Exploring the T-Maze: Evolving Learning-Like Robot Behaviors using CTRNNs

Jesper Blynel and Dario Floreano

Autonomous Systems Lab
Institute of Systems Engineering
Swiss Federal Institute of Technology (EPFL)
CH-1015, Lausanne, Switzerland
{Jesper.Blynel, Dario.Floreano}@epfl.ch

Abstract. This paper explores the capabilities of continuous time recurrent neural networks (CTRNNs) to display reinforcement learning-like abilities on a set of T-Maze and double T-Maze navigation tasks, where the robot has to locate and "remember" the position of a reward-zone. The "learning" comes about without modifications of synapse strengths, but simply from internal network dynamics, as proposed by [12]. Neural controllers are evolved in simulation and in the simple case evaluated on a real robot. The evolved controllers are analyzed and the results obtained are discussed.

1 Introduction

Learning in neural networks is normally thought of as modifications of synaptic strengths by for example back-propagation or Hebbian learning. This view was in 1994 challenged by Yamauchi and Beer in [12], where the authors described the abilities of fixed synapse continuous time recurrent neural networks (CTRNNs) to display reinforcement learning-like properties by exploiting internal network dynamics. The task studied was generation and learning of short bit sequences. In [11] this work was extended to an artificial agent task where the relationship between the positions of a goal and a landmark in an environment had to be learned. However, the movement of the agent was restricted and it was equipped with artificial high level goal- and landmark-detection sensors. These restrictions were loosened in the recent work by [10] where an extended version the same landmark navigation task was studied. In the present work we apply a similar approach in which a simulated Khepera robot has to navigate in first a simple and then a double T-Maze. The task for the robot is to locate and "remember" the location of a reward-zone in the environment it happens to be evaluated in. In contrast to the above mentioned work the evolved behaviors are verified by testing them on a real robot in a real environment. Previous work on T-Maze navigation in evolutionary robotics includes delayed response tasks where the robots had to perform one or several turns in a maze on the basis of light source cues given to the robot [5][13]. In contrast to these works our focus is how to retain information over successive trials in the same environment. This becomes possible by equipping the robot with a sensor to detect the position of the reward-zone used for fitness evaluation.[1]

A different line of research has studied how agents in a self-organized ways can learn internal models of the environment [9]. The authors successfully trained a hierarchy of recurrent neural networks to predict increasingly complex information about the environment. The high level information which emerged was in which of two rooms the agent was currently navigating. The authors argued that the model learned could later on be used to generate action plans for goal seeking behaviors as in [8]. In the present work no explicit model of the environment exists, but is tightly coupled with both the learning of behaviors and the generation of motor actions. This corresponds with our belief that, as pointed out by [12], a direct distinction between mechanisms responsible for behavior and mechanisms responsible for learning is hard to defend biologically.

2 Neural Architecture and Genetic Encoding

Continuous-time recurrent neural networks (CTRNNs) are utilized for the experiments in this paper. The state of each neuron can be described by the following differential equation:

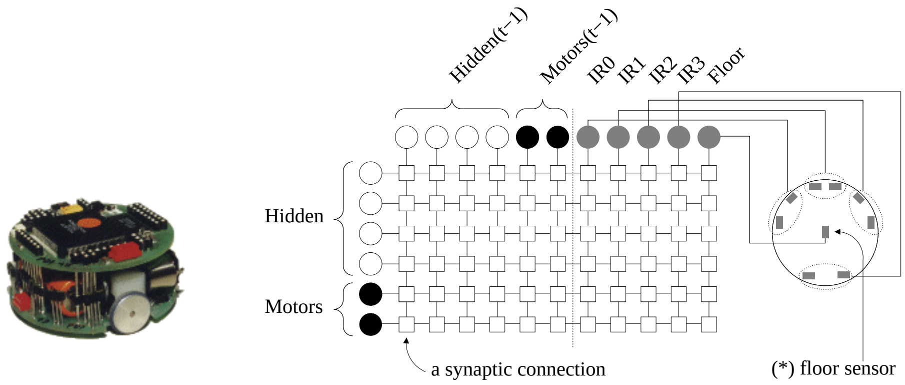
$$\frac{d\gamma_i}{dt} = \frac{1}{\tau_i}\Big(-\gamma_i + \sum_{j=1}^{N} w_{ij} A_j + \sum_{k=1}^{S} w_{ik} I_k \Big) \tag{1}$$

