

Evolving Internal Memory for T-Maze Tasks in Noisy Environments

DaeEun Kim

Cognitive Robotics

Max Planck Institute for Human Cognitive and Brain Sciences

Amalienstr. 33, Munich, D-80799

Germany

daeeun@cbs.mpg.de

Abstract

In autonomous agent systems, internal memory can be an important element to overcome the limitations of purely reactive agent behaviour. This paper presents an analysis of memory requirements for T-maze tasks well known as the road sign problem. In these tasks a robot agent should make a decision of turning left or right at the T-junction in the approach corridor, depending on a history of perceptions. The robot agent in simulation can sense the light intensity influenced by light lamps placed on the bank of the wall. We apply the evolutionary multiobjective optimization approach to finite state controllers with two objectives, behaviour performance and memory size. Then the internal memory is quantified by counting internal states needed for the T-maze tasks in noisy environments. In particular, we focused on the influence of noise on internal memory and behaviour performance, and it is shown that state machines with variable thresholds can improve the performance with a hysteresis effect to filter out noise. This paper also provides an analysis of noise effect on perceptions and its relevance on performance degradation in state machines.

keywords: T-maze, delayed response task, evolutionary robotics, finite state machines, evolutionary multiobjective optimization, internal memory

1 Introduction

Behaviour-based robotics has been a feasible and attractive approach in mobile robot control systems. It has proved that a reactive coupling between sensor readings and motor actions can be successful in physical world environments. Brooks, who proposed the concept of behaviour-based approach, built multiple primitive behaviours to handle many difficult situations and put the mechanism of action selection over those primitives [Brooks, 1986]. Nowadays the design of intelligent systems, especially in robotics, emphasizes the physical interaction with the environment, that is, more direct coupling of perception to action and dynamic interaction with the environment.

In many robotic tasks, the design of control systems is not a trivial problem. Evolutionary algorithms have been a popular approach to develop desirable control systems which can be adapted to complex environments [Harvey et al., 1992, Nolfi and Floreano, 2000]. The incorporation of state information permits an evolved control structure to behave better, using past information, than a pure reaction to the current sensor inputs. Yamauchi and Beer [1994] showed that dynamical recurrent neural networks could be evolved to integrate temporal sensory perceptions before an agent's decision action. In their experiments, the networks were evolved to process time-varying sonar signals to identify two different landmarks. Also one-dimensional navigation to learn the relationship between a landmark and the goal was achieved with the neural networks. As a result, they suggested the dynamic neural networks should be used to integrate reactive, sequential and learning behaviour. Evolutionary robotic approaches often use recurrent neural networks as controllers for complex behaviours [Harvey et al., 1992, Floreano and Mondada, 1996, Meyer et al., 1998, Nolfi and Floreano, 2000], combined with genetic algorithms to determine the network structure or parameterization. Such networks make effective controllers, offering the possibility for creating many possible dynamical systems with a variety of attractors, but it is hard to analyse them theoretically and to quantify the amount of memory a particular network provides to an agent.

The use of internal memory in agent behaviours has been of interest in evolutionary robotics or adaptive behaviour community. Internal memory can play a role in escaping difficult situations where the information of sensor readings is insufficient to direct motor actions. Situations¹ with the same sensory pattern but requiring different motor actions often occur when sensor readings are restricted. Especially an analytical study of internal states for an agent problem can be found in Bakker and de Jong [2000]'s work. They estimated the complexity of agents and environments with functional and mechanistic states. The mechanistic perspective is concerned with a variety of sensory features that an agent uses, for example, the activation levels of hidden nodes in feedforward neural networks. In contrast, the functional states correspond to internal states or memory that the agent needs to process a history of sensor inputs or perceptual aliasing. In their approach, Elman networks [Elman, 1990] were first trained by reinforcement learning and then finite state automata were extracted from the recurrent neural networks. The mechanistic states and the functional states were counted within an error bound to match the finite state automata and the recurrent neural networks. As

¹They are called perceptual aliasing [Whitehead and Ballard, 1991].

a difficulty measure of a given agent problem, they suggested to count the number of functional states to achieve a certain level of performance.

In fact, the two perspectives, mechanistic states and functional states, are correlated to each other [Kim, 2004], and the number of functional states needed for a given problem may depend on mechanistic states, which has been neglected in Bakker and de Jong’s approach. Moreover, their measure greatly relies on the training performance of recurrent neural networks and there was no effort to find the minimal form of recurrent neural networks in the learning procedure. Instead they used the Hopcroft algorithm [Hopcroft and Ullman, 1979] to minimize the number of functional states on the extracted finite state automata after training neural networks. There has been research of memory analysis in a noise-free grid world environment [Kim and Hallam, 2002], which uses an evolutionary multiobjective optimization with two objectives, memory size and behaviour performance. Generally agent behaviours are not progressed to a large extent in terms of evolutionary fitness values if the amount of memory is over the essential limit. It would be meaningful to find what is the basic memory limit to produce a certain level of behaviour performance.

Finite state machines (FSMs) have been used previously in evolutionary computation to represent state information [Fogel et al., 1966, Ashlock et al., 1995, Miller, 1996, Kim and Hallam, 2002]. They are advantageous for the analysis of memory by their discrete expressions even though they are less powerful in representation than recurrent neural networks. State machines have been applied to robotic tasks [Kim and Hallam, 2001, Kim, 2002] and it was proved that a few internal states improve the behaviour performance for robotic tasks and the internal states are served as connectives of decomposable tasks or a series of actions. FSMs have a feature of identifying each internal state, enumerating memory amount and stacking a sequence of subtasks. In this paper, control structures are based on FSMs to analyse the role of internal memory for a set of T-maze tasks, where necessary internal states will be quantified and identified.

In our experiments, an agent is supposed to make a decision of moving left or right at the T-junction after accumulating a series of sensor activations in the corridor. This T-maze task is one of the road sign problems, which can be classified as a delayed response task [Rylatt and Czarnecki, 2000]. There have been several approaches for the road sign problems, mostly with recurrent neural networks [Ulbricht, 1996, Jakobi, 1997, Rylatt and Czarnecki, 2000, Linåker and Jacobsson, 2001a,b, Bergfeldt and Linåker, 2002, Ziemke and Thieme, 2002]. Ulbricht [1996] started the work of memorizing a time-warped sequence of sensor activations and suggested a state layer placed before the hidden layer of the neural network, where neurons with self-recurrent feedback loops process a history of sensor readings. Later Rylatt and Czarnecki [2000] used Elman network [Elman, 1990] to solve the same task where the activations of hidden layer unit are fed back to a set of context units at the input layer. Bergfeldt and Linåker [2002] showed that FSMs can be extracted with two layers of neural networks for the road sign problem, where the upper layer dynamically modulates the lower layer of reactive mapping. The upper layer, in particular, consists of an unsupervised classifier as an event extractor, memory cells and gating units [Linåker and Jacobsson, 2001a,b]. The finite state automata were extracted from the upper layer configuration for a 3-choice delayed response task. Ziemke and Thieme [2002] showed that a short-

term memory can be realized for delayed response tasks through synaptic plasticity and dynamic modulation of sensorimotor mapping.

So far the research on the road sign problem has focused on the methodology of how to design neural network controllers to solve the problem. The analysis over the memory amount has not been studied in detail. We will thus use an evolutionary multi-objective optimization approach over two objectives, number of memory states (functional states) and behaviour performance. This could reveal the complexity of the road sign problem. Here, our approach is distinguished from the method of extracting state machines from recurrent neural networks. The suggested approach will directly evolve state machines to find the minimal form of control structure for desirable behaviour performance.

In this paper, we will assume that only a limited number of mechanistic states are available to design simplest controller for the road sign problem. It can help to understand the influence of internal states on the behaviour performance; more mechanistic states have a potential of reducing the size of internal memory [Kim, 2004]. FSMs need a threshold value to binarize or partition sensor readings. The performance of state machines may greatly depend on the threshold which builds sensory features. We will investigate how to organize sensorimotor mapping by varying the threshold level and observe the relationship among sensory features, internal states and behaviour performance.

In mobile robots, noise influences the behaviour performance by misleading the perceptions. It is one of challenging works in evolutionary robotics to handle the noise and to extract correct sensory perceptions from noisy signals. Jakobi et al. [1995] studied the effects of simulation noise level in their experiments. According to their results, when the noise levels in simulation are significantly different from the real noise in the real robot, the transfer from the simulated controller to the real world is not effective. They provided some guidelines for constructing simulations that would probably allow successful transfer of evolved behaviour to reality [Jakobi et al., 1995, Jakobi, 1997, 1998]. Miglino et al. [1995] also used the method transferring the best simulation controller to the physical world. They applied statistical analysis of real sensor noise to the simulation and observed that more realistic noise levels can provide more feasible solutions, reducing the gap between simulation and reality. However, It has not been considered how noise influences memory encoding of controllers. We will explore how internal memory handles noise in delayed response tasks with a limited sensor space, or how the noise level influences the performance for a given task.

We first introduce an evolutionary multiobjective optimization approach with FSM controllers. The approach is then applied to a set of T-maze tasks to investigate their memory requirement and to estimate the complexity of the delayed response tasks. Single-light / two-light experiments in T-maze environment and their results are demonstrated. FSM controllers are evolved in noisy environments, and they are tested with various levels of noise to see the robustness of evolved controllers and the impact of noise on internal memory. To handle noise appropriately, a method of varying thresholds on sensors in FSMs is provided. Finally we present an analysis of noise effect on perceptions to explain why noise is involved with performance degradation.

2 Methods

2.1 Robot configurations

In this paper, robotic tasks are investigated with simulation involving noisy signals in sensor readings and motor actions, and we take a Khepera robot model [Mondada et al., 1993] for simulation. There are ambient light sensors at the same angle positions as infrared sensors. The light sensors can detect light intensity in a given environment. In the simulation experiments, only two light sensors at angle 22° and 158° (the front direction 90°) are used. The light intensity is estimated with two factors, the distance of the light sensor from a lamp, and the lamp's angle deviated from the centre-line of the sensor. The motor commands for both left and right wheels are restricted to the set $\{-8, -6, -4, -2, 2, 4, 6, 8\}$ and only eight possible actions are allowed for each wheel. These motor commands will be used for turning left or turning right to reach the goal position. For T-maze task, a robot is forced to move forward in the approach corridor until it reaches the T-junction. The wheel dynamics are subjected to random noise. A random noise of $\pm 10\%$ is added to the motor speeds for the two wheels, and the direction of the robot is influenced by $\pm 5\%$ random noise. Infrared sensor readings ranges from 0 to 1023 and a random noise of 10% is added to the sensor values. In evolutionary experiments, the sensor readings will be binarized with a threshold of 500 for state machines. Ambient light sensors have values from 0 to 500 (0 means the highest light intensity). At the initial experiments a threshold of 250 will be used for light sensors (we select this threshold by the middle point of the sensor range) and then varying thresholds will be tested.

