved robots will always solve a task through unsupervised imitation of demonstrators when they get the opportunity to do so. There are many parameters in the experiments that can affect the robots' tendency to imitate, for instance the robot’s morphology, the task, and the environment. In order to be able to conclude something about evolved robots' tendency to solve tasks through unsupervised imitation in general, it must be investigated how the different parameters affect the results. It is beyond the scope of this thesis to explore the effect of all of the parameters. Instead, this thesis explores one of the parameters, namely the degree of difficulty of the task the robot is supposed to learn.

The overall difficulty of a task is hard to define and measure, but the degree of difficulty to learn a task using a certain training method could be estimated by measuring the time required to learn the task when no demonstrator is present. In the case of an evolutionary algorithm solving a task is the same as finding an individual with sufficient fitness. The degree of difficulty of finding an individual with sufficient fitness for a given fitness function can be measured by counting the number of generations required on average to produce an individual with sufficient fitness. A secondary aim this thesis is to explore the relation between the degree of difficulty of solving a task using an evolutionary algorithm and the robot's tendency to imitate

In summary the overall goal is to investigate if evolved robots will solve a task faster if they get the opportunity to observe a demonstrator perform the task, and not to find the best controller architecture for the robot or the best set of parameters for the evolutionary algorithm for any given problem.

3.1 Aims

The aim of this thesis is twofold:

- Investigate if giving evolved robots the opportunity to unsupervised imitate a demonstrator can speed up the evolutionary process when evolving robots to solve a certain task.
- If unsupervised imitation takes place, then explore the relation between the degree of difficulty of the task and the learner's tendency to imitate.

3.2 Objectives

In order to achieve the aims the following objectives must be attained:

- Study how the phrase "unsupervised imitation" has been used in the literature and propose a definition suitable for robotics and analyse the consequences of the proposed definition. This is necessary since there is no formal definition of unsupervised imitation.
- Investigate if unsupervised imitation takes place in a given test scenario. This involves performing a set of experiments where robots both with and without demonstrators are being evolved to solve the task, and analysing the results.
- If unsupervised imitation takes place, then perform another set of experiments where the degree of difficulty of the task is varied and measure how it affects the learner's tendency to imitate. The degree of difficulty will be varied by changing the arrangement of the walls in the environment.

3.3 Contributions

As mentioned above, work on imitation in robotics is motivated by the need to find easy ways to program robots. There has been previous work investigating learning through imitation where the learner and demonstrator are forced to synchronize with each other by the experimenter, see for example (Hayes and Demiris 1994). There has also been work investigating unsupervised imitation in a very simple system which learned to predict keystrokes on three different keys (Andry et al. 2000). However, there has to the author's knowledge not been any previous work investigating unsupervised imitation in mobile robots.

This thesis focuses on learning through unsupervised imitation in evolved and mobile robots, and therefore explores a previously unexplored area in the field of imitation learning. Since robots that imitate without supervision are more autonomous than those that are forced to imitate by the designer (Andry et al. 2000), this thesis contributes to the field of autonomous robotics by showing that *unsupervised imitation*, as opposed to synchronized, is possible in evolved robots.

4 Unsupervised imitation

This chapter proposes a definition of unsupervised imitation suitable for robotics. Section 4.1 presents a number of definitions of imitation and in section 4.2 a definition of unsupervised imitation is proposed, based on the definitions in 4.1.

4.1 Imitation

Imitation has a long history of study in ethology and psychology, and there are many definitions of what imitation really is. In a dictionary edited by Baldwin (1901) three definitions of imitation are presented:

"The performance in movement, thought, or both movement and thought, of what comes through the senses, or by suggestion, as belonging to another individual."

"Any repetition in thought, action, or both, which reinstates a copy."

"An organic reaction of the stimulus-repeating or self-sustaining type."

None of the three definitions from the dictionary stress that it is necessary to learn something in order to imitate. Thorndike (1911) included learning when he defined imitation as:

"Learning to do an act from seeing it done."

Bakker and Kuniyoshi (1996) study imitation in robotics and propose the following definition of imitation:

"Imitation takes place when an agent learns a behaviour from observing the execution of that behaviour by a teacher."

Bakker and Kuniyoshi point out that the roles of 'teacher' and 'agent' are not fixed, but can be reversed from one encounter to the next.

When studying animals, true imitation is distinguished from mimicry. As mentioned above, true imitation is the ability to replicate and, by so doing, learn skills that are not part of the animal's prior repertoire by observation of those performed by others, while mimicry is the ability to replicate a behaviour that is usually part of the usual animal repertoire (Billard 2002). Thorpe (1963) proposes another definition of true imitation: "copying of a novel or otherwise improbable act or utterance, or some act for which there is clearly no instinctive tendency". Bakker and Kuniyoshi (1996) argues that in robot imitation the interest lies in true imitation where a behaviour is observed, understood and reproduced. The proposed definition of unsupervised imitation in the next section is based on Bakker's and Kuniyoshi's definition of imitation.

4.2 Definition of unsupervised imitation

This section proposes a definition of unsupervised imitation. First a number of criteria that must be met when a robot learns a task through unsupervised imitation are presented. After the criteria have been presented a definition of unsupervised imitation is proposed.

In this thesis, *unsupervised imitation* refers to the situation where the imitation is not compelled nor guided. If a robot should learn a task through unsupervised imitation of a demonstrator, then the robot must not be forced, or programmed to follow the demonstrator. The robot may still be supervised to learn the task, but it must discover without outside intervention that it can learn the task by imitating the demonstrator. In order to be able to say that a robot has not been programmed to imitate the demonstrator, the designer of the robot must not have put any variables in the robot's control programme or in its fitness function that represent anything about the demonstrator. For instance, there must not be any variable representing the distance between the robot and the demonstrator in the fitness function. Variables representing different attributes of the demonstrator are only allowed if the meaning of the variables emerge as a consequence of the robot's interactions with the environment during the robot's training, as opposed to being defined in advance by the designer. A hidden node in an ANN with randomly initialised weights is an example of a variable whose meaning emerges.

Below are the five criteria that must be met when a robot learns a task through unsupervised imitation of a demonstrator. The first two criteria are derived from the paragraph above. The three other criteria are derived from Bakker's and Kuniyoshi's (1996) definition of imitation presented in the previous section. As mentioned in section 2.4, Schaal (1999) argues that true imitation is present only if i) the imitated behaviour is new for the imitator, ii) the same task strategy as that of the demonstrator is employed, and iii) the same task goal is accomplished. There are several problems when demanding novelty of imitated behaviour (Dautenhahn and Nehaniv 2002). For instance, parts of the imitated behaviour may already be familiar to the imitator, but not the combination of the parts. Since it is hard to define what makes a behaviour *new* to an imitator, the proposed definition below does not require the behaviour to be new to the imitator, but instead requires the imitator to learn the task faster than it would have done if no demonstrator was present. This eliminates the cases where the imitator already knows how to solve the task. Schaal's second requirement, that the imitator employs the same task strategy, is hard to define. Given proximal (from the point of view of the imitator or the demonstrator) descriptions of the behaviours of the imitator and the demonstrator, it can be hard to determine whether they use the same strategy if the control architecture of the imitator and the demonstrator differs. Given distal (from an external point of view) descriptions of the behaviours, it can be hard to determine whether they use the same strategy if the robots morphology differs. The proposed definition below does not require the imitator to use the same strategy as the demonstrator, but do require the robot to solve the task.