where N is the number of neurons, i $(= 1, 2, ..., N)$ is the index, γ_i describes the neuron state (cell potential), τ_i is the time constant, w_{ij} the strength of the synapse from the presynaptic neuron j to the postsynaptic neuron i, $A_j = \sigma(\gamma_j - \theta_j)$ is the activation of the presynaptic neuron where $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic function and θ_j is a bias term. Finally, S is the number of sensory receptors, w_{ik} is the strength of the synapse from the presynaptic sensory receptor k to the postsynaptic neuron i and I_k is the activation of the sensory receptor ($I_k \in [0, 1]$). As in [12] the *Forward Euler* numerical integration method with step size $\Delta t = 1$ is applied to equation (1). The range of the rest of the parameters are the following:

$$\tau \in [1, 50],\;\; \theta \in [-1, 1] \;\; and \;\; w \in [-5, 5]$$

The network architecture is shown in figure 1(b). The network consists of 6 fully interconnected neurons (4 hidden + 2 motor outputs) and 5 sensory receptors. Every neuron has synaptic connections from all neurons and all sensory receptors. The receptors are configured as follows: 4 inputs from the infrared proximity sensors paired two-by-two and 1 additional input from a floor sensor pointing downwards measuring the surface brightness. The 4 proximity values

[1] Note that in contrast to traditional reinforcement learning no direct reward is given to the robot. The evolved robots has to discover themselves the relationship between the input of this sensor and the fitness score.



(a) Khepera Robot.

(b) Neural Network Architecture.

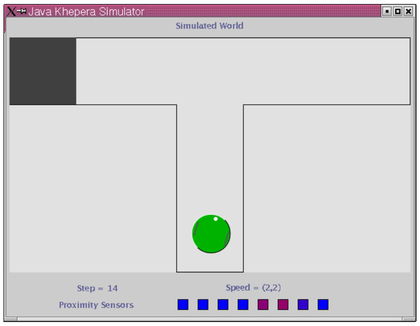
Fig. 1. The *Khepera robot* (a) used in the experiments. A standard Khepera has 8 infrared sensors distributed around the body used for object proximity and light intensity measurement. An extra infrared sensor pointing downwards (*) has been added in the center of the robot body in order to measure the surface brightness below the robot. *Neural Architecture* (b): The network consists of 6 fully interconnected neurons (4 hidden + 2 motor outputs) and 5 sensory receptors. Every neuron has synaptic connections from all neurons and all sensory receptors. The *sensory receptors* are wired to the robot as shown (b, right) (motor connections not shown for clarity).

are scaled between 0 and 1. The floor sensor input is set to 1 if the robot is inside a black reward-zone and 0 otherwise. The activation of the 2 output neurons, linearly scaled between -10 and 10, are used to set the wheel-speeds of the robot.

The network parameters are encoded in a bitstring genotype. Each neuron has 13 encoded parameters: A time constant (τ), a bias threshold (θ), and 11 synaptic strengths (w_{ij}). Each of the 78 network parameters is encoded linearly within its range using 5 bits, resulting in a total genotype length of 390 bits.

3 Experiment 1: Simple T-Maze

In the first experiment a robot has to navigate a simple T-maze (fig. 2). The experiment is carried out in a realistic simulation of the Khepera robot (fig. 1(a)) based on sensor sampling [6] and adding 5% uniform noise to the sampled values. Initially the robot is positioned as shown in figure 2, and the task is to find and stay on the black reward-zone which can be positioned in either the left or the right arm of the maze. The position of the reward-zone stays fixed during each epoch. The robot is tested for 4 epochs of 5 trials each - two epochs with the reward-zone in each arm of the maze. The neural network controlling the robot is initialized (by setting the state of each neuron to zero) at the beginning



Step = 14

Speed = (2,2)

Fig. 2. The simple T-maze environment used in the first experiments. The reward zone (black square) can be positioned either the left arm (shown above) or in the right arm of the maze.