2.2 Finite state machines

A Finite State Machine (FSM) can be considered as a type of Mealy machine model [Kohavi, 1970, Hopcroft and Ullman, 1979], so it is defined as $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ where q_0 is an initial state, Q is a finite set of states, Σ is a finite set of input values, Δ is a set of multi-valued output values, δ is a state transition function from $Q \times \Sigma$ to Q , and λ is a mapping from $Q \times \Sigma$ to Δ , where $\lambda(q, a)$ will be a member of the output set Δ . $\delta(q, a)$ is defined as the next state for each state q and input value a , and the output action of machine M for the input sequence $a_1, a_2, a_3, \dots, a_n$ is $\lambda(q_0, a_1), \lambda(q_1, a_2), \lambda(q_2, a_3), \dots, \lambda(q_{n-1}, a_n)$, where $q_0, q_1, q_2, \dots, q_n$ is the sequence of states such that $\delta(q_k, a_{k+1}) = q_{k+1}$ for $k = 0, \dots, n-1$. The finite state machine we consider is encoded for the evolutionary algorithm as a sequence of pairs (state number, state output) on each sensor value in canonical order of state number; one chromosome will be represented as an one-dimensional array of genes.

[Insert figure 1 about here]

A chromosome representation including state q_0, q_1, \dots, q_r (total $r + 1$ states) is given in figure 1, where for state q_i , d_i^k is the motor action, q_i^k is the next state for the k -th sensor configuration and τ_{ij} is the threshold for the j -th sensor to binarize the corresponding sensor readings; for a special case, $\tau_{i1} = \tau_{i2} = \dots = \tau_{in}$. For n

sensors, $m = 2^n$ sensor configurations are available. For the experiments with a fixed threshold on light sensors, evolving thresholds will be omitted. For state machines, the chromosome size is dependent upon the number of states, since a series of the state transition and motor action will be repeated as many as the number of states. Each FSM has a pre-defined number of internal states. In the genetic pool of evolutionary computation, variable length chromosomes are allowed for varying sizes of states in FSMs.

2.3 Evolutionary multiobjective optimization

To determine how many memory elements are required to solve a particular agent-environment interaction problem, evolutionary computation will try to optimize agent performance for each level of memory amount. By doing so, the trade-off between performance and quantity of memory can be explored. If sensors are discretized, a controller with internal memory can easily be expressed as a finite state machine. The amount on memory needed for a given task will here be defined as the number of states used to achieve a desirable behaviour score. This approach follows estimating the complexity of a given task by the memory states [Bakker and de Jong, 2000, Kim and Hallam, 2002].

The method to see the behaviour performance for each level of memory amount is to use the Evolutionary Multiobjective Optimization (EMO) [Goldberg, 1989, Fonseca and Fleming, 1993, Zitzler, 1999, Coello et al., 2002]. The EMO approach has been applied in quantifying internal memory required for the Woods problem which is one of goal-search problems in a grid world [Kim and Hallam, 2002]. In this paper, two objectives, behaviour performance and memory size, are used in the Pareto optimization to try to maximize behaviour performance and minimize number of controller states. The shape of the Pareto surface after an evolutionary run indicates a desirable number of memory elements for a given performance level. As a result, one can often determine a threshold amount of memory needed to achieve a task. We assume that the quantity of memory in the control structure needed for a given task represents the complexity of the problem faced by the agent.

The Pareto scoring in EMO approach has been popular [Goldberg, 1989, Fonseca and Fleming, 1993] and it can maintain a diverse population over multi-objectives. A dominance rank is defined to specify the rank of members in the Pareto distribution.

A vector $X = (x_1, x_2, \dots, x_m)$ for m objectives is said to *dominate* $Y = (y_1, y_2, \dots, y_m)$ (written as $X \prec Y$) if and only if X is partially less than Y , that is,

$$(\forall i \in 1, \dots, m, x_i \leq y_i) \wedge (\exists i \in 1, \dots, m, x_i < y_i).$$

A *Pareto optimal set* is said to be the set of vectors that are not dominated by any other vector.

$$\{X = (x_1, \dots, x_m) | \neg(\exists Y = (y_1, \dots, y_m), Y \prec X)\}$$

To obtain a Pareto optimal set, a dominating rank method² [Fonseca and Fleming, 1993] is applied to the tournament selection of group size four in this paper. The

²The dominating rank method defines the rank of a given vector in a Pareto distribution as the number of elements dominating the vector. The highest rank is zero, for an element which has no dominators.

tournament selection initially partitions the whole population into multiple groups for the fitness comparison. Inside the tournament group, Pareto score of each member is compared each other and ranked. A higher rank of genomes in the group have more probability of reproducing themselves for the next generation. In our approach, a population is initialized with random length chromosomes by variable state machines. The two best chromosomes are selected using a dominating rank method over the two objectives, memory size and behaviour performance. They reproduce themselves only with mutation and the two worst chromosomes are replaced by new offspring produced using one point crossover followed by mutation on the two best genomes. In the application of variable state machines, offspring with variable numbers of states should be generated to keep diversity in the genetic pool. Thus, a size modifying genetic operator is introduced to maintain variable length coding. When offspring are produced, the number of memory states³ is randomly pre-selected for each new offspring. The chromosome size for each offspring will depend on this chosen amount of memory. The size-modifying operators are used to produce new offspring such that they have characteristics of their parents and have the pre-chosen chromosome length. It concatenates several copies of the parent chromosomes and then cuts the string at the pre-chosen length [Kim and Hallam, 2002]. After applying the size-modifying operator, crossover is applied to the two offspring. The crossover point is selected inside both chromosome strings after aligning the prefixes of two strings. During this crossover process, mutation is used to change one integer value in the strings into a random number. The size-modifying operator is applied to 75% of new offspring, in experiments with variable state machines. When it is not used, the offspring keep the size of their parents. New offspring are thus produced with a size modifying operator, crossover and mutation by turns. Crossover rate is 0.6, and mutation rate is set to two over chromosome length in the experiments.

The multi-objectives in the EMO consist of behaviour fitness and the number of memory states. The single EMO run produces a Pareto-optimal set in which all members have a rank zero. The set corresponds to the best behaviour performance for each level of memory amount. It will easily display what is the desirable number of memory elements for a given behavior performance. The method can relatively decrease the computation time required to take multi-experiments of independent runs for each size of memory elements.

2.4 Sensory configurations

[Insert figure 2 about here]

The memory amount needed for an agent problem may depend on sensory configurations. In this paper, we will assume only a limited number of sensor features are

³Memory size is an objective in our EMO approach, and the range of memory states is finite and small so that we can enumerate all the possible values. The memory size is related to the structure of genomes, and so the diversity of genomes is maintained by randomly selecting memory size for a new genome. It has an effect of searching for the whole range in the objective of memory size.

available. If each sensor has its own threshold τ_i , the parameter space is divided into two non-overlapping subsets, depending on τ_i ; it has a dichotomy $S_i > \tau_i$ and $S_i \leq \tau_i$ where S_i is the i -th sensor value. For T-maze tasks, two light sensors are used and so the effective sensor space will have four subdivisions. They reflect four sensory configurations (mechanistic states), irrespective of the threshold value. We will explore how a variation of sensory features can influence the memory requirement for a fixed number of mechanistic states. Figure 2 shows possible sensory feature structures depending on thresholds. For some T-maze task, we will evolve three types of controllers shown in figure 2 to check the effect of sensory feature structures. This study will show the importance of the quality of sensory features for a fixed number of sensory configurations.

3 Experiments

We test a set of T-maze tasks as the road sign problem. The T-maze decision behaviour will require long short-term memory to remember the past perceptions. An agent will pass through a narrow corridor and must decide its directional move, left or right, depending on its perception experience when it sees the T-junction in front. This T-maze experiment has been tested with robots by Jakobi [1997]. In his experiments with one lamp, robots maintained internal states with recurrent neural networks to remember which side the lamp was on such that they could make a correct decision at the junction. In our experiments, we will identify internal states of controllers in noisy environments and also several light positions will be tested with one or two light lamps.

For the T-maze task, random controllers will have 50% probability for each direction, left and right. Desirable controllers will, for their choice of actions, depend on perceptual cues or landmarks that have been experienced while they are moving forward. Robots in the T-maze will use light intensity instead of complex landmarks. This will demonstrate an example of how robots can organize their memory from their perceptions.

Instead of allowing free movement, robots are forced to move forward until they reach the T-junction; this is unlike Jakobi [1997]’s work where robot controllers were evolved to go forward to the junction before the decision of turning. Two ambient light sensors at angle 22° , 158° and two infrared sensors at angles 45° , 135° on a Khepera robot will be used in the experiments (the front direction is 90°), where the infrared sensors are supposed to detect obstacles or walls in the front (the sensor readings are used to recognize the T-junction). A threshold of light intensity is assumed to binarize their intensity. The intensity values of ambient light sensors are influenced with random noise. Later we will see the decision performance depending on the noise level or varying thresholds. While robots move forward from starting positions to the T-junction, their light sensors continue to change and binarized signals will flip from 0 to 1 or from 1 to 0, depending on how close to light robots are.

In evolutionary experiments, the penalty fitness function is defined as follows:

$$f_T = 1 - \frac{1}{\kappa \cdot m} \sum_{j=1}^{\kappa} \sum_{i=1}^m E(d_{ij}, g_{ij}) \quad (1)$$

where d_{ij} is the direction that robots choose at the T-junction, g_{ij} is the direction for goal position, κ is the number of environments to test robots, and m is the number of experimental runs. Expression $E(a, b)$ is the equality function to be set to 1 if the directions of a, b are the same and otherwise 0. The fitness evaluation is influenced by noise and so the evolutionary algorithm will prevent accidental success or failure of controllers by evaluating one robot controller several times (m times). The success rate of the T-maze decision can be estimated by $1 - f_T$.

As the average performance over samples is considered, each chromosome in a population is evaluated five times to estimate the fitness in equation (1) ($m = 5$). The best chromosome of the population is evaluated again 25 times with $m = 25$ in equation (1) (more runs will produce a better estimate of the prediction success rate for a chromosome) and the fitness score of the best chromosome will be monitored. The EMO method is applied to see significance statistics results on variable state machines for T-maze tasks in noisy environments. The EMO evolutionary run will be repeated 25 times.

