

Evolved Motor Primitives and Sequences in a Hierarchical Recurrent Neural Network

Rainer W. Paine and Jun Tani

RIKEN Brain Science Institute, Laboratory for Behavior and Dynamic Cognition,
2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan
{rpaine, tani}@brain.riken.jp

Abstract. This study describes how complex goal-directed behavior can evolve in a hierarchically organized recurrent neural network controlling a simulated Khepera robot. Different types of dynamic structures self-organize in the lower and higher levels of a network for the purpose of achieving complex navigation tasks. The parametric bifurcation structures that appear in the lower level explain the mechanism of how behavior primitives are switched in a top-down way. In the higher level, a topologically ordered mapping of initial cell activation states to motor-primitive sequences self-organizes by utilizing the initial sensitivity characteristics of nonlinear dynamical systems. A further experiment tests the evolved controller's adaptability to changes in its environment. The biological plausibility of the model's essential principles is discussed.

1 Introduction

It is widely believed that behavior systems develop certain hierarchical or level structures for achieving goal-directed complex behaviors, and that these structures should self-organize through interactions with the environment. It is reasonable to assume that an abstract event sequence is represented in a higher level, while its detailed motor program is generated in a lower level. Arbib [1] proposed the idea of movement primitives (also referred to as perceptual-motor primitives, motor schemas, or motor programs), which are a compact representation of action sequences for generalized movements that accomplish a goal. Evidence of such primitives in animals has been found ([2], [3]), and human studies also indicate their role in complex movement generation [4]. Once such motor primitives develop early in the life of an organism, diverse behaviors can emerge by learning to combine them in a multitude of complex sequences.

In this paper, we study the dynamic adaptation process of multiple levels of continuous time recurrent neural networks (CTRNNs) applied to a navigation task using a simulated robot. Through the experiments, we will demonstrate that a hierarchically organized network can perform well in adapting to complex tasks through combination with a genetic algorithm (GA). We will focus on how motor primitives are self-organized in the lower level, and how they are manipulated in the higher level.

2 Methods

The neural network model utilized in the current paper consists of two levels of fully connected CTRNNs ([5], [6]). The lower level network, as shown in Figure 1, re-

ceives sensory inputs and generates motor commands as outputs. This network is supposed to encode multiple sensory-motor primitives, such as moving straight down a corridor, and turning left or right at intersections or to avoid obstacles in the navigation task adopted in this study.

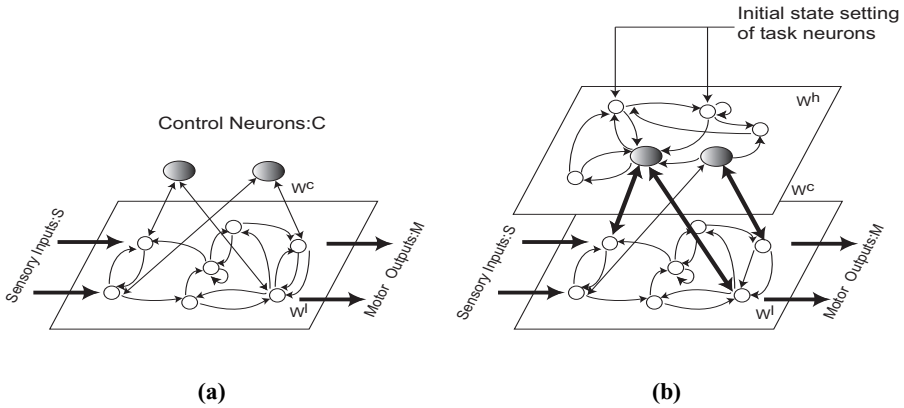


Fig. 1. Conceptual diagram of network architecture

A set of external neural units, called the “control neurons”, are bidirectionally connected to all neurons in the lower level network. The control neurons influence lower level network functions and favor the generation of particular motor primitives. Through evolution of both the lower level internal synaptic weights (W_i) and the interface weights (W_c) between the control neurons and the lower level neurons, a mapping between the control neurons’ activities and the sensory-motor primitives stored in the lower level network is self-organized. Modulation of the control neurons’ activities causes shifts between generating one primitive and another. The scheme is analogous to the idea of the parametric bias [7] and the command neuron concept ([8], [9], [10]). How might more complex tasks, such as navigation in an environment, be generated? Such tasks require generating sequences of motor primitives. We propose that a higher level network may modulate the activities of the control neurons through time to generate sequences of lower level movement primitives (Figure 1b).