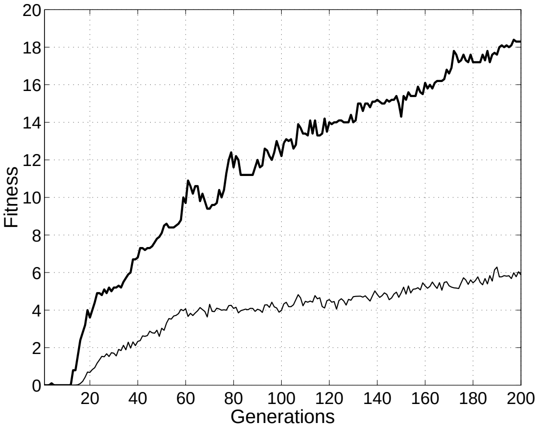
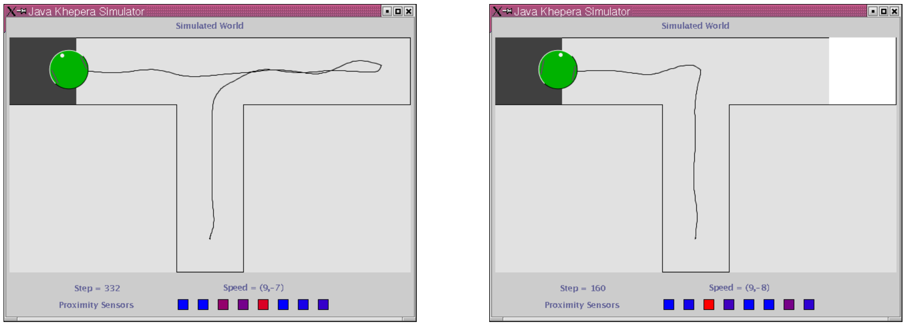


Fig. 3. *T-Maze Task.* Thick line shows best fitness and thin line shows population mean (both are averaged over 10 replications of the experiment).

of each epoch but *not* between trials within the same epoch. This means that the robot can potentially build up and store information in the dynamic state of the network between trials within the same epoch. The optimal behavior of the robot in this environment is to use the first trial of each epoch to locate and "remember" the position of the reward-zone, and thereafter move directly towards it for the remaining trials of the epoch. To put additional evolutionary pressure on this behavior, the number of available sensory-motor steps is 360 in the first trial of each epoch and only 180 in the remaining 4 trials. Given the size of the maze this means that the robot only has time to explore the whole maze during the first trial of each epoch. In addition a poison-zone (white square in figure 4(b)) is positioned opposite of the reward-zone in the last 4 but *not* the first trial of each epoch. An individual is immediately killed if it steps over the poison-zone. The *fitness function* is simply the sum of trials an



(a) *Trial 1*: The robot explores the environment, and after some time locates and stays on in the reward-zone.

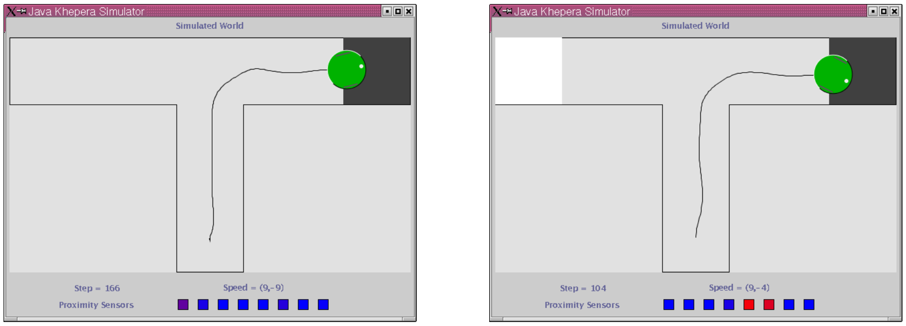
(b) *Trials 2-5*: For the remaining trials the robot exploits the "knowledge" gained in trial 1 and moves directly towards the reward-zone.