- The robot must not be programmed to, or rewarded for following the demonstrator. This means that the designer of the robot must not have put any variables in the robot's control programme or in its fitness function that represent anything about the demonstrator.
- The robot must not be forced to follow the demonstrator by any other agent, including the demonstrator.
- The robot must be able to observe the demonstrator.
- The demonstrator must do something, i.e. perform a task.
- The robot must learn the task, and learn the task faster than it would have done if it had not been able to observe the demonstrator.

The fifth criterion states that the robot must learn the task, which means that the robot must also be able to perform the task after the demonstrator has been removed. Based on these criteria the following definition is proposed in this thesis:

A robot is learning a task through unsupervised imitation of a demonstrator if and only if the robot is not being explicitly programmed to, nor forced to follow the demonstrator and if it learns the task faster than it would have done if it had not been able to observe the demonstrator and if the robot is able to repeat the task after the demonstrator has been removed.

In order to clarify what the definition of unsupervised imitation, the rest of this chapter presents examples of behaviours. Some of them are, while some of them are not, classified as unsupervised imitation according to the definition proposed above.

A robot is being evolved to find the exit of a maze. The start position is always the same and the robot must find the exit within 60 seconds. The robot receives fitness based on how long time it needs to reach the exit. At first the robot is alone in the maze, and it takes 100 training epochs before the robot learns to find the exit within 60 seconds. Then the memory of the robot is reset and the experiment is repeated, but this time a demonstrator is present. A rope is tied to the demonstrator and the robot. The demonstrator drags the robot through the maze to the exit in the first 50 training epochs and is then removed. After 10 epochs on its own, in epoch 60, the robot reaches the exit in less than 60 seconds. Even though the training time was reduced, the robot did not learn the task through unsupervised imitation since the robot was forced to follow the demonstrator.

Another set of experiments are performed. The setup is identical to that of the previous example, except this time there is no rope tied to the robot and the fitness function of the robot is slightly modified. When the demonstrator is present, the average distance between the robot and the demonstrator during the epoch is calculated and subtracted from the robot's fitness. The robot quickly learns to follow the demonstrator and as in the previous example the demonstrator is removed after 50 epochs. When the demonstrator is removed the average fitness drops, but after 10 epochs the robot manages to reach the exit within 60 seconds. As in the previous example, the training time was reduced but the robot did not learn the task through unsupervised imitation since it was programmed to follow the demonstrator.

A third set of experiments are performed. The setup is identical to that of the first example except for the rope. This time the fitness function remains the same, and the average distance between the robot and the demonstrator is not calculated. The robot learns to follow the demonstrator, but when the demonstrator is removed in epoch 50, the average fitness drops and the robot does not learn to reach the exit until epoch 120. This time the training time was not reduced and therefore the robot did not learn the task through unsupervised imitation.

Finally, a fourth set of experiments are performed. The setup is identical to that of the third example. As in the previous example, the robot learns to follow the demonstrator, and the demonstrator is removed after 50 epochs. This time the robot learns to reach the exit in epoch 60 and has therefore learned the task through unsupervised imitation of the demonstrator since the training time was reduced and the robot was not forced to nor programmed to follow the demonstrator.

5 A test scenario for unsupervised imitation

The second objective of this thesis involves determining a test scenario and performing a set of experiments where robots are being evolved to solve a task, and analysing the results to see if unsupervised imitation takes place. This chapter describes the experiments made to reach this objective. The task the robot is supposed to solve is a navigation task and is presented in section 5.1. Section 5.2 gives a short presentation of the Khepera I robot, which is the robot that was chosen for the experiments. In order to save time the experiments were made in a simulator. Section 5.3 describes the simulator used in the experiments. The robots in the experiments are controlled by ANNs trained with an EA. The ANN and the EA used in the experiments are described in section 5.4 and 5.5 respectively. The results of the experiments are presented in section 5.6 and are analysed in section 5.7. The choice to use Khepera robots controlled by evolved ANNs is motivated by the fact that this approach has been successfully applied to a wide range of problems, see for example Nolfi and Floreano (2000). The navigation task is chosen because navigation tasks are easy to set up in the simulator, and there have been previous work where Khepera robots were trained to navigate through mazes.

5.1 A navigation task

The robot is placed into an environment and is being evolved to navigate through three gates (see figure 5.1). It must pass the gates in the correct order, and receives 10 fitness points for every gate it reaches. In order to promote speed, the robot also receives a small penalty for every gate it passes equal to $t * 0.001$, where t is the number of timesteps elapsed when the robot reaches the gate. Thus the fitness f can be calculated according to the following formula:

$$f = 10 * g - 0.001 * (t1 + t2 + t3)$$

Where f is the fitness, g the number of gates the robot manages to reach in the right order and ti is the timestep at which the robot passes gate i or 0 if the robot never reaches gate i. One timestep equals 100 ms. An epoch never lasts more than 800 timesteps and is terminated at timestep 400 if the robot has not yet passed the first gate, and at timestep 600 if the robot has not reached the second gate.

The environment into which the robot is placed is surrounded by walls forming a rectangle measuring 1300x575 mm. The positions of the gates are marked in figure 5.1. Each gate is placed at the beginning of a corridor. The corridors are 75 millimetres wide, which is approximately 136 percent of the diameter of the robot. The gates themselves are not visible by the robot, but are only imaginary lines used by the fitness function. There is a zone with a special colour on the ground surrounding each gate. The two diagonal walls near each gate have a lighter colour than all the other walls in the environment. The robot always starts each epoch at the same position in the top right corner facing left. When a demonstrator is present, it always starts at a position 150 millimetres in front of the robot.