The higher level network evolves to encode abstract behavior sequences utilizing the control neurons. It is assumed that the desired sequences will be generated if adequate nonlinear dynamics can be self-organized in the higher level network. As will be described in detail later, the robot becomes able to navigate to multiple goal positions when starting from the same initial position in the maze environment. Therefore, the higher level network must encode multiple sequence patterns which have to be retrieved for the specified goal.

We utilize the initial sensitivity characteristics of nonlinear dynamic systems in order to initiate different sequences. When the robot is placed at the initial position in the environment, the internal values of all the higher level neurons are set to 0.0, except for two neurons called the task neurons (Figure 1b). The initial activity values of

the task neurons determine the subsequent turn sequence, and the goal which is found. These goal-specific initial task neuron activities were evolved through the same genetic algorithm that yielded the network's synaptic weights.

3 Experiments

All experiments reported here were executed using a simulated Khepera II robot in the Webots 3 robot simulator (www.cyberbotics.com). The same CTRNN equations, parameters, and genome encoding were used as in [6], with the addition of the initial task neuron activities, $\gamma_{task}(0) \in [-10, 10]$, to the genome for experiment 2. A standard GA was employed, with a 2% bit-mutation rate, an 80 robot population with the best 20 reproducing, and no mutation of one offspring of the best parent ([6], [14], [15]).

3.1 Experiment 1: T-Maze Task

Experiment 1 is designed to evolve a bottom level network which contains movement primitives of left and right turning behavior at intersections as well as collision-free straight movement in corridors. The same lower level and control neuron weights are used for both right and left turns. The only difference between the left and right turn controllers is in the bias values (θ) of the two control neurons. Intuitively, this might correspond to different sets of cortical "control" neurons becoming associated with each of the lower level movement primitives. Parallel connections from the bottom level to the control neurons might develop, yielding the same weights to both sets of control neurons. Intrinsic differences in the control neurons' responses (as through the different θ bias values used here) to the lower level signals would determine with which motor primitives each set of control neurons became associated.

The evolutionary runs consisted of up to 200 generations with 2 epochs and 3 trials per robot. Each trial was run for 500 time steps, starting at the same position at the bottom of the T maze. Different bias values (θ) evolved in the control nodes for the left and right turning tasks in epochs 1 and 2, respectively. All other parameters were identical in the left and right turning tasks. In epoch 1, fitness was awarded to robots that turned to the left at the intersection based on the following fitness rule. In epoch 2, fitness was awarded to robots that turned to the right. Each robot ran 3 trials per epoch.

Experiment 1 uses a two-component fitness rule. The first component consists of a reward for straight and fast movements with obstacle avoidance [16]. The second component of the fitness rule rewards the robot for finding a goal. The goal is located to the left of the intersection for epoch 1, and to the right for epoch 2. The robot is linearly rewarded, based on its position, for approaching and reaching the goal. Greater reward per time step is received linearly as the robot approaches the goal, starting at the middle of the top of the T maze.

At the start of each trial, the robot was placed at the same starting position at the bottom of the T maze. Three different starting orientations (facing 135° , 90° , and 45° ; that is, left, straight, and right, respectively) were employed, one for each of the three trials

per epoch. Further, motor noise was added to the wheel speed commands sent to the robot. The integer speed command was increased or decreased by one with a probability of 20% on each simulation time step. The varying starting orientations and motor noise were used to ensure that the robot would experience wall collisions early during evolution, so that controllers with obstacle avoidance would be more likely to evolve. Both the bottom level and control neurons were free to evolve in this experiment.