3.1 Single-Light Experiments

[Insert figure 3 about here]

First, a simple task with single light was tested. In the experiments, light is placed on the left or right block bank. Light can be placed in the middle of each bank as shown in figure 3; robots do not detect light initially, but when they move forward, they can sense the light as they approach it. When they move away, they cannot sense the light again. When robots are close enough to the T-junction, they decide which direction to move in. If the light is placed on the left side, robots should move left at the T-junction. The goal position is at the end of corridor on the left side. If light is placed on the right side, then robots should move right to reach the goal on the right side. They detect walls with infrared sensors to find the moment of turning to the left or right. The task will test if robots can do conditional movement depending on one position of the light. In fact, this experiment has been studied in the road sign problem [Ulbricht, 1996, Jakobi, 1997, Rylatt and Czarnecki, 2000]

[Insert figure 4 about here]

The EMO approach is applied for memory analysis of the single light T-maze with a population size of 50 and 100 generations. The result⁴ shows that the task requires at least two internal states as displayed in figure 4. The experiment was run 25 times and its average performance is estimated with 95% confidence intervals by assuming t -distribution. Purely reactive controllers have 50% success rate and they only choose one direction in any condition of perceptions. When standing at the T-junction, purely

⁴The Pareto set in evolutionary runs has two objectives, the behaviour fitness f_T and the number of states, but for convenience, the success rate ($1 - f_T$) vs. states will be displayed in the paper.

reactive robots do not see any light and they cannot consider any previous light condition. They can just choose to turn left or right regardless of light that has been seen in the corridor path. The best controllers with at least two states have internal states to be triggered by left light lamp or right light lamp. For example, a two-state controller has its internal state for turning right by default and ignores light on the right side but if it sees light on the left side, it changes its internal state into the mode for turning left as shown in figure 5. When the robot reaches the T-junction, the turning move will depend on its internal state.

[Insert figure 5 about here]

However, the task can be purely reactive if robots can move freely in the corridor [Jakobi, 1997, Bergfeldt and Linåker, 2002]. When they see light on the left side, they can begin to turn left and follow left walls until they reach the goal position. Otherwise, they follow right walls. Robots determine their movement in advance before they cross the T-junction at the instant that they can see light. Thus, it can be easily achieved with purely reactive controllers. When robots are forced to move forward in spite of any perception, they internalise perceptual states using available memory states. The internal states will determine which move they should choose at the T-junction. We evolved controllers with various noise levels ranging from 0 to ± 40 for this task, and we obtained controllers with the perfect performance irrespective of noise level. For this single light task, noise did not decline the behaviour performance or the convergence speed to desirable controllers. Figure 5 shows a set of state machines evolved in a noise-free environment with a success rate of 0.5, 1.0, 1.0 and 1.0, respectively. At this point, we checked if each evolved controller can survive the noisy environment. State machines in figure 5 were evaluated 500 times ($m = 500$ in equation (1)) with noise level ± 30 , and they showed the success rate 0.5, 1.0, 0.8785, and 1.0, respectively. The three-state machine has a significant drop of performance in noisy environments (a higher level of noise induces worse performance), even though it showed a perfect score with 500 evaluations in a noise-free environment. It is caused by the fact that the evolved noise-free environment provides no information about a state transition to be prevented and the FSM experiences an inappropriate state transition by noisy sensor readings. The three-state machine in figure 5 has a stable state q_0 for the sensory event⁵ 01, when a light appears on the right side. After moving away from light, the robot detects no light and the sensory event is switched to 00. It triggers a state transition from q_0 to q_2 and the agent keeps the stable state to make a decision of turning right. This situation, however, can be distorted by noise. Noise often changes the sensory event 01 into event 00 or event 00 into event 01, when the light intensity is close to the threshold level for FSMs. The agent may experience state transitions from q_0 to q_2 and then q_2 to q_1 , or a cyclic transition from q_0 to q_2 , q_1 , q_0 in sequence. The state transition can accidentally stop at the state q_1 at the moment that no light is sensed any more. It will lead to turning left at the T-junction, which is a false movement. In contrast, the

⁵We will represent the sensor activation as a binary value, binary 1 means a high activation of the sensor. For instance, the sensory feature event 01 indicates that the left light sensor has a low light intensity and the right sensor value has high, that is, the agent detects light on the right side.

two-state machine and the four-state machine in figure 5 shows a stable state transition for any noise level.

[Insert figure 6 about here]

To test the robustness of the evolved controllers (by 25 trials), each best controller is evaluated 500 times with a variety of noise levels ranging from 0 to ± 40 . Then the average performance of 25 best controllers and its error bar are calculated. Figure 6 shows how much the performance of the evolved controllers changes in test environments. The controllers evolved with a low noise level often experience degradation of performance. Testing them with higher noise levels on light sensors produces worse performance. The reason is that transient error signals caused by noise can severely influence state transition in state machines. More internal states tend to produce a larger drop of performance. It implies that FSMs with a large number of internal states can be sensitive to noise and structurally unstable for this case. As shown in figure 6(b), controllers evolved with a high level of noise are more robust. This test experiment says that noise can significantly defect evolved state transitions. Evolving controllers in noisy environments with a high level of noise can reduce the possibility of ill-posed state transition by noise. We will explore more about the noise effect in other T-maze tasks.

3.2 Two-Light Experiments

[Insert figure 7 about here]

More complex situations can be made with two lights. Four possible cases with two lights are created to distinguish environmental situations, as displayed in figure 7; both lights can be placed on the left side or on the right, or one light is placed in the lower area on the one side and another light is in the upper area on the other side. Figure 7 also shows examples of noisy light intensity signals. If light is close to an ambient light sensor, its sensor readings becomes very low (ambient light sensors on a Khepera robot have low sensor readings for high light intensity). The initial experiments use a fixed threshold of 250 to binarize the intensity signals.

[Insert table 1 about here]

We design three possible tests over four cases with two lights as in table 1; they will be called `tmaze-env1`, `tmaze-env2`, `tmaze-env3`. Each experiment will be tested to see how many memory states are required to make correct decisions for moving to the goal position. The EMO approach is applied to each experiment for memory analysis with a population size of 50. The above experiments will evolve controllers which are attentive to light signals. For the setting `tmaze-env1`, goal positions depend on lights in the upper area and robots should be able to detect lights near the T-junction by ignoring lights near starting positions. Direction of turning left or right

will rely on which side light is placed in the upper area. In the environmental setting `tmaze-env2`, goal positions will depend on lights in the lower area. Robots should recognize the direction of lights near the starting position and they should decide their turning actions using their first perception of light. For the setting `tmaze-env3`, the formation of light should be memorized to decide the move. It will be a more complex task since the perceptions of lights should be sequenced and the detection of the second light after the first light is required to make appropriate actions.

3.2.1 `tmaze-env1` task

[Insert figure 8 about here]

[Insert figure 9 about here]

[Insert figure 10 about here]

The `tmaze-env1` setting has two lights but the environmental situations require only two memory states. One of the best controllers, as shown in figure 8(a), changes its internal state depending on light sensation; while passing light on the left side, robots mark their internal state in advance so that they will decide on turning left and when they reach the T-junction, they move left by following their internal state. If light is seen on the right side, robots change their internal state to move right later. While the robot moves forward in the corridor, it may see light on the alternate side. The robot then toggles its internal state into the state of light direction. In this way two memory states, one state for each direction, are sufficient to predict goal positions correctly. The EMO approach result in figure 9(a) also shows that two states are required to achieve the task. Figure 8(b) is a 5-state FSM evolved in a noise-free environment. The FSM showed a success rate of 1.0, 1.0, 0.997, 0.94, 0.873 and 0.8355 over 500 evaluations for a noise level of 0, ± 5 , ± 10 , ± 20 , ± 30 , ± 40 , respectively. A higher level of noise is spawning worse performance. Transient error signal on perceptions by noise tends to change the intended internal state into irrelevant states. For example, a sequence of light perception in a noisy environment, $(00)^*(10)^*(00)(10)(00)^*$, encounters a state transition from q_0 to q_1, q_2, q_4 , and then from q_4 to q_3, q_0 in order. It finally ends at the state q_0 corresponding to the mode of turning right, which is a false decision, although this type of state transition cannot occur in a noise-free environment. For the T-maze task `tmaze-env1`, we tested various noise levels for controllers evolved with a noise level of ± 5 and ± 20 . Similar to the result in the single-light experiments, we can see that more states tend to be sensitive to noise and evolving controllers with a high level of noise helps to find robust controllers – see figure 10.

3.2.2 `tmaze-env2` task

[Insert figure 11 about here]

Memory analysis in figure 9(b) for `tmaze-env2`, which is similar to the test environment by Ziemke and Thieme [2002], says that it requires at least three memory states. One of the best control strategies is to set the robot’s internal state into the mode of turning left or right at initial perception of light. It will ignore any light configuration after the robot decides its turning action in advance. In this environment, two-state controllers, as well as purely reactive controllers cannot be successful in every case. Desirable controllers need to disambiguate the situations where light is near the T-junction from those where it is near the starting position on the same side. As a result, it will require one more memory state than the `tmaze-env1` experiment. Figure 11 shows examples of state machines evolved in a noise-free environment. The two-state machine on the left side shows a success rate of 0.75 over 500 evaluations with a noise level of ± 30 , and another two-state machine on the right has a success rate of 0.626. The former machine correctly predicts the goal position for case 1, case 2 and case 3 in table 1(b) and it always fails for case 4. The latter machine correctly predicts the goal position only for case 4, but takes a random decision over the other three cases by alternate switch between two internal states, which shows a relatively low performance in noisy environments (theoretically the prediction rate can be calculated as $(1/4) + (3/4) \cdot (1/2) = 5/8$, which is close to our experimental result). This FSM was often evolved in a noise-free environment and it seems that it recorded the time-span of light perception on the left side. However, this machine was never found among a set of controllers evolved in noisy environments, for instance, with a noise level of ± 30 . The 3-state machine in figure 11 showed a perfect prediction rate for any noise level.

[Insert figure 12 about here]

The FSM controllers in the EMO approach were evolved 25 times with noise level ± 10 , ± 20 , ± 30 , ± 40 , respectively, and then each of the best evolved controllers was tested over 500 evaluations. The results are shown in figure 12. It is observed that controllers evolved with a high level of noise, ± 30 or ± 40 are very robust, while a noise level of ± 10 or ± 20 produces a significant drop of performance in the test with a higher level of noise.

3.2.3 `tmaze-env3` task

In the `tmaze-env3` experiment, the decision of turning cannot be simply determined after passing the corridor. This task is more complex than the previous two-light experiments, since robots should recognize the formation of lights. When noise is involved with sensor readings, the perceptual states may become unstable, which will render the task more difficult. Thus, it is investigated whether or not noise in sensor readings can affect the memory requirement for the given task. Experiments with and without noise in sensor readings are tested with several memory states in the problem `tmaze-env3` (but noise still remains on infrared sensors and motor actions for any case). Each evolutionary experiment is repeated 25 times with a population size of 50.

[Insert figure 13 about here]

Figure 13 is the result of EMO application to the environment `tmaze-env3`. It shows the performance hierarchy of memory states. Without noise, the performances of two, three and four states are significantly different – see figure 13(a). For four states or more, the fitness is stabilized. Various noise levels for ambient light sensors ranging from ± 5 to ± 40 were tested. Purely reactive controllers are significantly worse than two- or three-state controllers for every case. Sensor noise makes it a harder problem and the hierarchy of behaviour performance is a little changed. Four or more states are remarkably better than two- or three-state controllers in the experiments with noise levels ranging from 0 to ± 20 . From the figures, the T-maze problem for `tmaze-env3` needs at least four memory states. Even though there is an optimal solution with four states, more capacious memory has a tendency to improve solutions rapidly. Purely reactive controllers are only able to choose one direction. Its decision success rate was 0.5. It is almost the same as random decision in performance.