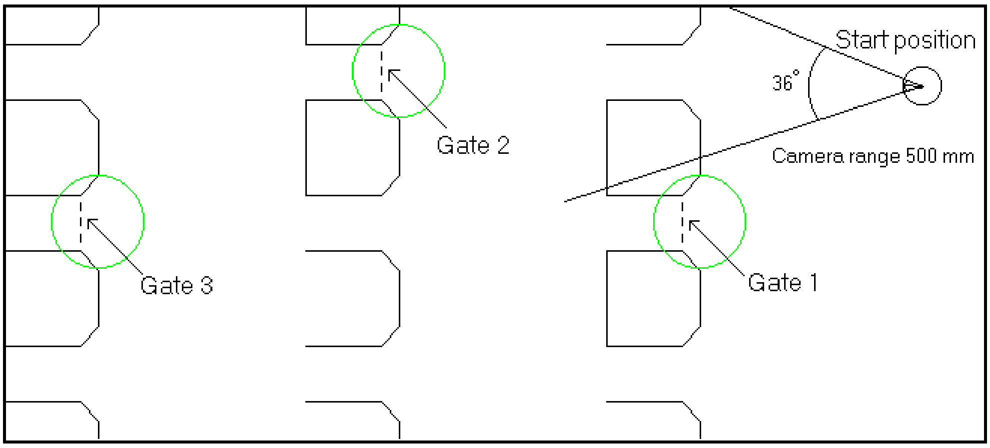

Figure 5.1 The environment. The dashed lines mark the gates. The ground within the circles has a special colour detectable by the ground sensor of the robot. There is a camera mounted on the robot which has a field of view of 36 degrees and a range of 500 mm.

In the first experiment the robot is alone in the environment, and the evolutionary algorithm runs for 120 generations. The second experiment is identical to the first one, except for the presence of a demonstrator. The demonstrator is another Khepera robot that follows an almost optimal path through the three gates and will always stand still whenever the distance between the demonstrator and the other robot exceeds 200 mm. In order to test if the robot has learnt the task the demonstrator is removed after 70 generations which means that the robot is alone for the last 50 generations. Both experiments are repeated 30 times.

5.2 Khepera robot

The Khepera I robot shown in figure 5.2 and developed by K-Team is a small robot originally designed as a research and teaching tool. The robot is controlled by a Motorola 68331 processor with 256 KB RAM. It is equipped with two wheels and a number of sensors. The sensors used in the experiments in this report are described in more detail in section 5.4. The robot has a diameter of 55 mm, and is 30 mm high. There are several modules that the Khepera robot can be equipped with, including gripper, radio, and different kind of cameras.

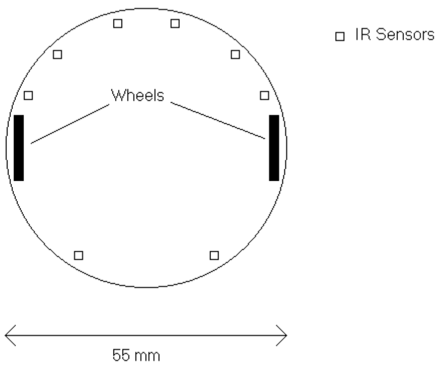

Figure 5.2 The Khepera robot and the locations of its IR-sensors.

5.3 YAKS

Yet Another Khepera Simulator (YAKS) is a simulator of K-Team's Khepera I robot and was developed at the University of Skövde. YAKS can simulate multiple robots in an environment made up of walls, light sources, and moveable obstacles. The simulator supports construction of ANNs and allows the user to specify the topology of the networks. An evolutionary algorithm is implemented that evolves the ANNs according to a user defined fitness function. YAKS has a graphical user interface which makes it easy to observe the behaviours of the simulated robots. More information can be found at the YAKS homepage at: http://r2d2.ida.his.se/

5.4 Controller architecture

This section describes the sensors and motors of the robot used in the experiment as well as the ANN that connects the input nodes to the output. Figure 5.3 gives an overview of the controller.

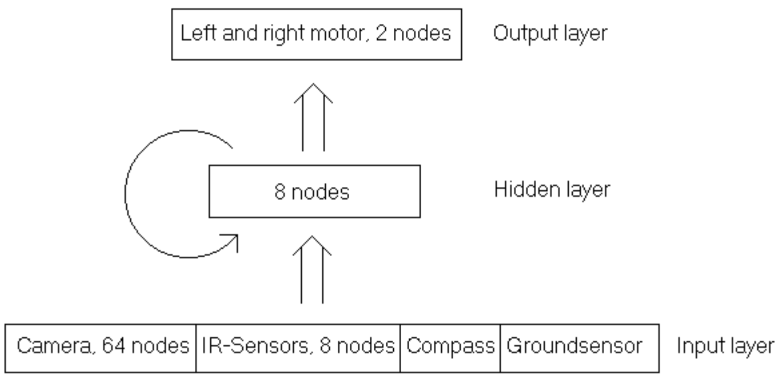

Figure 5.3 The RNN controlling the robot. Each node in the input layer is connected to all of the eight nodes in the hidden layer. All of the nodes in the hidden layer are connected to the two nodes in the output layer and to all the nodes in the hidden layer.

The robot receives input from eight infrared sensors, a camera, a compass and a ground sensor. All of the sensors are connected to a number of the input nodes of the ANN.

The IR-sensors measure the distance to the nearest object and the values are scaled between 0 and 1, where 0 means there is no object within the sensors range and 1 means an object is very near. The sensors have a range of 50 mm. Six of the IR-sensors are located at the front and two at the back of the robot.

A K213 Vision Turret is used on the top of the robot. The horizontal plane is projected on an one-dimensional image of 64 pixels. The 64 pixels covers a field of view of 36 degrees and each pixel has a resolution of 256 grey levels. In the simulator the vision turret has a range of 500 mm.

The compass measures the heading of the robot. The value is scaled between 0 and 1, where a value of 0 or 1 indicates that the robot is facing the same direction as the positive x-axis (east), and a value of 0.25 indicates the robot is facing the same direction as the positive y-axis (south in the coordinate system used by the simulator). The compass only exists in the simulator, the real Khepera I robot does not have one.

The ground sensor measures the colour of the ground below the center of the robot. In the simulator, only two values can be returned: 0 if the center of the robot is not inside any of the zones in the environment and 1 if the center of the robot is inside at least one of the zones. The ground sensor only exits in the simulator, the real Khepera I robot does not have one.

The robot has two motors connected to the two wheels of the robot. Each motor is connected to one of the output nodes of the ANN. A node activation of 0 sets the motor to full reverse, 0.5 stops the wheel, and 1.0 sets the motor to full forward. The maximum speed is 600 mm/s.

The ANN is a recurrent neural network with 74 input nodes, eight hidden nodes, and two output nodes. Eight of the input nodes are connected to the IR-sensors, 64 to the camera, one to the compass sensor and one to the ground sensor. The two output nodes are connected to the motors. Each layer is fully connected to the layer above, and the hidden layer is also fully connected to itself. See figure 5.3.

5.5 Parameters for the EA

The robots are controlled by ANNs with a fixed topology. The weights of the ANNs are evolved using an EA. Each solution is represented by an array of 682 floating point values, each corresponding to one of the weights of the network. Each generation has a population of 100 candidate solutions. The weights of the solutions of the first generation are randomly initialized. For every generation, all of the candidate solutions are evaluated and assigned a fitness value. The fitness function is described in section 5.1. An elite selection method is used. In every generation the 25 candidate solutions with the highest fitness are kept as parents. For each parent three offspring are generated by randomly changing some of the weights of the parent. The 25 parents and the 75 offspring constitute the next generation.