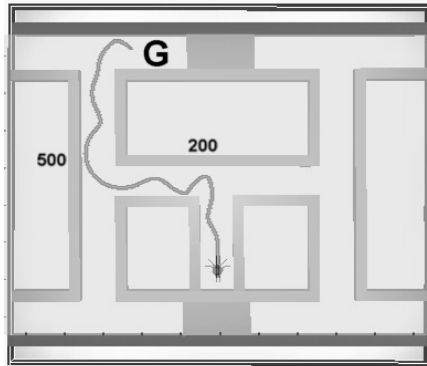


Fig. 2. Eight-Goal Maze environment, showing trajectory for a Left, Right, Right sequence. (G = Goal location). Turns occur at steps 200 and 500 in the neuronal activity traces for this sequence in Figure 4

3.2 Experiment 2: Eight-Goal Task

The Eight-Goal maze depicted in Figure 2 is a combination of T maze-like environments with eight different goals at the ends of each T maze component. Thus, combinations of the same turn primitives evolved in experiment 1 should allow the robot to reach the different goals. In experiment 1, different sets of control neurons with differing internal dynamics (due to differing θ bias values) became associated with particular motor primitives, as will be described later. In experiment 2, it is shown how the activity of a *single* set of control neurons can be modulated over time to generate a *sequence* of motor primitives. Further, it is shown how varying only the initial activation of the task neurons in the higher level network can lead to the generation of different network activation time courses by which multiple primitive sequences are generated. Thus, the routes to multiple goals can effectively be stored in a single network, with a single set of synaptic weights and corresponding initial activation values of the task neurons.

The bottom level genome, including the weights for the connections to the control neurons, from experiment 1 was used in this experiment and held constant. Only a

single set of synaptic weights and parameters in the higher level network, and multiple sets of the initial task neuron activities, were free to evolve. The experiment

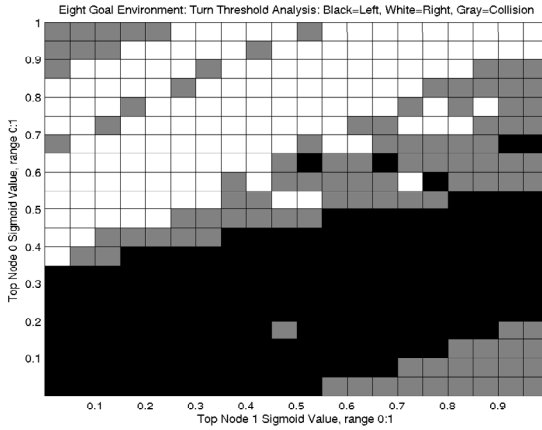


Fig. 3. Phase analysis of turn direction as a function of control neuron activation. X and Y axes: neuronal activities of control neurons 1 and 0, respectively. (Black = Left turn, White = Right turn, Grey = Collision with wall)

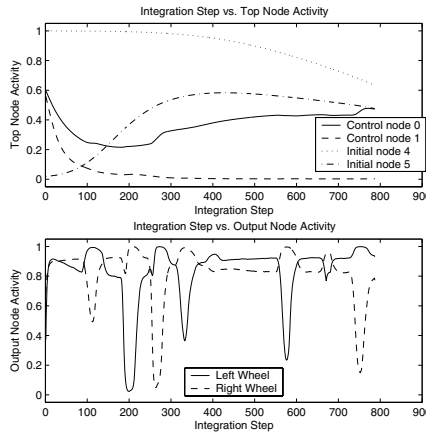


Fig. 4. Left-Right-Right turn sequence (trajectory in Figure 2). (Top) Neuronal activity of control and task (initial node) neurons; (Bottom) Lower level motor output node activity

consisted of up to 200 generations, with 12 epochs per generation and 2 trials per epoch. Each of the 12 epochs evaluated a different set of higher level task neuron initial activities, using the fitness rule described below. Further, each task neuron set was run for two trials, in order to evaluate the stability of the robot’s goal-finding ability.