[Insert figure 14 about here]

One of the best evolved controllers predicted the decision with 100% success rate among 500 evaluations. This controller is shown in figure 14 and noise does not severely influence its state transition. Two states (q_1, q_2) represent the state of detecting light on the left and on the right. If robots detect light on the same side, they keep their internal state. They change their state into a new state (q_3) if a light is observed on the other side. In our experiments, noise often changes the perception of light, that is, varies the perception event by Hamming distance 1, for instance, 10 into 00, 01 into 00, 00 into 01, or 00 into 10. Figure 14 shows how to handle this misleading perception and each state keeps its state invariant to noise. Thus, we presume that the task can be accomplished with at least four memory states, regardless of any noise in light sensor readings. It was observed that controllers with more than four states can also handle transient errors in perceptions. Their control strategies change their internal state into a different internal state temporarily when noise creates transient error signals in light intensity, and when the signal becomes stable, the previous state is restored.

[Insert table 2 about here]

To see the hierarchy of behaviour performance, a success is defined as the case in which the success rate after 500 generations is greater than 0.90, 0.95 or 0.99. Table 2 shows the number of success cases among 25 trial runs with population size 50. Controllers with less than four states have no success at all and so the record in the table again confirms that at least four states are required to achieve the T-maze task. A high level of noise, ± 30 or ± 40 , significantly degrades the performance and makes the task more difficult. We presume that this difficulty of evolution is caused by the fitness function and its deceptive property depending on noise level. A higher level of noise seems to produce more local optima in the fitness landscape.

[Insert figure 15 about here]

We also check the robustness of evolved controllers by providing different levels of sensor noise. Each best controller is evaluated 500 times with a variety of noise levels. Figure 15 shows how much the performance of the evolved controllers changes in test environments. The controllers evolved with a low noise level on light sensors often experience the performance decline in the test with a higher noise level. If the evolved controllers are tested in the environment with a lower level of noise, there was no degradation in performance but rather a slight improvement although it is not statistically significant. The decline occurs only when they are tested with a higher level of noise. It partly supports Jakobi’s argument that the noise level in simulation should match that in reality for successful transfer of the evolved controllers [Jakobi et al., 1995, Jakobi, 1997]. The noise level reflects the distortion level of the original sensor values and it influences the performance. This will be explained in more detail in the analysis of noise effect in section 4. In the robustness test, more than two states tend to experience a prominent drop of the performance, and the degradation level is stabilized for four states or more, with any noise level evolution.

For the next set of experiments, we introduce variable thresholds for sensor configurations as shown in figure 2. First, state machines with one varying threshold (ranging from 50 to 450 for light sensors) for each state are evolved. Two light sensors share one threshold for each internal state, and each state has its own threshold to binarize the light intensity. Second, we test evolving an independent threshold for each sensor.

[Insert figure 16 about here]

Figure 16 shows that variable thresholds (sharing one threshold for each state) in state machines can significantly improve the decision performance even for a high level of noise. The EMO approach proved three states are sufficient for state machines with variable thresholds. State machines with a fixed threshold of 250 required four internal states, but variable thresholds has the effect of reducing the number of internal states needed for the T-maze task `tmaze-env3`. Moreover, the former controllers did not succeed in achieving high success rate with noise level ± 30 and ± 40 as shown in figure 13(e)-(f). For finite state machines with variable thresholds, the evolved controllers showed similar performance regardless of the noise level. The variable thresholds effectively process the sensor noise and so help more robust controllers evolved.

[Insert figure 17 about here]

Figure 17 displays one of the best controllers evolved with variable thresholds. At the initial state, the agent monitors light sensor values and checks if both sensor values are inactive, that is, larger than a threshold of 350. Once no light is perceived on either side, it lowers its threshold level into 150 to prevent state transition caused by any noise. Thus, the agent needs a perception of high light intensity to change its internal state. This process can be seen as a kind of hysteresis phenomenon to block transient error by random noise. If we have a fixed threshold value, the noisy sensor values around the

threshold will toggle the internal state frequently. It thus delivers a negative effect of misleading the perceptions and as an alternative, it requires another redundant internal state to handle such noise.

[Insert figure 18 about here]

[Insert figure 19 about here]

To see the robustness, evolved state machines with variable thresholds were tested in a variety of noise level. Figure 18-19 show the test results after 1000 generations. Their evolving and test performances are mostly better than those of evolving controllers with a fixed threshold of 250. Also the performance degradation by noise in the test experiments is rather smaller, because they find effective thresholds. Here evolving controllers with noise level ± 30 or more may lead to different styles of controllers, because both left and right light sensors together can have high intensity values due to a high level of noise. That is, state transitions can be evolved with the sensory event 11 which is not observed with a low noise level. Thus, the pattern of the graphs is changed in test experiments with noise level ± 30 . Allowing two varying thresholds for each state does not improve the performance significantly, but more clearly shows that three states are sufficient for the T-maze task.

4 Analysis of Noise Effect on Perceptions

We have shown that noise has a negative effect of misleading perceptions and so produces a bad performance on memory-encoding controllers. Now we analyse why a higher level of noise tends to produce worse performance in test experiments.

[Insert figure 20 about here]

The sensory event over one light sensor has a binary value and the event value flips by noise when the sensor readings surpasses the threshold or when it becomes lower than the threshold. We assume the sensor readings on light sensors is linearly increasing or decreasing when the robot moves away from or towards light. For increasing sensor readings, a negative effect of noise occurs when one sensor value is over the threshold at a given time step and the next sensor reading is below the threshold (see figure 20), that is, $s_n + \eta_n > T$ and $s_{n+1} + \eta_{n+1} < T$ where T is the threshold and s_i is the noise-free sensor value and η_i the noise at time i with $\eta_i \in [-\rho, \rho]$. Also the event flipping can happen when $s_n + \eta_n < T$ and $s_{n+1} + \eta_{n+1} > T$ for decreasing sensor readings. We will only consider the first case, since the other case is the same as the former case and only differs in the inequality sign. If we take the equation

$$s_n + \eta_n > T, \quad s_{n+1} + \eta_{n+1} < T$$

, then $s_n + \eta_n - T + T - s_{n+1} - \eta_{n+1} > 0$. We can write it as follows:

$$\eta_n - \eta_{n+1} > s_{n+1} - s_n = \delta_n$$

where δ_n is the change rate of sensor readings for one time step. η_n, η_{n+1} are random noise $\pm\rho$ ranging from $-\rho$ to ρ in which ρ is the maximal noise in the range. Then the flipping probability can be calculated as the shaded region over the whole squared area as displayed in figure 20. Hence, the flipping probability p is estimated as

$$p = \frac{(2\rho - \delta)^2}{2 \cdot 4\rho^2} = \frac{(2\rho - \delta)^2}{8\rho^2} = \left(\frac{1}{\sqrt{2}} - \frac{\delta}{\sqrt{8\rho}}\right)^2.$$

According to the above equation, the flipping probability depending on noise level is shown in figure 21(a). If the noise level ρ increases, then the flipping probability p also increases for a fixed change rate of sensor readings, δ . If δ is larger than twice the maximal noise level, then the noise flipping can never happen. It implies that a slow robot movement has more potential of being sensitive to noise, and its internal state may experience an undesirable state transition. Similarly we can estimate the flipping probability for Gaussian noise as shown in figure 21(b), which is achieved with numerical analysis.

[Insert figure 21 about here]

In the experiments, solutions evolved with a high level of noise improve the performance when they are tested with lower noise levels. If the noise level decreases, the flipping probability also reduces and the memory encoding with state machines will less often experience an inappropriate state transition when we assume that the evolved state machines are not a perfect solution to handle noise. It thus improves the performance in the test with lower noise levels.

[Insert figure 22 about here]

Now we show how state machines with variable thresholds can overcome this noise flipping. At initial state, the sensory event is set to a and a threshold T_0 is given a higher value than the normal threshold as shown in figure 22. If the sensor readings reaches the first threshold T_0 , then the internal state is changed to hold a new sensory event b , and the threshold value is newly switched into T_1 , smaller than the normal. For any noise within the range $T_0 - T_1$, the state machines keep the internal state and the sensory event b constant. Only when the sensor readings are below the new threshold T_1 , they can return to the sensory event a and the previous internal state is restored. The diagram of the state/event change is similar to the hysteresis effect often observed in magnetic materials. It is also equivalent to the Schmitt trigger operation in electronic circuit; the Schmitt trigger application is a comparator to stabilize switches from the noisy input voltage, and it can prevent rapid triggering by noise, using two reference voltages. The above hysteresis performance will largely depend on the two thresholds, T_0 and T_1 . In this paper, our experiments successfully evolved FSM controllers with the hysteresis effect, and the structure processes noise as in the above analysis. Schmitt trigger can be simply achieved with variable thresholds in FSMs.

5 Discussion

We applied the finite state machines to identify memory states needed for the T-maze problem. The approach used axis-parallel sensor configurations with a dichotomy of sensor space for each sensor, while complex sensory configurations are available with a combination of sensor values and the weight parameters in neural networks. Two types of controllers, FSMs with a fixed threshold and FSMs with variable thresholds, showed that T-maze tasks can be achieved with several memory states and simple sensory configurations. We suggest that evolving FSMs can be a useful tool to design simple controllers and to understand the operation of internal control memory for a given task. The above T-maze tasks can be extended to more complex situations, for example, decision with more light lamps or finding the goal by counting the number of lights. It would require more memory amount to handle such problems.

Neural network approaches have been popular to handle the delayed response tasks. Here the agent-environment complexity issues were exploited for the tasks. By following Bakker and de Jong [2000]’s criterion on the agent complexity based on functional states, internal memory was quantified in the FSMs for a set of T-maze tasks. In neural networks, sensory configurations are complex and it is difficult to analyse and measure the memory amount needed for the tasks, even though there is an approach to extract FSMs from neural networks [Omlin and Giles, 1996]. In our approach, the FSMs are directly evolved to quantify the memory amount without the extraction process from recurrent neural networks. It can produce a simpler design of FSM controllers by handling the parameter of memory states easily. The results of extraction from neural networks may be different from those of evolving directly the FSM to see the necessary memory amount. We leave the comparison of the two approaches for future work.

Some approaches that tackled the road sign problem have used the filtering process by unsupervised learning [Linåker and Jacobsson, 2001a,b] or by a combination of neuron weights and sensor values [Ziemke and Thieme, 2002]. It is presumed that the methods may have misleading perceptions by noise unless they employ short-term memory. State transitions in evolved FSMs can be seen as a conceptual abstraction of internal memory as Ulbricht [1996] showed in her work. In our approach, the sensory feature can be obtained directly by dividing sensor space with thresholds. State transitions include two parts, a part of recording internal memory and a part of handling sensory features. The experiments with the hysteresis effect showed that a limited sensor space needs states, which should be disambiguated from internal states of processing the abstraction of memory. We argue that the sensory feature needs short-term memory to process noise. Linåker and Jacobsson [2001a] showed that unsupervised classifiers process sensory features or events in an early stage and then the events are handled with internal nodes including feedback loops, which correspond to internal memory. In their approach, the extraction of sensory features requires no short-term memory. In contrast, the state machines with variable thresholds use short-term memory for the hysteresis effect to process sensory features, and the memory is not separated from internal memory of controlling a sequence of events. As a result, it reduces the number of memory states needed for the T-maze task.