5.6 Results

The average fitness values of the best individual of each generation are presented in figure 5.4. The fitness values are shown for both the robots that were evolved alone and for the robots that were evolved with a demonstrator.

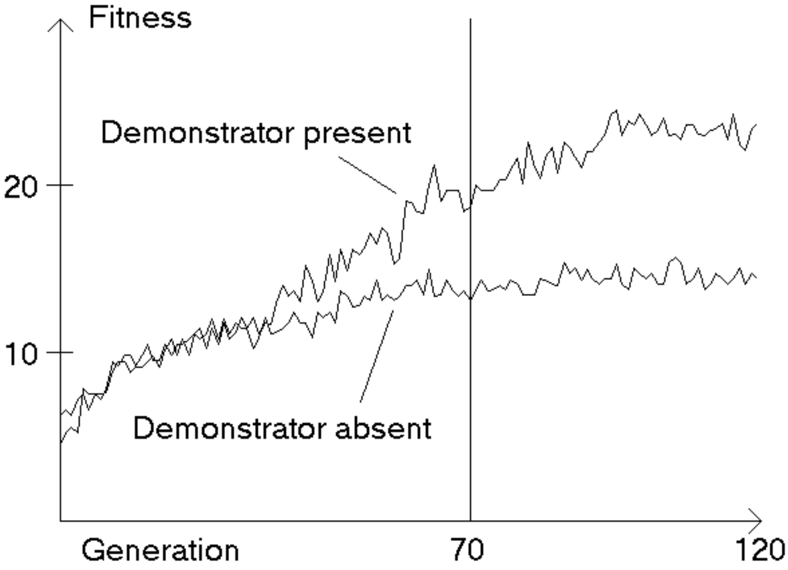

Figure 5.4 The average fitness value of the best individual of each generation. One of the two curves shows the fitness for the robots evolved without demonstrator while the other shows the fitness for robots evolved with a demonstrator present in the first 70 generations. The average values are based on 30 repetitions of the experiments.

A summary of the performance of the best individuals of generation 120 is presented in figure 5.5.

Number of gates reached	1	2	3
Demonstrator present	5	7	18
Demonstrator absent	22	2	6

Figure 5.5 The number of repetitions of the experiments where the best individual of generation 120 reached 1, 2 or 3 gates. In 18 of the 30 repetitions the best robot of generation 120 managed to reach all three gates when the demonstrator was present. When the demonstrator was absent the best robot of generation 120 reached all three gates in only six of the 30 repetitions.

As the curves in figure 5.4 show, the fitness of the robots evolved with a demonstrator is almost equal to the fitness of the robots evolved without demonstrator for the first 30 generations. During the first 30 generations the robots learn to find the first gate and the presence of the demonstrator does not seem to have any effect on the fitness. However, in the generations that follow most of the robots who can watch a demonstrator learn to find the second and third gate while most of the robots that are evolved without any demonstrator do not. This is also shown in figure 5.5 which shows that in most cases the robots evolved without a demonstrator did not learn how to find the second or third gate within 120 generations, while the robots who could watch a demonstrator in most cases found all three gates. The curves in figure 5.4 also show that the robots evolved with a demonstrator learned to perform the task in the demonstrator's absence, since the demonstrator is not present after generation 70.

The second objective in this thesis is to investigate if unsupervised imitation takes place in a given test scenario. The task is to go through the three gates in the right order as fast as possible and the results show that unsupervised imitation takes place in the scenario presented above.

5.7 Analysis

In an attempt to explain how the robots can learn the task through unsupervised imitation, the behaviours of the robots are analysed in this section. The analysis focuses on typical behaviours in three cases. In the first case the robot has been evolved for 70 generations with a demonstrator and had the opportunity to observe the demonstrator. In the second case the robot has been evolved for 70 generations with a demonstrator, but the demonstrator is removed. In the third case the robot has been evolved for 120 generations without any demonstrator, and no demonstrator is present.

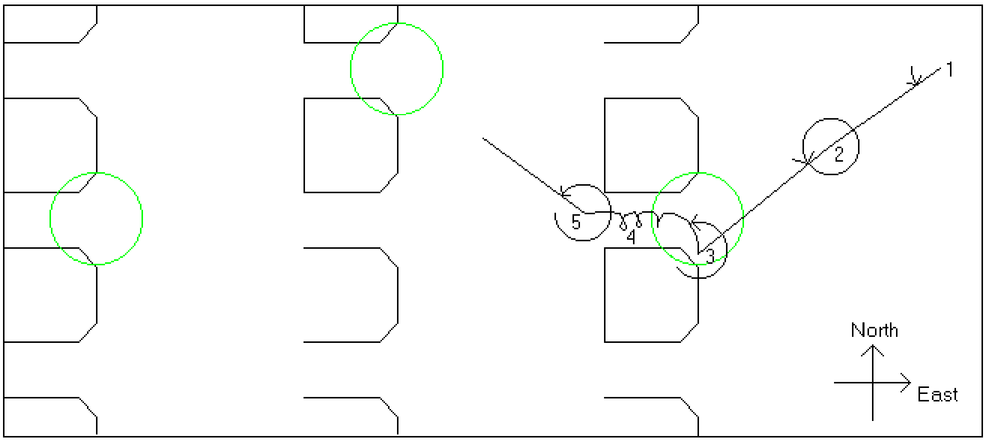

Figure 5.6 A typical behaviour of the best robot of generation 70 when a demonstrator is present. The robot was evolved with a demonstrator present.

In the 30 repetitions of the experiments a number of unique solutions was found, but in most of the repetitions with the same experimental setup the solutions found were quite similar. In the experiments where a demonstrator was present, the robots with the highest fitness in generation 70 usually behaved in a way that is shown in figure 5.6. The robot starts in the north east corner (1) facing west with the demonstrator in front of it. The first thing the robot does is to rotate counter clockwise until it has the southern part of the first gate in the center of its field of view. It then moves forward approximately 200 mm and then stops (2) and rotates about 360 degrees. After that it runs into the diagonal wall south of the gate (3). The IR-sensors are activated for the first time when the robot gets near the diagonal wall and remains active until the robot has gone though the corridor. After hitting the diagonal wall the robot starts rotating counter clockwise. It starts moving forward when facing north, but continues to turn left. Once it is inside the corridor (4) it keeps rotating and moves forward whenever the demonstrator is in the field of view. The robot also drives forward at low speed when facing north and south. At this point the demonstrator is usually located 200 mm west of the robot. When the robot has gone through the corridor (5) it rotates counter clockwise and starts moving forward when the second gate or the demonstrator is in the field of view. At this point the demonstrator is usually located between the robot and the second gate. The robot goes through the second and third gate almost the same way as it goes through the first gate.