Fig. 4. Robot traces of an epoch with the reward-zone to the left

individual ends its life inside the reward-zone. Since each individual is tested for a total of 20 trials the maximal possible fitness is 20. Notice that there is no direct pressure on evolving fast moving robots given this fitness function. This is however compensated by the fact that the number of steps in each trial is limited and has been adjusted to fit the size of the environment.

The experiments are carried out using a standard genetic algorithm with rank-based selection. A population of 200 randomly generated neural controllers is evolved for 200 generations. At every generation the best 40 individuals make 5 copies each. One copy each of the 5 best individuals remains unchanged (elitism). For the rest of the population single-point crossover with a probability of 0.04 and bit-switch mutation with a 0.02 probability per bit is applied. The whole experiment is repeated 10 times using different initializations of the computer's pseudo-random number generator.

The fitness results of the evolutionary runs on this experiment are shown in figure 3. The thick line shows best fitness and the thin line shows population mean, both are averaged over 10 replications of the experiment. The evolutionary process found individuals able to collect the maximal fitness of 20 in 6 out of the 10 replications of the experiment. The maximal fitness in the 4 remaining runs was around 16. The behavior of an individual from the final generation of one of the successful runs is shown in figure 4 and 5. The robot starts out in trial 1 of the first epoch (figure 4(a)) by exploring the maze until it locates the reward-zone where it stays the remaining time of the trial. In the following trials (figure 4(b)), the robot is able to retain the "knowledge" gathered during trial 1 and always turns left at the T-junction in order to move towards and stay on the reward-zone. In the epochs with reward to the right the robot moves directly towards the reward-zone in trial 1 (figure 5(a)), since the default behavior of this



(a) *Trial 1*: The default behavior of this individual of turning right takes it directly to the reward-zone.

(b) *Trials 2-5*: For the remaining trials this behavior is repeated.

Fig. 5. Robot traces of an epoch with the reward-zone to the right.

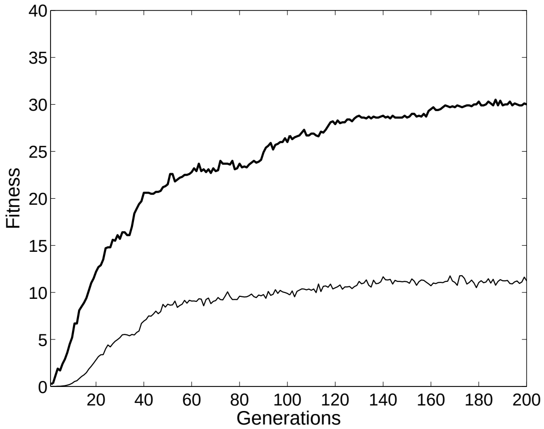


Fig. 6. *T-Maze task with reward switching.* Thick line shows best fitness and thin line shows population mean (both averaged over 10 replications of the experiment).

individual is to turn right at the first junction after an re-initialization of the neural controller. For the remaining trials this successful behavior is repeated (figure 5(b)).

3.1 Analysis

In order to better understand the functioning of the evolved neural controllers, some further analysis on an individual from one of the successful runs was done. The neural activities of each neuron were recorded over two epochs - one with reward to the left and one with reward to the right. It was found that the essential information about the current environment is stored in one of the hidden neurons. The activity of this neuron approaches zero at the end of the trials with the reward to the left and approaches one otherwise. By initializing every other neuron in the network as normal, but setting this neurons activity to either zero or one, it could be controlled which way the robot turns at the T-junction. In other words the state of this neuron "stores" the robots current assumption about the environment and is updated when these assumptions are not met. For more details about the analysis performed please refer to [1].