In this paper, we do not intend to imply that finite state machines can produce better solutions than recurrent neural networks or that finite state machines have a better fea-

ture of memory encoding. The state machines with a fixed threshold over sensor space have shortcomings of being sensitive to noise, even though it is essential to filter out noise for a delayed response task. However, the FSM approach helps to understand the role of internal memory to achieve a given T-maze task. The experiments also prove that an appropriate choice of threshold is significant to handle noisy sensor readings. FSMs with variable thresholds improve the performance and this implies that the quality of sensory configurations, when their number is fixed, can play an important role of evolving desirable controllers.

According to Jakobi’s argument, the noise in simulation should be close to that in the reality. However, it is noted that a noise level higher than in reality can also assist successful transfer. In our experiments, controllers evolved with a high noise level improve the performance in the test with lower noise levels. Also there is a limit of noise level to preserve the original environmental situation; too low or too high levels of noise may evolve controllers adapted to slightly different environments. It would be an interesting question what is the limit of noise level for the successful transfer of evolved controllers.

6 Conclusion

The T-maze problem, as a small domain of the road sign problem, needs long short-term memory to recognize the environmental situations before decisions of turning left or right. In this paper the evolutionary multiobjective optimization (EMO) approach was applied for an analysis of memory requirement for T-maze tasks. The EMO approach produced significance statistics on the performance difference of variable state machines in noisy environments. Purely reactive controllers could not succeed in predicting correct decisions from perceptual experience in the T-maze problem, and multi-states were required to memorize perceptual states. For T-maze tasks, several experiments were tested with lights acting as environmental landmarks. The intensity of light signals fluctuates depending on the distance between the sensors and the lights. When one light was used for the direction of future movements, two internal states were required. When two lights were placed at four possible positions and the turning decision relied on the formation of lights, a different size of memory states from two to four was required. If more lights are used at various positions, more memory states would be required to record various environmental situations.

We also studied the noise effect on perceptions. A high level of noise has a potential of distorting sensory perceptions and triggering inappropriate state transitions in FSM controllers, which can lead to false T-maze decisions. A higher level of noise leads to a high probability of false movements. When evolved controllers are tested with a lower level of noise, the performance can be improved under the condition that a history of sensory perceptions in the evolved environment is not so different from that in the test environment. We need further study to find what ranges of noise can improve the performance. The experiments showed that more memory states tend to be sensitive to noise and have a potential of degrading the performance in test environments, but the degradation level is stabilized after some number of internal states.

Allowing variable thresholds in state machines can effectively extract sensory fea-

tures and exploit the memory states. It may need a smaller number of internal states for the state machines to achieve correct T-maze decision and as well it can design more robust controllers. We believe that variable thresholds can discover more accurate or prominent sensory features in a noisy environment by using a hysteresis effect to avoid misleading sensory perceptions.

7 Acknowledgements

The author would like to thank Dr. Tom Ziemke and two anonymous reviewers for their valuable suggestions and comments, and also Dr. John Hallam for his introduction of Pareto scoring and hysteresis phenomenon.

References

- D. Ashlock, M.D. Smucker, and A.A. Stanley. Iterated prisoner's dilemma game with choice and refusal of partners. In *Advances in Artificial Life : European Conference on Artificial Life*, 1995.
- B. Bakker and M. de Jong. The epsilon state count. In *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behaviour*, pages 51–60. MIT Press, 2000.
- N. Bergfeldt and F. Linåker. Self-organized modulation of a neural robot controller. In *Proceedings of the International Joint Conference on Neural Networks*, pages 495–500, 2002.
- R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Vol. RA-2, No.1:14–23, 1986.
- C. Coello, D. van Veldhuizen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.
- J. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.
- D. Floreano and F. Mondada. Evolution of homing navigation in a real robot. *IEEE Transactions on Systems, Man and Cybernetics*, 26(3):396–407, 1996.
- L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial intelligence through simulated evolution*. Wiley, New York, 1966.
- C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. Morgan Kaufmann.
- D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, MA, 1989.

- I. Harvey, P. Husbands, and D. Cliff. Issues in evolutionary robotics. In *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behaviour*, pages 364–373. MIT Press, 1992.
- J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison Wesley, Reading, MA, 1979.
- N. Jakobi. Evolutionary robotics and the radical envelope of noise hyperthesis. *Adaptive Behavior*, 6, 1:131–174, 1997.
- N. Jakobi. *Minimal Simulations for Evolutionary Robotics*. Ph. D. dissertation, School of Cognitive and Computing Sciences, University of Sussex, 1998.
- N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Life: Proceedings of the Third International Conference on Artificial Life*, Moran, F. Moreno, A., Merelo, J. Chacon, P. (eds.). Springer-Verlag, 1995.
- D. Kim. *A Quantitative Approach to the Analysis of Memory Requirements for Autonomous Agent Behaviours using Evolutionary Computation*. Ph. D. dissertation, University of Edinburgh, 2002.
- D. Kim. Analyzing sensor states and internal states in the Tartarus problem with tree state machines. In *Parallel Problem Solving From Nature 8*. Springer-Verlag, 2004.
- D. Kim and J. Hallam. Mobile robot control based on boolean logic with internal memory. In *Advances in Artificial Life, Lecture Notes in Computer Science vol. 2159, Proceedings of European Conference on Artificial Life*, pages 529–538. Springer Verlag, 2001.
- D. Kim and J. Hallam. An evolutionary approach to quantify internal states needed for the Woods problem. In *From Animals to Animats 7, Proceedings of the International Conference on the Simulation of Adaptive Behavior*, pages 312–322. MIT Press, 2002.
- Z. Kohavi. *Switching and Finite Automata Theory*. McGraw-Hill, London, 1970.
- F. Linåker and H. Jacobsson. Learning delayed response tasks through unsupervised event extraction. *International Journal of Computational Intelligence and Applications*, 1(4):413–426, 2001a.
- F. Linåker and H. Jacobsson. Mobile robot learning of delayed response tasks through event extraction: A solution to the road sign problem and beyond. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 777–782, 2001b.
- J. A. Meyer, P. Husbands, and I. Harvey. Evolutionary robotics: A survey of applications and problems. In *First European Workshop on Evolutionary Robotics*, Paris, France, 1998. Springer.

- O. Miglino, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial Life*, 2:417–434, 1995.
- J. H. Miller. The coevolution of automata in the repeated prisoner’s dilemma. *Journal of Economics Behavior and Organization*, 29(1):87–112, 1996.
- F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturisation: A tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Experimental Robotics*. Kyoto, Japan, 1993.
- S. Nolfi and D. Floreano. *Evolutionary Robotics : The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.
- C.W. Omlin and C.L. Giles. Extraction of rules from discrete-time recurrent neural networks. *Neural Networks*, 9(1):41–52, 1996.
- R.M. Rylatt and C.A. Czarnecki. Embedding connectionist autonomous agents in time: The ‘road sign problem’. *Neural Processing Letters*, 12:145–158, 2000.
- C. Ulbricht. Handling time-warped sequences with neural networks. In *From Animals to Animats 4*, pages 180–189. MIT Press, 1996.
- S. D. Whitehead and D. H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83, 1991.
- B. Yamauchi and R. Beer. Integrating reactive, sequential, and learning behavior using dynamical neural networks. In *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour*, pages 382–391. MIT Press, 1994.
- T. Ziemke and M. Thieme. Neuromodulation of reactive sensorimotor mappings as a short-term memory mechanism in delayed response tasks. *Adaptive Behavior*, 10 (3/4):185–199, 2002.
- E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Ph. D. dissertation, Swiss Federal Institute of Technology, 1999.

τ_{01}	τ_{02}	...	τ_{0n}	q_0^1	d_0^1	q_0^2	d_0^2	q_0^3	d_0^3	...	q_0^m	d_0^m
τ_{11}	τ_{12}	...	τ_{1n}	q_1^1	d_1^1	q_1^2	d_1^2	q_1^3	d_1^3	...	q_1^m	d_1^m
...
τ_{r1}	τ_{r2}	...	τ_{rn}	q_r^1	d_r^1	q_r^2	d_r^2	q_r^3	d_r^3	...	q_r^m	d_r^m

Figure 1: Chromosome representation with integer encoding ($r + 1$ states are available)

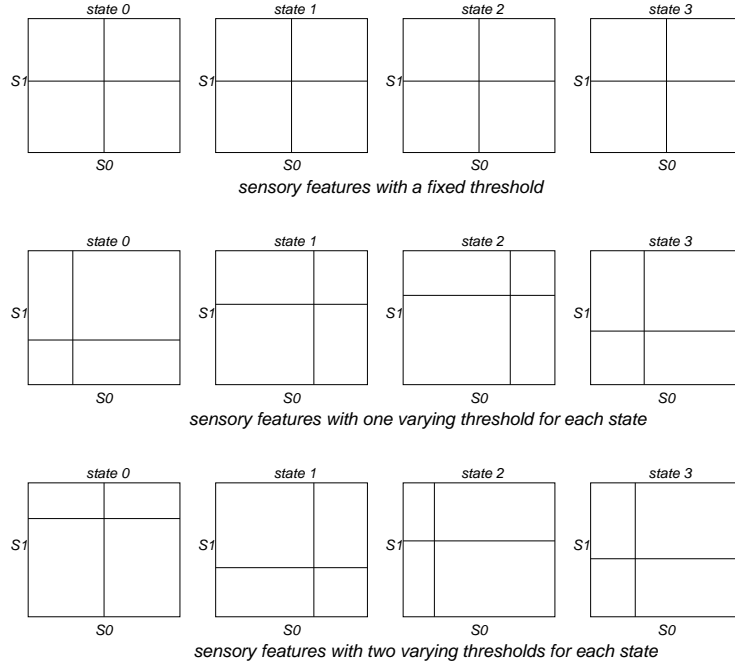


Figure 2: Diagram of possible sensory feature structures with four internal states

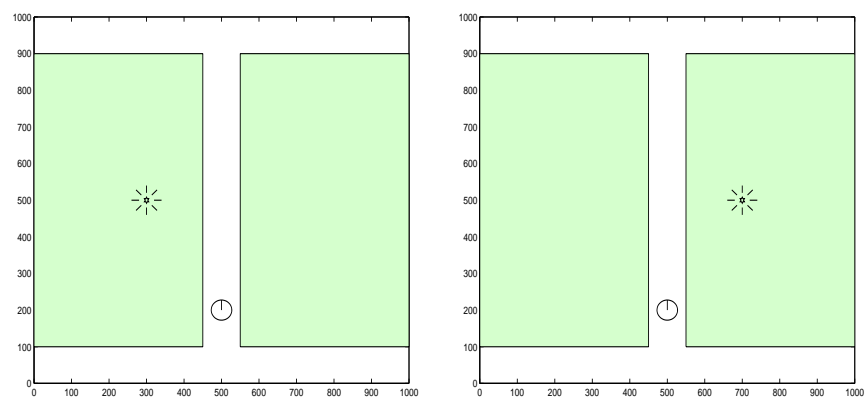


Figure 3: T-maze environment 1

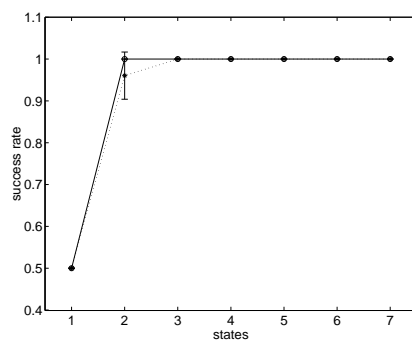


Figure 4: EMO result over the single light T-maze experiment with a noise level ± 10 (dotted: 50 generations, solid: 100 generations)