As mentioned above, the colour of the diagonal walls near the gates are lighter than all the other walls in the environment. Parts of the demonstrator also have a colour that is lighter than the walls in the environment. The robot seems to find the way through the three gates by simply rotating and moving forward when there is a bright light in the field of view. However, when the robot is between the two diagonal walls on each side of the gate the robot is surrounded by three objects with light colours, the two walls and the demonstrator. If the robot chooses to follow the demonstrator and follows it all the way to the next gate, then it will receive higher fitness. It is therefore likely that robots who move towards bright lights, but not towards the diagonal walls when standing near the gates, will be evolved. It is possible for the robot to learn not to move towards the diagonal walls when standing near the gates between the diagonal walls since the IR-sensors and the compass inputs distinguishable patterns when the robot is facing the north diagonal wall, the south diagonal wall, or the demonstrator.

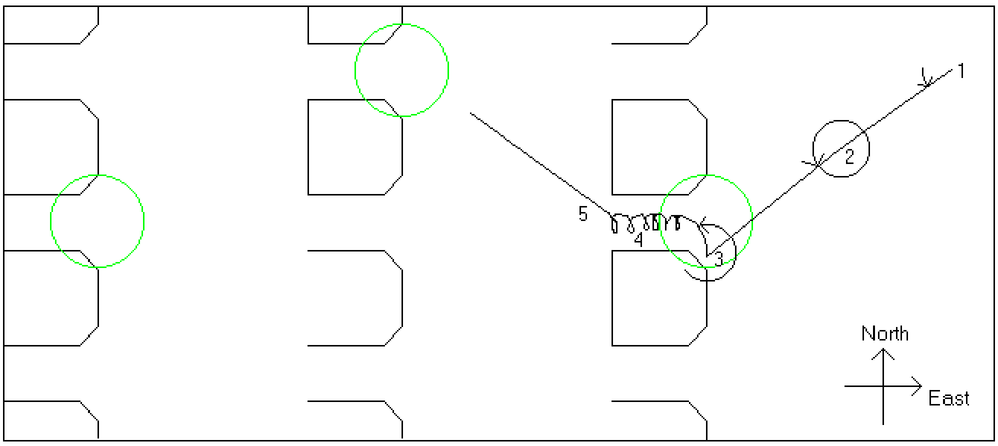

Figure 5.7 A typical behaviour of the best robot of generation 70 when the demonstrator is absent. The robot was evolved with a demonstrator present.

Robots that behave as described above when a demonstrator is present, often behave in a similar way when the demonstrator is removed. Figure 5.7 shows a typical behaviour of one of those robots when the demonstrator is absent. The robot behaves the same way as when the demonstrator is present, except when it is in the corridor (4). It still rotates in the corridor and moves forward at low speed when facing north and south. In the case where the demonstrator is present and the robot is facing west, it usually moves forward at high speed. But in this case where the demonstrator is absent, it only moves forward at low speed when facing west. The robot does not move at all when facing east. This way the robot is slowly finding its way west through the corridor until the second gate is within sight (5). It then drives towards the second gate at high speed and goes through the second and third gate almost the same way as it goes through the first gate.

When the robot has the diagonal walls in its field of view it moves forward the same way it does when the demonstrator is present. However, when the robot is inside a corridor it can not see the demonstrator. Still it moves forward when facing west, but does not move when facing east. One possible explanation to why the robot does not move east could be that even when the demonstrator was present the robot still had to learn not to move towards the diagonal walls when the robot is inside the corridor.

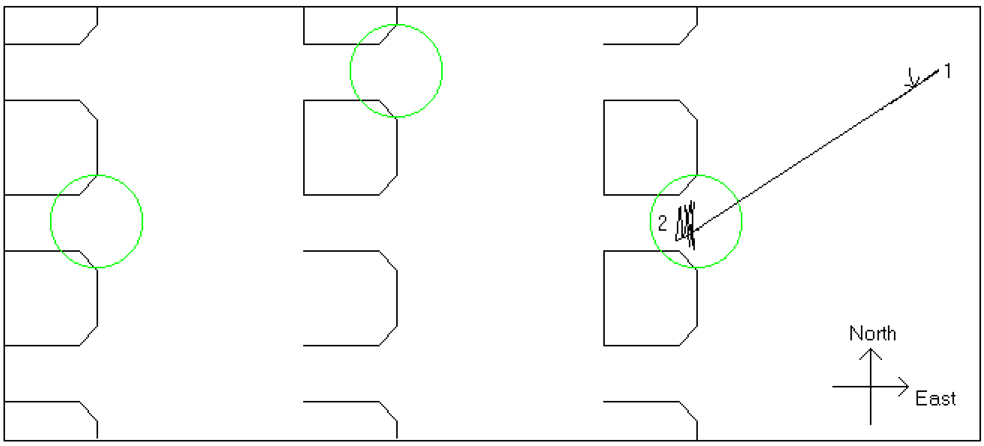

Figure 5.8 A typical behaviour of the best robot of generation 120 when the robots were evolved without any demonstrator present.

In most of the repetitions of the experiment where the robots were evolved without a demonstrator the robots only managed to reach the first gate. Figure 5.8 shows how the best robot of generation 120 behaves in most of the repetitions of the experiment where the demonstrator is absent. The robot starts (1) by rotating left until the first gate is in the center of its field of view and then moves forward at maximum speed until it runs into the south wall of the corridor (2). It then starts to move back and forth between the two diagonal walls and continues to do so for the rest of the epoch.

A robot that is evolved without demonstrator quickly learns to find the first gate by simply moving towards the lighter walls. Once it has reached the gate it has no motivation to go through the corridor and usually does not find the second gate. Robots that are evolved with a demonstrator also learn to move towards objects with light colours. Some of them will by chance move towards the demonstrator and will be rewarded with higher fitness if they follow the demonstrator all the way to the second gate. The presence of the demonstrators changes the fitness landscape. Assume the robots have already learnt to find the first gate. When the demonstrator is present, the robot only has to learn which light source it should follow in order to find the second gate. If, on the other hand, the demonstrator is absent, the robot has to learn to avoid the bright light of the diagonal walls once it reached the corridor. Since it probably found the first gate by moving towards the bright light of the diagonal walls it has to make extensive adjustments of its behaviour in order to avoid the same walls and be able to find the second gate. This leads to the following conclusion:

It is easier to learn the task in the demonstrator's presence than in its absence.

Even though it is easier to learn the task in the demonstrator’s presence, the robots will still have to overcome some of the problems associated with the task. They must learn to first move towards the diagonal walls near the gate, but must not move towards them when they are in the corridor. Since the robots learn to do this they avoid getting stuck near the diagonal walls by the first gate even when the demonstrator is absent. This leads to the following conclusion:

The robots that are trained in the demonstrator's presence find general solutions in the sense that they are able to repeat the task in the demonstrator's absence.

The two conclusions above are of course only valid for the experimental setup presented in this chapter. In order to be able to say anything about a demonstrators effect on a robot's training in general, many experiments with different setups would have to be performed. The two conclusions and the reasoning behind them can however be of help when explaining how the robots learn the task through unsupervised imitation.