3.2 Transfer to the Real Robot

A way of verifying the evolutionary robotics results obtained in simulation is to test the evolved neural controllers on a real robot. For this purpose the T-Maze shown in figure 7 was built. The best individual from each of the 10 replications of experiment 1 task was tested. Initially however, the results of the tests were rather poor. None of the 10 controllers were able to reliable navigate the robot. This result indicates that the functioning of the evolved CTRNNs was specific to the sensory-motor conditions encountered in the simulator. This observation confirms our earlier results that CTRNNs, despite of their ability to display learning-likes abilities, lack the sensory-motor adaptability found in e.g. plastic Hebbian synapse networks [2]. However, several techniques for reducing this "reality gap"-problem, by adding noise at different levels of the simulation, have been proposed [5][6]. With this in mind the simulator was changed in the following way: Sensor noise levels were increased from 5% to 10%. In addition, 10% uniform noise was added to the distance traveled by each wheel at each timestep. Furthermore, the initial conditions of each tested individual changed in the following way: The starting position was randomized within a 4 by 4 cm square, and the orientation was randomized within the range forward +/- 15 degrees. With these modifications an incremental evolution lasting 20 generations was launched, seeded with a population from one of the original runs. This time the transfer to the real robot was perfect. The best individual of the last generation was able score a fitness value of 20, i.e. finding the reward-zone in every trial. No significant behavioral differences compared to the simulation were observed.

4 Experiment 2: Simple T-Maze with Reward Switching

In order to further investigate the learning-like capabilities of CTRNNs the task for the robot was now made slightly more complex. In experiment 1 the robots could rely on the fact that the position of the reward-zone remained fixed during a whole epoch. Evolved robots were able to explore the whole environment during trial 1 of each epoch in order to locate this position, but would they also be able to adapt if the reward position was changed later on within the same epoch. This turned out not to be the case. When testing the individual analyzed in section 3.1 by placing the reward-zone to left for 5 trials and then switching

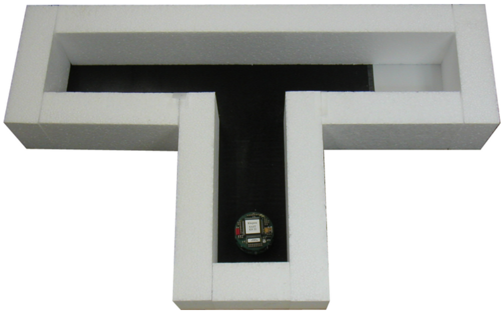


Fig. 7. *The real T-maze environment.* (Note: surface colors have been reversed compared to the simulated environment, but the processed input from the floor-sensor remains 1 inside reward-zone and 0 otherwise) .

the reward position to the right *without* resetting the neural network, the robot would continue to turn left a the T-Junction in the trials after the switch took place.

A new experiment was now set up in order to check if this lack of adaptivity to environmental changes taking place later on in an epoch was due to a limitation in the learning capabilities of the network, or simply given by the fact this condition was never met during evolution. The evolved robots could simply have found a minimalistic solution. In this new evolution the duration of each epoch was increased to 10 trials. The reward-zone position remained fixed in the first 5 trials but was then switched to the other side in the 5 last trials of each epoch. Each individual was still tested for 4 epochs, 2 with the reward initially to the left and 2 with the reward initially to the right. The fitness function remained the same, and since each individual was tested for 40 trials in total the maximum possible fitness was now 40.

The average result of 10 replications of the experiment is shown in figure 6. The resulting best fitness in the 10 replications varied between 22 in the worst case and 38 in the best. In the latter case the best individual did realize that the reward position had changed in trial 6, and was able to locate the new position. However in some of the following trials it would still turn the wrong way thus ending up in a the poison-zone. In order to increase the performance the last bit an incremental evolutionary approach was now applied. The evolutionary conditions remained the same, but instead of seeding the evolution with a random population, it was initially seeded with a population consisting of the best individual of each of the 200 generations from the best replication of the previous evolution. Again 10 replications were performed. In most of the runs the fitness level stayed at 38, and even dropped to 32 in one case (graph not shown). It seemed that level 38 solution was a local optimum which was difficult to escape. In one replication, however, the fitness level reached the maximal value of 40, and when tested afterwards the best individual from this replication could reliably solve the task. When the reward position switched at trial 6 of each epoch, the

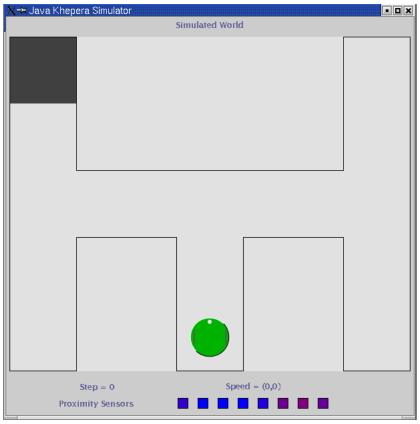


Fig. 8. *Double T-Maze.* The maze now has three T-junctions. The reward-zone is positioned either in the upper left (as shown) or upper right corner. The 3 remaining corners each contains a poison-zone during trials 2-5 (but *not* trial 1) of each epoch.

robot would at first move towards the previous location, but when not finding the reward-zone here anymore it would turn around and initiate a search until the new reward position was located. In the remaining trials of each epoch the robot then again turned directly towards the reward-zone, resulting in the total fitness of 40.