In experiment 2, the fitness rule (F) consists solely of a reward for finding goals consistently (Equation 1). Here, a fixed reward (R) per new goal (g) found is given to the robot. In experiment 2, $N_{goals} = 8$, and $N_{tasks} = 12$. Each robot has 12 sets of task neuron initial activities (i) which are evaluated. Each set has two trials in which to find a goal. A robot which finds a goal on both trials receives twice the reward of finding the goal on only one trial. If a different set of task neuron initial activities

$$F = \sum_{g=1}^{N_{goals}} \max_{i=1:N_{tasks}} \left(\sum_{trial=1}^2 R_{gi} \right) \quad (1)$$

leads to the same goal, then the reward is the maximum of the reward given to the two different task neuron sets. Thus, a robot with multiple task neuron sets that find a goal on only one of the two trials will receive less reward than a robot with one task neuron set that finds the goal on both trials.

3.3 Experiment 3: Sequence Retention in an Enlarged Environment

An interesting question is whether travel time or distance is intrinsically represented in the dynamics of the evolved neural network, or whether the turn sequence can be generated over any arbitrary distance and time. In order to answer this question, the following experiment was conducted.

The size of the eight-goal environment from experiment 2 (Figure 2) was doubled. More specifically, the corridor distances were doubled, while the corridor widths remained the same. The evolved controller from experiment 2 was then used to control the robot in this larger environment. The initial task neuron values which led to the six stable goals in experiment 2 were then loaded into the controller, and the resulting movement sequences were observed.

4 Results/Analysis

4.1 Experiment 1: T-Maze Task

Collision avoidance and left and right turning behavior emerged within 63 generations. Left turns were generated for one set of control neuron bias values, and right turns were generated for another set. Although both left and right turns could be generated, the robot exhibited oscillatory movements after collision avoidance. That is, it turned away from one wall too much and headed towards the opposite wall instead of straightening its path through the lower part of the T maze. (As mentioned previously, the robot started from three different orientations, leftward, straight up, and rightward, requiring it to avoid wall collisions early during each trial.) The controller was therefore allowed to evolve further. By generation 189, fewer fluctuations occurred after collision avoidance and the lower level genome was used for experiments 2 and 3.

The relation between turn behavior and the activities of control neurons 0 and 1 is quantified in the phase plot of Figure 3. 441 trials with different sets of the two control neurons' activities ($A=[0:1]$, step size = 0.05), held constant for each trial, were run and the resulting turn directions recorded. Note that this figure also applies to the activity of the control neurons evolved in experiment 2, since the same bottom level genome, including the synaptic weights between the bottom level and control neurons, was used in both experiments 1 and 2. The turn phase plot of Figure 3 shows a

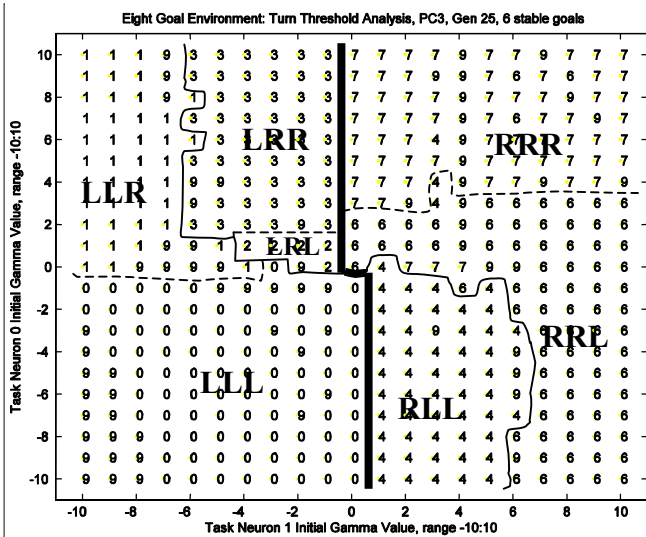


Fig. 5. (a) Phase analysis of three-turn sequence generation as a function of task neuron initial activity, $\gamma_{task}(0)$. X and Y axes: Initial activities of task neurons 1 and 0, respectively. Plotted numbers correspond to sequences (L=Left, R=Right) in figure. (9=collision)