In summary, this chapter shows that unsupervised imitation takes place in the presented test scenario. The task is to go through the three gates in the right order as fast as possible. However, if the task would be defined as "go through the first gate as fast as possible", then the results would show that unsupervised imitation has not taken place, since the robots usually learns to find the first gate at almost the same time no matter if a demonstrator is present or not. It is obvious that finding the first gate is easier than finding all three gates. In the next chapter it is investigated how the degree of difficulty of the task affects the robot's tendency to learn through unsupervised imitation.

6 The effect of changing the degree of difficulty

The third objective in this thesis is to explore how the degree of difficulty of the task affects the robot's tendency to imitate. The degree of difficulty of a task can be measured by counting the number of generations required to learn the task when no demonstrator is present. In this chapter two new sets of experiments are presented. In the first set the task is designed to be easier than the task presented in the previous chapter, while the task in the second set of experiments is designed to be more difficult. The two sets of experiments in this chapter are identical to the experiments presented in the previous chapter except for the arrangement of the walls in the environment and the number of generations the experiments are allowed to run. The first set of experiments with the easier task and the results are presented in section 6.1 and the experiments with the more difficult task and the results are presented in section 6.2. The two sets of experiments are repeated ten times in order to identify overall trends.

There are many ways to change the degree of difficulty of a task. For example, the distance between the gates, the colours of the walls or the resolution of the robots' sensors can be altered in order to change the degree of difficulty. In this chapter only two different experimental setups are presented. The results of the experiments are therefore not very general, but they can give an indication of how the degree of difficulty of the task affects the robots' tendency to imitate.

6.1 An easy task

In the easy task the walls have been rearranged and form one big corridor, see figure 6.1. The gates have the same positions as in the task in the previous chapter. With this arrangement of the walls, it is possible to reach all three gates by simply following the walls on one side of the corridor. Since this task is more easier the experiment will only run for 100 generations in order to save time. The demonstrator is removed after 50 generations.

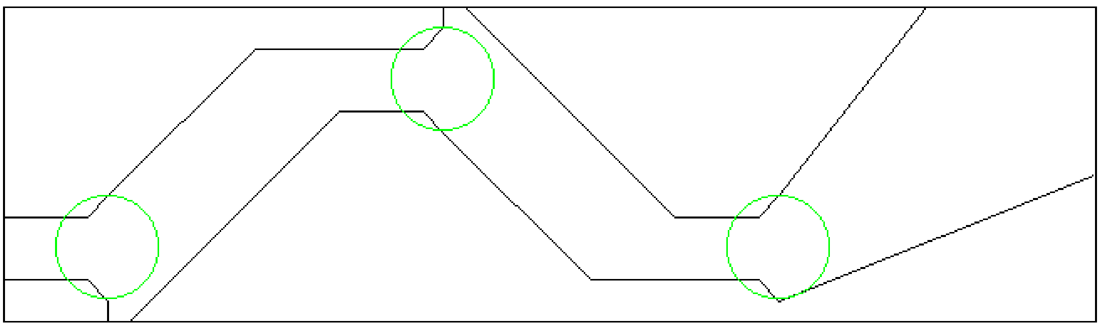

Figure 6.1 The environment in the easy task

The curves in figure 6.2 show the fitness of the robots with and without a demonstrator.

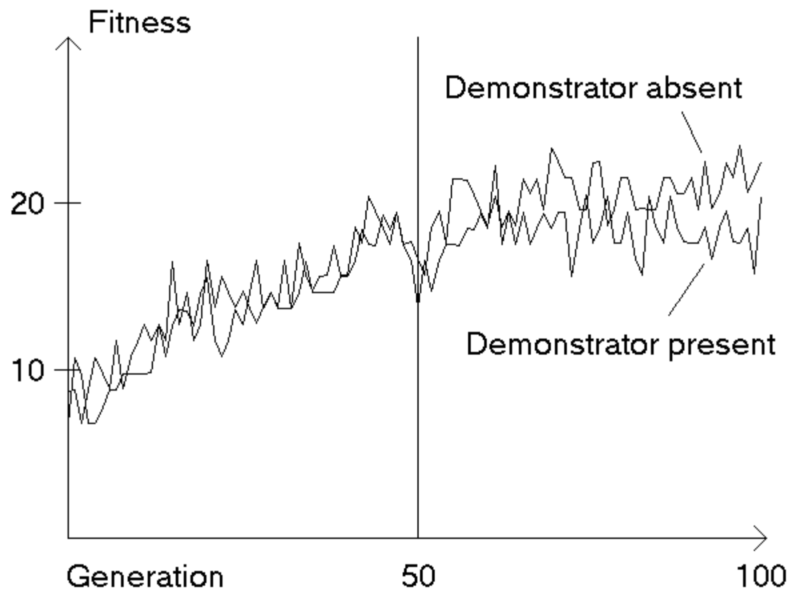

Figure 6.2 The fitness of the robots in the easy task

In most of the ten repetitions of the experiment the robots that were evolved without any demonstrator learned to find all three gates. The robots that were evolved without any demonstrator in the experiments in the previous chapter usually did not learn to find all three gates within 120 generations. Since the robots are more likely to learn the task presented in this section than the task in the previous chapter, it can be argued that the task in this section is indeed easier than the task presented in the previous chapter.

The curves in figure 6.2 show that the robots that are evolved with a demonstrator do not learn the task quicker than those that are evolved without any demonstrator. Hence the robots do not seem to learn the easy task through unsupervised imitation.

6.2 A difficult task

The environment in the more difficult task is identical to the one in chapter 5 except for an extra wall between the first and second gate, see figure 6.3. The extra wall blocks the view and thereby prevents the robot from seeing the second gate until it has gone around the wall.

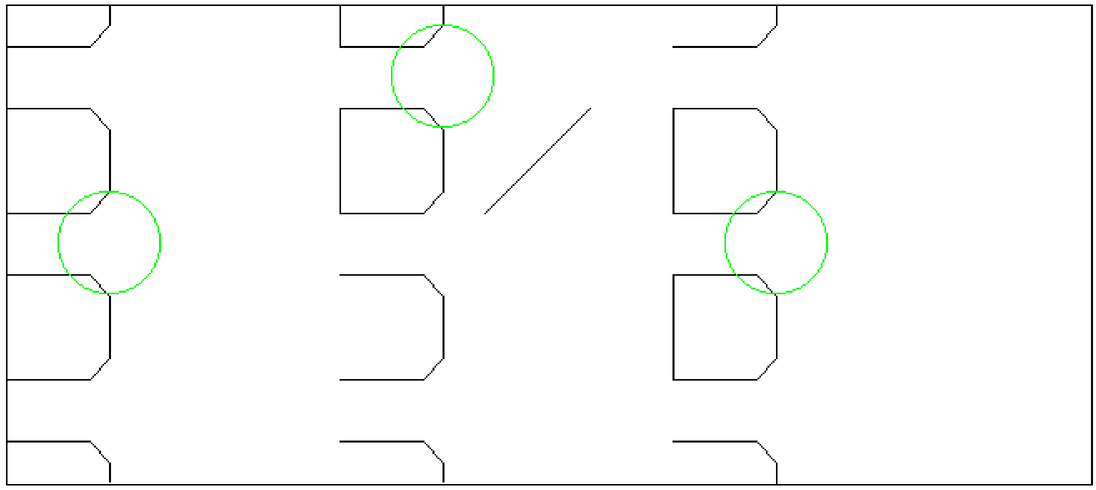

Figure 6.3 The environment in the difficult task.