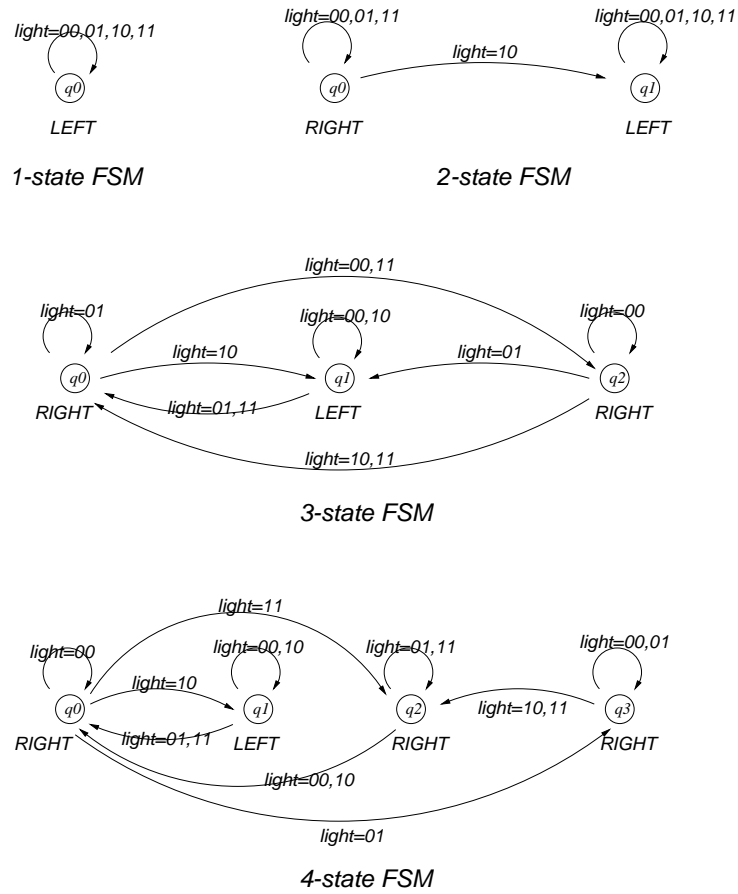


Figure 5: Examples of evolved state machines for a single light T-maze

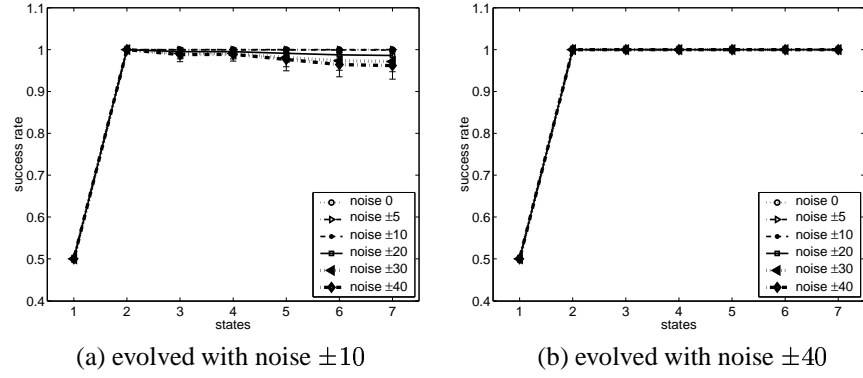


Figure 6: Test experiments of evolved FSMs over single light T-maze problem (tested with different noise levels on ambient light sensors)

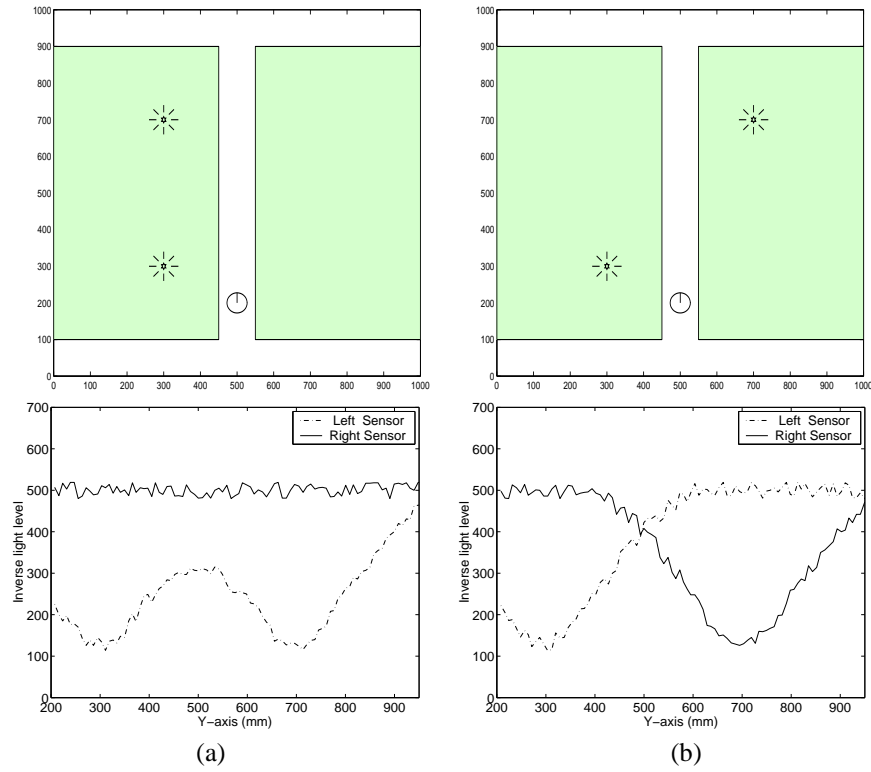


Figure 7: Examples of T-maze environment `tmaze-env3` with light sensor noise level ± 20 (higher intensity has a lower light level) (a) case 1 (b) case 3

cases	goal position	light position
case 1	left	both lights are on the left
case 2	left	one light in the upper left and the other in the lower right
case 3	right	both lights are on the right
case 4	right	one light in the lower left and the other in the upper right

(a) `tmaze-env1`

cases	goal position	light position
case 1	left	both lights are on the left
case 2	left	one light in the lower left and the other in the upper right
case 3	right	both lights are on the right
case 4	right	one light in the upper left and the other in the lower right

(b) `tmaze-env2`

cases	goal position	light position
case 1	left	both lights are on the left
case 2	left	both lights are on the right
case 3	right	one light in the lower left and the other in the upper right
case 4	right	one light in the upper left and the other in the lower right

(c) `tmaze-env3`

Table 1: Experiments with two lights

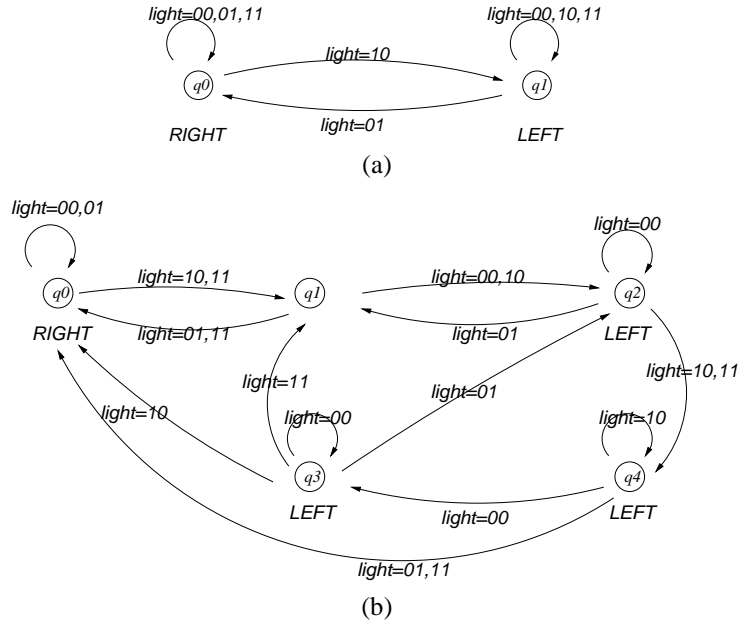


Figure 8: Examples of state machines for T-maze `tmaze-env1` (a) 2-state FSM (b) 5-state FSM

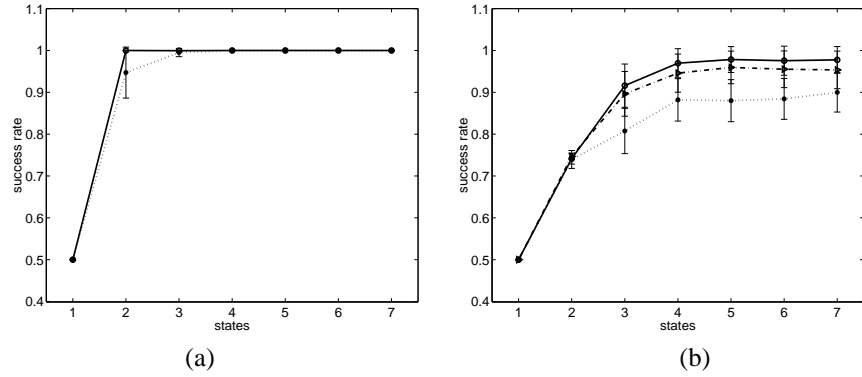


Figure 9: EMO result over T-maze problem `tmaze-env1` and `tmaze-env2` with a noise level ± 10 (a) `tmaze-env1` (dotted: 50 generations, solid: 100 generations) (b) `tmaze-env2` (dotted: 100 generations, dashdot: 200 generations, solid: 300 generations)

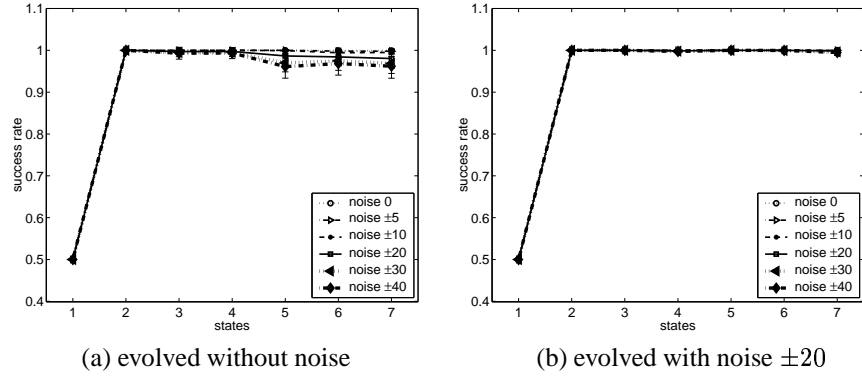


Figure 10: Test experiments of evolved FSMs over T-maze problem `tmaze-env1` with different noise levels on ambient light sensors