5 Experiment 3: Double T-Maze

In the previous section it was shown that evolved robots were able to completely solve the simple T-Maze even in the case when the reward position was switched during an epoch. However, the robots only had to retain one piece of information based on previous experiences, namely whether to turn left or right at the T-junction. The investigations were now turned towards a double T-maze with several T-junctions thus further complicating the task (see figure 8). To compensate for the increased size of the maze the number of sensory-motor cycles was increased to 400 in trial 1 and 200 in the remaining trials. The reward-zone could appear in either upper-left and upper-right corner of the maze.

During the first trial only the reward-zone was present, and in the following trials poison-zones were placed in the 3 remaining corners. The other parameters remained unchanged. As in the simple T-Maze, the case where the reward position remained unchanged during an entire epoch was tested first. An evolutionary process seeded with populations of random individuals was launched, but the results were very poor under these conditions. Basically the fitness remained at zero all the time, with very few exceptions of fitness 1 coming and going for a couple of generations in one of the replications of the experiment. In fact this result is not that surprising considering the fitness function used.

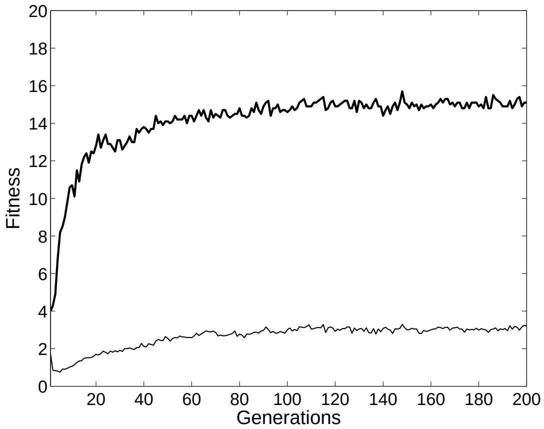


Fig. 9. *Double T-Maze task.* Thick line shows best fitness and thin line shows population mean (both are averaged over 10 replications of the experiment).

In order for a random initial individual to collect some fitness and kick-off the evolutionary progress it had to navigate all the way to one of the upper corners. If it could not do that it simply got zero fitness. A solution to this problem could have been to design an incremental fitness function, where individuals would get some fitness for partially solving the task. Instead it was decided to again rely upon an incremental evolutionary approach. The genetic algorithm was seeded with a population consisting of the best individuals from one of the runs of experiment 1 on the simple T-Maze. The results of 10 replications of the experiment are shown in figure 9. In the best run the fitness reached a level of 18 out of 20. During trial 1 the best individual always turned right at the first junction and left at the second. This would take the robot to the upper right corner of the maze. In epochs with reward to the right the robot would soon find and stay in the reward-zone. When the reward was on the left, on the other hand, the robot was not capable of searching for the reward-zone as it did in the simple T-maze case (fig. 4(a)). Instead the robot simply crashed into the upper wall. In the following trials of these epochs, however, the robot would now turn left at the first junction and right at the second, reaching the reward-zone in the upper-left corner. One trial was wasted, but crucial information about the reward position was gathered and used in the following trials, thus resulting in fitness of 18 out of 20. An attempt to further improve this behavior by an additional incremental evolution was now conducted, seeding evolution with a population of individuals from this experiment. This time, however, the attempt was not successful and the fitness level remained at 18 in every replication of the experiment. This results suggests that the maximal problem complexity solvable for the genetic algorithm and neural network used had been reached. This suggestion was confirmed by an unsuccessful attempt to apply reward-switching to the double T-Maze task, as was done in experiment 2 in the simple T-Maze case. No reliable learning behaviors were observed in the evolved controllers in this case.