Since this task is more difficult the experiments run for 200 generations. The demonstrator is removed after 150 generations. The curves in figure 6.4 show the performance of the robots with and without a demonstrator.

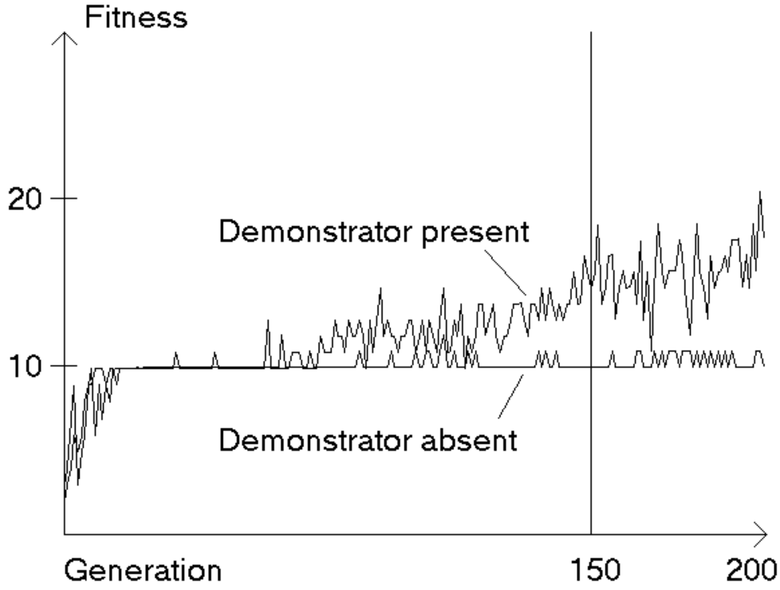

Figure 6.4 The fitness of the robots in the difficult task.

None of the robots evolved without the demonstrator managed to find all three gates within 200 generations. The robots evolved without any demonstrator in chapter five found all three gates in six cases of 30. It can therefore be argued that this task is indeed more difficult than the one presented in chapter five.

While none of the robots evolved without any demonstrator learned to find all three gates, the robots evolved with a demonstrator learned to find them in most of the repetitions of the experiment. As can be seen in figure 6.4, the robots that learned the task in the demonstrator's presence are able to repeat the task in the demonstrator's absence. Hence the robots learn the difficult task through unsupervised imitation.

6.3 Analysis

The results of the experiments in this chapter indicate that evolved robots tend to solve tasks through unsupervised imitation when the task is not too easy. The results indicate that the impact of having the opportunity to imitate a demonstrator is greater the more difficult the task is. One possible explanation of why the robots do not imitate when the task is easy could be that it is easier to learn the task than it is to learn to imitate the demonstrator. In order to imitate the demonstrator the robot must learn to locate and follow the demonstrator, and if this is more difficult than it is to learn the task itself then the robots will not benefit from having the opportunity to observe the demonstrator.

7 Conclusions

Imitation learning is a research area that has received much attention. One of the goals is to construct robots that a human could program by demonstrating the task the robot should do. In most of the previous work on imitating robots the learner and demonstrator are forced to synchronize with each other by the experimenter. This thesis presents another approach, unsupervised imitation, where the learner is not being explicitly designed to imitate the demonstrator. There has been previous work investigating unsupervised imitation in very simple systems. In this thesis it is investigated whether evolved mobile robots can solve tasks through unsupervised imitation.

The overall goal of this thesis is to investigate if evolved robots will solve a task faster if they get the opportunity to observe a demonstrator perform the task when the robots are not programmed to follow the demonstrator. A secondary goal of this thesis is to explore the relation between the degree of difficulty of the task the robot is supposed to solve and the robots' tendency to imitate.

Since there is no formal definition of unsupervised imitation, the concept is clarified in this dissertation. In this thesis unsupervised imitation refers to when a robot learns a task faster when it has the opportunity to observe how a demonstrator performs the task, while not being explicitly instructed to imitate the demonstrator.

In order to reach the goal this thesis presents a number of experiments with different setups where robots are evolved to solve tasks of varying difficulty. The tasks are a number of navigation tasks where the robots must learn to find and go through three gates. For each experimental setup two sets of experiments are made. In the first set the robots are evolved without any demonstrator. In the second set of experiments a demonstrator performing the task the learning robot is supposed to learn is present in the first generations and thereafter removed. The behaviour of the learning robot before and after the demonstrator is removed is analysed to determine if the learner is imitating. By comparing the performance of the robots in the two sets, it is determined if the presence of a demonstrator in the environment speeds up the evolutionary process. By comparing the results between different experimental setups, the relation between the degree of difficulty of the task and the learner's tendency to imitate is explored.

The results show that the robots that are evolved with a demonstrator learn the more difficult tasks faster than the robots that are evolved without any demonstrator. The demonstrator did not seem to have any effect on the training time when the robots were evolved to solve the easiest task. In all of the experiments where the robot learned to follow the demonstrator, the robot was able to repeat the task in the demonstrator's absence. The results indicate that evolved robots are able to solve tasks through unsupervised imitation. The results also indicate that evolved robots are less likely to solve a task through unsupervised imitation if the task is too easy.

The analysis of the robots’ behaviours shows that it is easier for the robots to solve the task in the demonstrator’s presence. Even though it is easier to learn the task in the demonstrator’s presence, the robots still have to overcome some of the problems associated with the task. Since the robots learn to solve these problems they have a good chance of performing the task successfully in the demonstrator's absence.

Since it has been shown that evolved robots are able to solve a task through unsupervised imitation of a demonstrator, this training method is worth considering when training a robot to learn a task. The advantage with this approach is that the presence of the demonstrator speeds up the training and the designer does not have to worry about how the robot should be able to imitate the demonstrator. The approach has the limitation that it only works when a demonstrator is available.

7.1 Discussion

As mentioned above, unsupervised imitation can be used when training a robot to learn a task. There has been much research where the goal is not to teach the robot to perform a certain task, but to teach the robot to mimic a demonstrator. It could be possible to use unsupervised imitation to train a robot to mimic a demonstrator. If the task, or the fitness function, frequently changes during the evolution and the demonstrator's behaviour changes accordingly, it would seem likely that robots that are good at repeating the actions of the demonstrator are evolved. This approach has the advantage that the designer does not have to worry about how the robot should learn to mimic the demonstrator. However, it remains to be seen if this is an approach that works in practise.