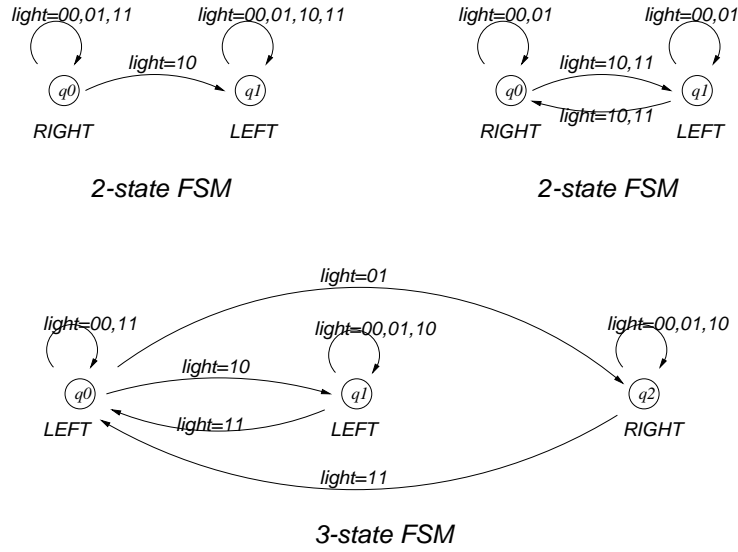
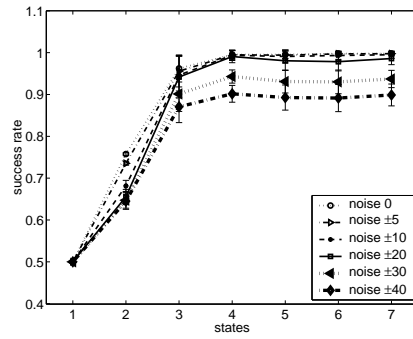
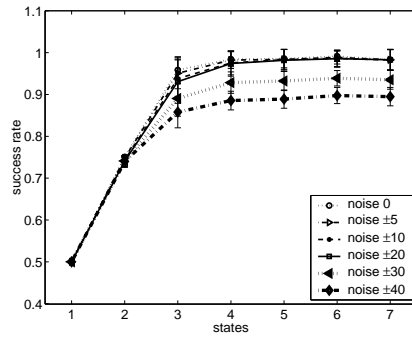


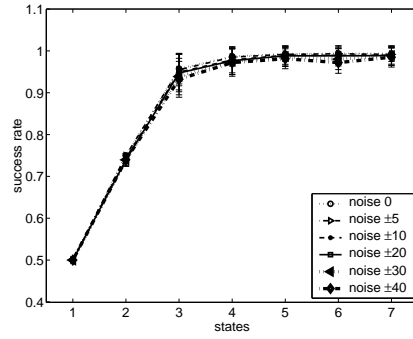
Figure 11: Examples of evolved state machines for T-maze `tmaze-env2`



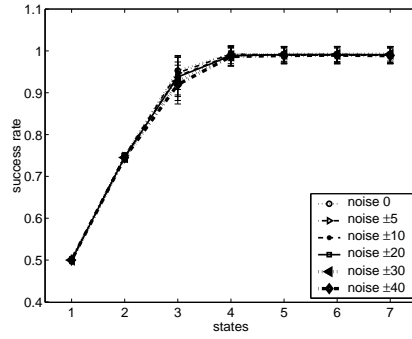
(a) evolved with noise ± 10



(b) evolved with noise ± 20

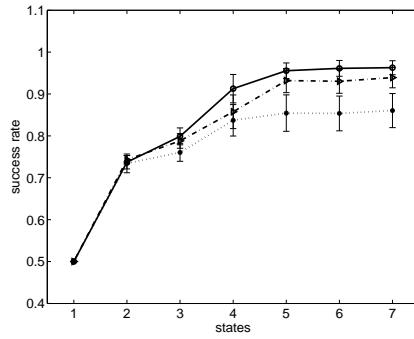


(c) evolved with noise ± 30

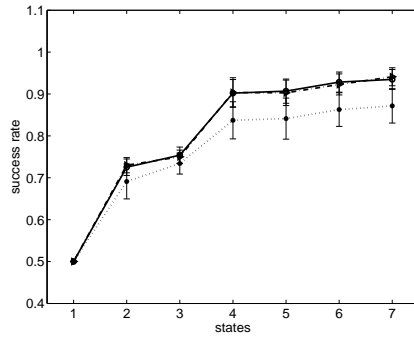


(d) evolved with noise ± 40

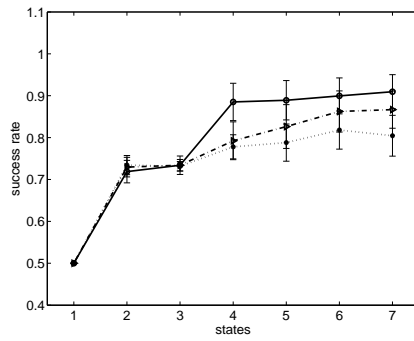
Figure 12: Test experiments of evolved FSMs over T-maze problem `tmaze-env2` (tested with different noise levels on ambient light sensors)



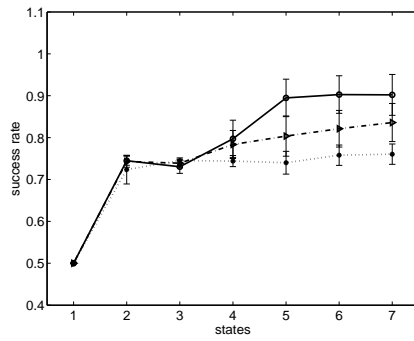
(a) evolved without noise



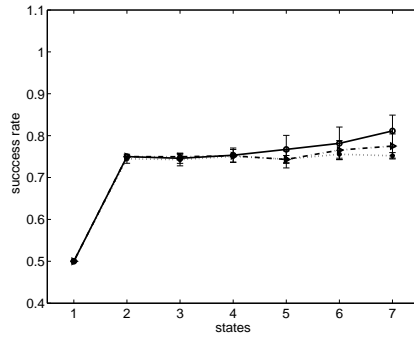
(b) evolved with noise ± 5



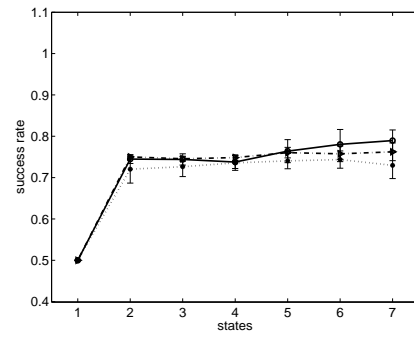
(c) evolved with noise ± 10



(d) evolved with noise ± 20



(e) evolved with noise ± 30



(f) evolved with noise ± 40

Figure 13: EMO result over T-maze problem `tmaze-env3` with various noise levels on ambient light sensors (dotted: 100 generations, dashdot: 200 generations, solid: 500 generations)

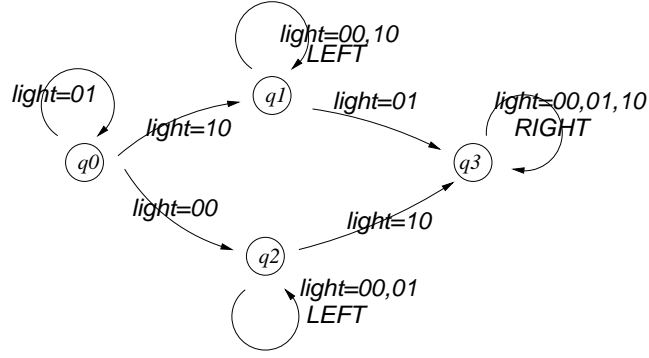


Figure 14: State transition diagram with four states for `tmaze-env3` (sensory configurations involving infrared sensors are not displayed)

parameters		memory states						
noise level	success rate	1 state	2 states	3 states	4 states	5 states	6 states	7 states
noise ± 0	0.90	0	0	0	18	22	22	23
	0.95	0	0	0	9	13	15	16
	0.99	0	0	0	7	9	10	10
noise ± 5	0.90	0	0	0	11	15	12	18
	0.95	0	0	0	8	8	8	9
	0.99	0	0	0	8	6	8	9
noise ± 10	0.90	0	0	0	12	13	16	14
	0.95	0	0	0	11	13	13	13
	0.99	0	0	0	7	11	10	9
noise ± 20	0.90	0	0	0	7	15	15	16
	0.95	0	0	0	2	11	14	14
	0.99	0	0	0	1	5	8	8
noise ± 30	0.90	0	0	0	1	2	4	4
	0.95	0	0	0	1	2	3	3
	0.99	0	0	0	0	0	1	1
noise ± 40	0.90	0	0	0	0	2	3	2
	0.95	0	0	0	0	0	2	1
	0.99	0	0	0	0	0	0	0

Table 2: Number of successes for various memory states in T-maze problem `tmaze-env3` (each column represents the number of successes among 25 trials for a given success rate and the success rate is measured by the fitness value ($m = 25$); there is no success at all for less than 4 states with any noise level)