6 Conclusion

We have shown that evolution of learning-like properties *is* possible without modifications of synapse strengths, but simply by relying on complex internal dynamics of CTRNNs. In experiment 1 the robot had to navigate a simple T-Maze with the reward position fixed during each epoch. Direct evolution of this task was possible and the analysis showed that the employed strategy of an evolved network was to store essential environment information in one of the hidden units. The rest of the neurons would update the activity of this neuron based on current environmental feedback. With a few simulator modifications evolved behaviors were successfully transfered to a real robot. In experiments 2 the reward position would vary within the same epoch forcing evolved robots to keep on adapting their strategy to the current environmental conditions. Successful individuals solving this task were evolved in a two-step incremental evolution. Because of the increased complexity of the maze, direct evolution was not possible in experiment 3. However, by seeding evolution with a population from experiment 1 individuals capable of "learning" were found. It was not possible to perform additional reward switching as in experiment 2 on the simple T-Maze, suggesting that maximal task complexity had been reached.

In this work incremental evolution has proven to be a powerful tool in evolving complex robot behaviors, however evolving CTRNNs as shown here will face evolvability problems if the task complexity is to be further increased. In principle a sufficiently large CTRNN is able to display arbitrarily complex dynamics. However, the problem of how to evolve such networks will have to be addressed in the future. Possible solutions could be to explore new neural mechanisms for information "storage", or to investigate how to preserve the learning capabilities of CTRNNs in networks generally thought of to be easier to evolve such as Hebbian synapse networks or spiking neural networks. Our work will focus on these aspects in the future.

As pointed out by one of the reviewers it can be argued whether the experiments presented in this paper should be classified as learning-like behaviors or simply as internal dynamics investigations. It true that the view of learning presented in this paper is quite different from the traditional computational view of learning, where some update of the control systems always takes place. However neurophysiological experiments have indicated that the way animals and humans perceive, classify, and memorize, for example in the olfactory system, is by transitions between chaotic attractors in dynamical systems formed by large numbers of neurons in the brain [3][7]. These results correspond nicely with the view of memory and learning presented in this paper.

Acknowledgements

The authors wish to thank Tom Ziemke for useful comments and suggestions. This work was supported by the Swiss National Science Foundation, grant nr. 620-58049.

References

1. J. Blynel. Evolving reinforcement learning-like abilities for robots. In *(to appear) Proceedings on the 5th International Conference on Evolvable Systems (ICES'03): From Biology to Hardware*. Springer Verlag, Berlin, 2003.
2. J. Blynel and D. Floreano. Levels of dynamics and adaptive behaviour in evolutionary neural controllers. In Hallam et al. [4], pages 272–281.
3. W. J. Freeman. The physiology of perception. *Scientific American*, 264:78–85, 1991.
4. B. Hallam, D. Floreano, J. Hallam, G Hayes, and J-A Meyer, editors. *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*. MIT Press-Bradford Books, Cambridge, MA, 2002.
5. N. Jakobi. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior*, 6(2):325–368, 1997.
6. O. Miglino, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial Life*, 2(4):417–434, 1995.
7. R. Pfeifer and C. Scheier. *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.
8. J. Tani. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26:421–436, 1996.
9. J. Tani and S. Nolfi. Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems. *Neural Networks*, 12(7–8):1131–1141, 1999.
10. E. Tuci, I. Harvey, and M. Quinn. Evolving integrated controllers for autonomous learning robots using dynamic neural networks. In Hallam et al. [4], pages 282–291.
11. B. Yamauchi and R. D. Beer. Integrating reactive, sequential, and learning behaviour using dynamical neural networks. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, *From Animals to Animats III: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 382–391. MIT Press-Bradford Books, Cambridge, MA, 1994.
12. B. Yamauchi and R. D. Beer. Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2(3):219–246, 1994.
13. T. Ziemke and M. Thieme. Neuromodulation of reactive sensorimotor mappings as a short-term memory mechanism in delayed response tasks. *(to appear) Adaptive Bebavior*, 10(3–4), 2003.