Breazeal and Scassellati (2002) identify a number of problems that should be solved when constructing imitating robots. Three of them are: How does the robot know when to imitate? How does the robot know what to imitate? How does the robot map observed actions into behavioral responses? None of these problems are trivial. The results in this thesis indicate that evolution is able to solve these problems without any influence from the designer. Unfortunately, the solutions evolution came up with in the experiments in this thesis came in the form of a set of weights for an ANN and it is not easy to analyse the weights to see what the answers to the three questions are.

7.2 Future work

This thesis shows that unsupervised imitation takes place in a given test scenario. It remains to be seen if unsupervised imitation takes place in more complex robots. The robots used in the experiments in this thesis only have two wheels, which limits what kinds of actions they can perform. A robot with a more human-like morphology able to imitate humans would have many practical applications. It is therefore motivated to investigate if unsupervised imitation is possible in such robots. This investigation could use an approach similar to the one used in this thesis.

A drawback when letting robots learn through imitation is that a demonstrator must be present. If the demonstrator is a human it is desirable to minimize the number of times the human has to repeat the demonstration. In the experiments in this thesis the demonstrator was present from the first generation and was only absent in the last 50 generations. In the experiments in chapter five the demonstrator did not seem to have any effect on the fitness during the first 30 generations. Perhaps the demonstrator only has to be present during a few critical generations? This could be tested by performing a set of experiments where the period during which the demonstrator is present is altered.

7.3 Final words

Since it has been shown that evolved robots are able to solve a task through unsupervised imitation, and the relation between the degree of difficulty of the task and the robots' tendency to imitate has been explored, the two aims of this thesis have been achieved. The main conclusion of this thesis is that when evolving robots to solve a task the presence of a demonstrator performing the task can reduce the training time needed to find a solution even if the robots have not been programmed to follow the demonstrator. The thesis contributes to the field of autonomous robotics since unsupervised imitation is an approach that has not been tried before on evolved robots and the approach has the advantage that the robots do not need explicit instructions about when, what and how to imitate.

References

Alissandrakis, A., Nehaniv,C. L., and Dautenhahn, K., 2004 Towards robot cultures? Learning to imitate in a robotic arm test-bed with dissimilarly embodied agents, *Interaction Studies* 5:1, pp. 3-44.

Andry, P., Gaussier, P., Moga, S., Revel, A., and Nadel, J., 2000. Imitation: Learning and Communication. *6th conference of Society for Adaptive Behavior*, SAB2000, pp 354-362.

Baldwin, J. M. 1901. *Dictionary of philosophy and psychology*. Macmillan, New York.

Bakker, P., and Kuniyoshi, Y., 1996. Robot see, robot do: An overview of robot imitation, *AISB96 Workshop: Learning in Robots and Animals*, 1996.

Billard, A. G., 2000. Learning motor skills by imitation: a biologically inspired robotic model. *Cybernetics & Systems*, 32, 1-2, pp 155-193.

Billard, A. G. 2002. Imitation. In M. A. Arbib (ed.), *Handbook of Brain Theory and Neural Networks*, MIT Press.

Breazeal, C., and Scassellati, B., 2002.Challenges in building robots that imitate people, *Imitation in Animals and Artifacts*, Dautenhahn, K., and Nehaniv C., MIT Press.

Dautenhahn, K. and Nehaniv, C., 2002. *Imitation in Animals and Artifacts*. MIT Press, Cambridge, Mass., USA.

Gatsoulis, Y., Maistros, G., Marom, Y and Hayes, G., 2002. Learning to Forage Through Imitation. *Proceedings of the Second IASTED International Conference on Artificial Intelligence and Applications* (AIA2002), Malaga, Spain, September 2002, pp. 485-491.

Grupen, R. A., 2003. A Developmental Organization for Robot Behavior. *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* 101,Boston, MA, USA, pp. 25-36.

Hayes, G., and Demiris, J., 1994. A Robot Controller Using Learning by Imitation in *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, Grenoble, France, July 1994.

Lungarella, M., and Metta, G., 2003. Beyond Gazing, Pointing, and Reaching: A Survey of Developmental Robotics. *Proceedings Third International Workshop on Epigenetic Robotics:*

Modeling Cognitive Development in Robotic Systems 101, Boston, MA, USA. Prince, C.G., Berthouze, L., Kozima, H., Bullock, D., Stojanov, G. and Balkenius, C., pp. 81-89.

Mataric, M., 2000. Getting Humanoids to move and imitate, *IEEE Intelligent systems,* July 2000, pp 18-24.

Meltzoff, A. N. and Moore, M. K., 1983. Newborn infants imitate adult facial gestures. *Child Dev*., 54, 702-9.

Nehmzow, U., 2002. Physically Embedded Genetic Algorithm Learning in Multi-Robot Scenarios: The PEGA algorithm. *Proceedings Second International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* 94, Edinburgh, Scotland. pp. 115-123.

Nolfi, S., and Floreano, D. 2000. *Evolutionary Robotics - The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, Cambridge.

Paquier, W., Huu, N. D., and Chatila, R., 2003. A Unified Model For Developmental Robotics. *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems* 101, Boston, MA, USA, pp. 173-174.

Russel, S., and Norvig, P. 1995. *Artificial Intelligence, a Modern Approach*, Prentice Hall International Editions, New Jersey.

Saunders, J., Nehaniv, C. L., and Dautenhahn, K., 2004. An Experimental Comparison of Imitation Paradigms used in Social Robotics. *Proc. IEEE Ro-man 2004, 13th IEEE International Workshop on Robot and Human Interactive Communication,* September 20-22, 2004 Kurashiki, Okayama Japan, IEEE Press, pp. 691-696.

Schaal, S., 1999. Is Imitation Learning the Route to Humanoid Robots?, *Trends in cognitive science*, Vol. 3, Issue 6, pp 233-242.

Streichert, F., 2002. *Introduction to Evolutionary Algorithms*. Presented at the Frankfurt MathFinance Workshop April 2-4, 2002.

Thorndike, E.L. 1911. *Animal intelligence*, Macmillan, New York.

Tomasello, M., Savage-Rumbaugh, S. and Kruger, A. C., 1993. Imitative learning of actions on objects by children, chimpanzees, and enculturated chimpanzees. *Child Dev*., 64, 1688-705.

Tomasello, M., 1999. *The Cultural Origins of Human Cognition*, Harvard University Press.

Walker, J., and Garret, S., Wilson, M, 2003. Evolving Controllers for Real Robots: A survey of the Literature, *Adaptive Behavior*, Vol. 11, No. 3, Sage Publications, pp 179-203.

Whiten, A., 2000. Primate culture and social learning. *Cognitive Science*, 24, pp 477-508.