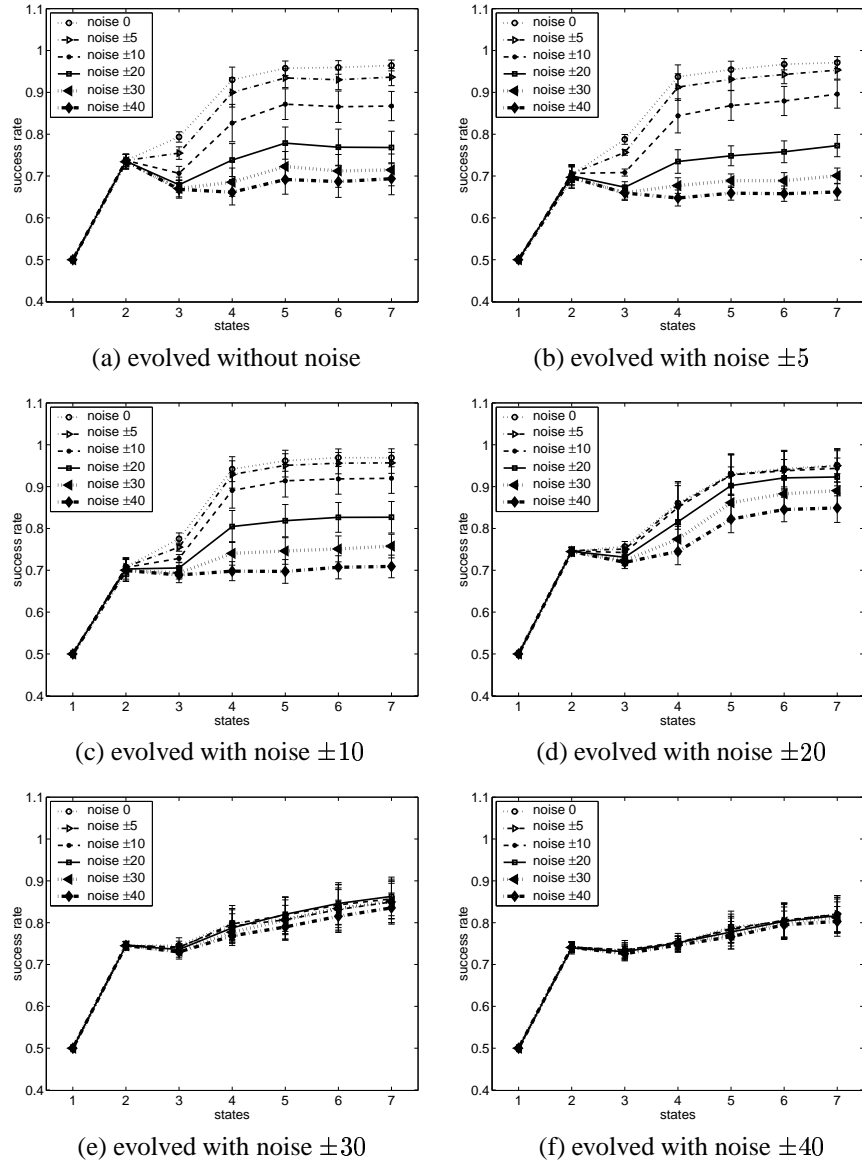


Figure 15: Test experiment of FSMs evolved with a fixed threshold of 250 (25 trial evolutionary runs produce 25 best controllers; each of the best evolved controllers is evaluated 500 times and its average score is calculated)

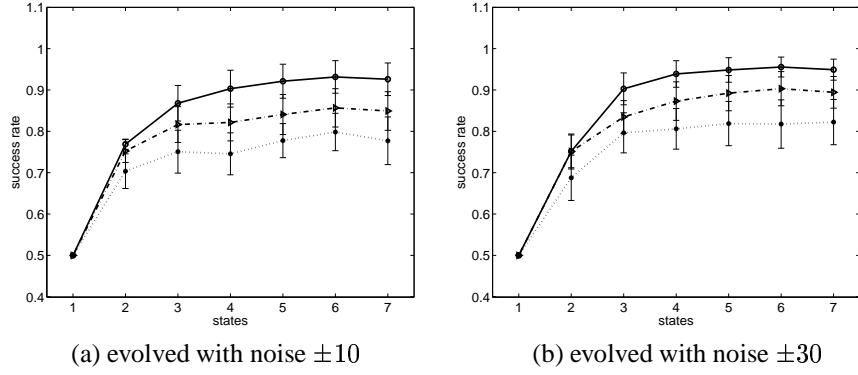


Figure 16: EMO result over T-maze problem `tmaze-env3` with one varying threshold for each state (dotted: 100 generations, dashdot: 200 generations, solid: 500 generations)

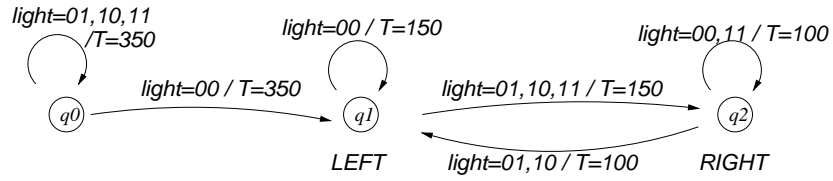
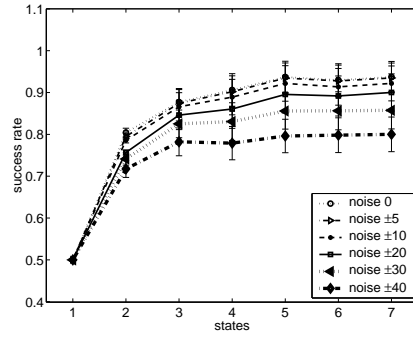
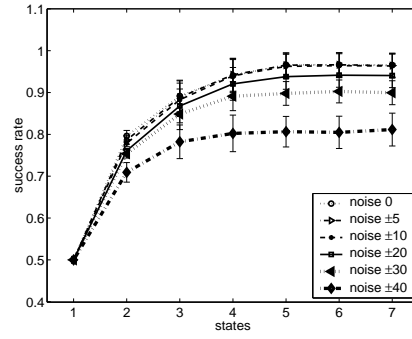


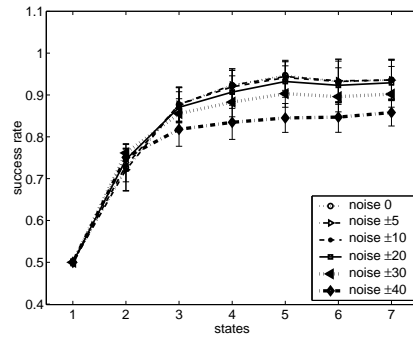
Figure 17: An example of state machines with variable thresholds for `tmaze-env3` (T: threshold value)



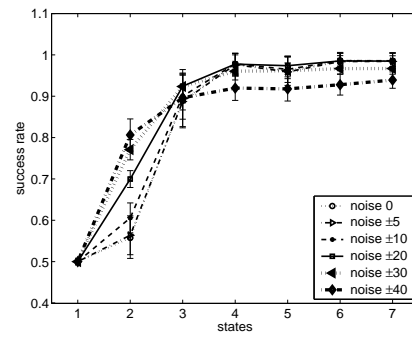
(a) evolved with noise ± 5



(b) evolved with noise ± 10

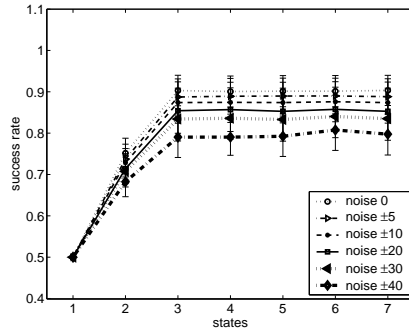


(c) evolved with noise ± 20

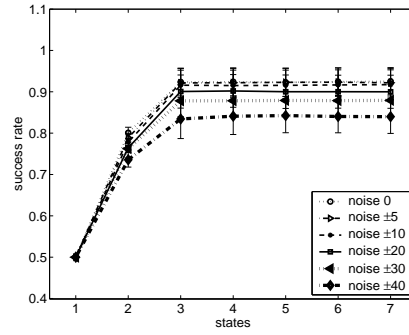


(d) evolved with noise ± 30

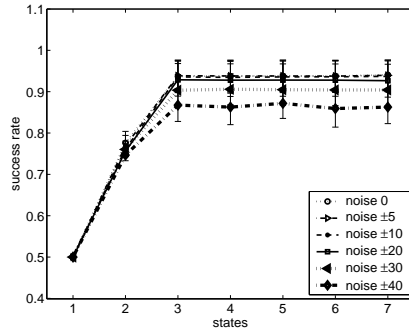
Figure 18: Test experiments of FSMs with one varying threshold for each state over T-maze problem `tmaze-env3` (tested with different noise levels on ambient light sensors)



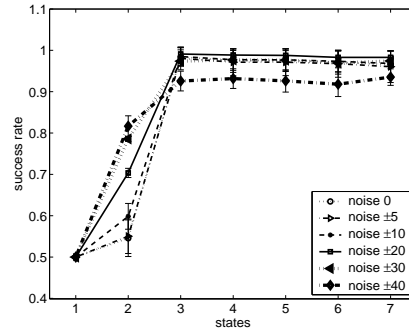
(a) evolved with noise ± 5



(b) evolved with noise ± 10



(c) evolved with noise ± 20



(d) evolved with noise ± 30

Figure 19: Test experiments of FSMs with two varying thresholds for each state over T-maze problem `tmaze-env3` (tested with different noise levels on ambient light sensors)

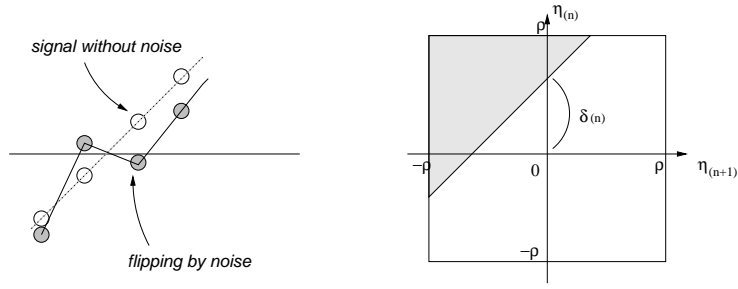


Figure 20: A picture of undesirable event flipping and the noise flipping zone

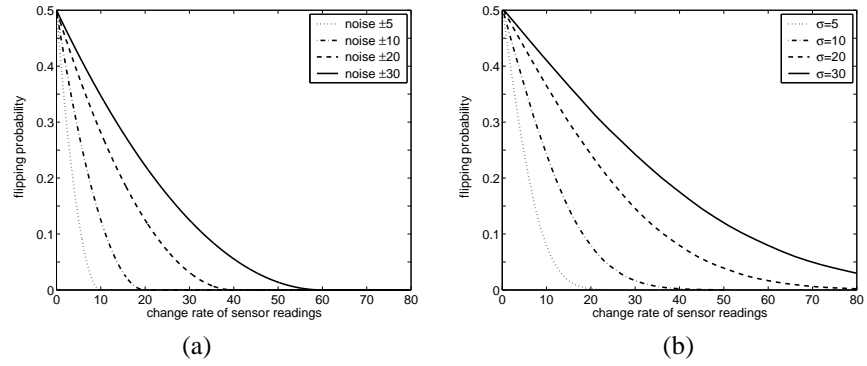


Figure 21: Flipping probability near the sensor threshold (a) random noise (b) Gaussian noise (mean 0, variance σ^2)

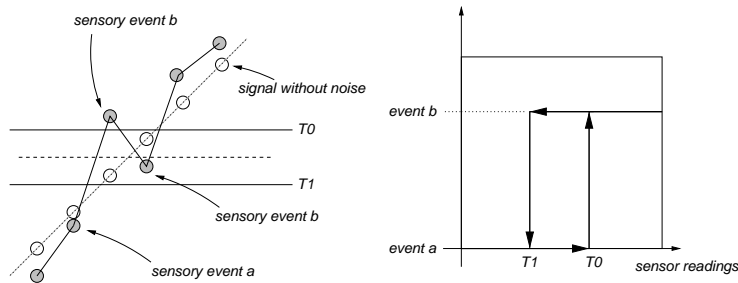


Figure 22: Decision of sensory event with hysteresis effect to handle noise