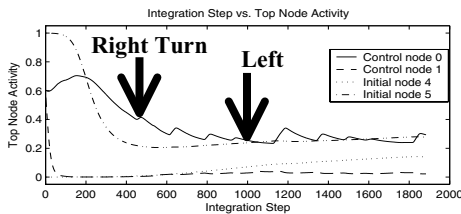


Fig. 6. Experiment 3. Control and Task neuron (*initial node*) activities when the distance to the goal is doubled. In the original smaller environment of Experiment 2, this set of initial task neuron activities led to a Right-Right-Left turn sequence. When the distance to the goal was doubled, a Right-Left-Left sequence occurred since the activities of control neurons 0 and 1 passed the turn threshold prior to reaching the second intersection. The second turn (*Right*) occurred at step 450 in the original environment, and at step 1000 (*Left*) in the double-length environment

clear bifurcation, or appearance of new movement behavior with the control neuron activity change, with two distinct regions of stability for left (black) and right (white) turns, for the corresponding combinations of control neuron 0 and neuron 1 activities. The gray squares indicate unstable regions in which wall collisions occur. In both experiments 1 and 2, the evolved control neuron weights tend to suppress the activity of control neuron 1. The phase plot shows that the smallest region of instability between left and right turns occurs for such small control neuron 1 activities. Further, the weights from control neuron 0 to the two motor output nodes ($w_{jk} = 5.0, -3.4$) of the bottom level have a greater magnitude than the weights from control neuron 1 ($w_{jk} = -1.1, 2.0$). Thus, control neuron 0 has the dominant effect on the turn direction of the lower level's output neurons. It therefore appears that the lower level, which receives and processes all sensory inputs, has a dominant role in collision avoidance. Collision avoidance competes with the control neurons' "turn" signals. In the unstable regions of the phase plot (grey in Figure 3), the control neurons' "turn" commands inappropriately override the lower level's collision avoidance, triggering collisions when the robot turns toward and collides with a wall.

4.2 Experiment 2: Eight-Goal Task

The best robot became able to reach up to 7 different goals stably within 24 generations (~ 2 days of run time) by evolving the higher level network and control neurons. The turn sequence for each goal was determined by a particular set of initial task neuron activities ($\gamma_{task}(0)$). Since 12 different sets of $\gamma_{task}(0)$ were evaluated per robot, multiple sets would sometimes lead to the same goal. Each $\gamma_{task}(0)$ set was evaluated for the stability of its goal. Some values of $\gamma_{task}(0)$ led to repeatable goal-finding performance, while others were unstable, leading to different goals on different trials, or even to wall collision. Although evolution led to controllers which could find all 8 goals, the best controller could only find 7 goals stably. The definition of stability used here is reaching a particular goal, with the corresponding evolved $\gamma_{task}(0)$ values, on at least 70% of trials.

Different $\gamma_{task}(0)$ activities led to differently fluctuating patterns in the control neuron activities (Figure 4). As in the results of experiment 1 reported above, control neuron 0 had the greatest influence on turn direction, and exhibited the greatest fluctuations during various turn sequences. Left turns were generated when its activity was below a threshold of approximately 0.35, and right turns were generated when the activity was above the threshold. The activity of control neuron 1 tended to be suppressed, leading to a smaller region of instability at the transition from left to right turns in the phase plot of Figure 3. Figures 2 and 4 show the trajectory and neuronal activities, respectively, for a left-right-right turn sequence. The results shown here are for a controller which learned to reach six goals stably in 25 generations. They show large amplitude fluctuations of the control neurons, due to a smaller evolved value of the neuronal time constant, τ .

The amplitude of the control neurons' fluctuations is also significant because larger amplitude fluctuations may render the controller more robust to noise. Although 5%

sensor noise was used throughout these experiments, motor noise (increasing or decreasing the wheel speed command by 1 unit with 20% probability) was used only in experiment 1 in order to facilitate the development of obstacle avoidance. Motor noise was also tested during separate evolutionary runs in experiment 2, and found to decrease the number of stable goals found from a maximum of 7 (without motor noise) to 5 (with motor noise). Note that the total number of goals found was eight, with or without motor noise. Thus, additional noise added to the wheel motor commands increases the instability of the turn sequences.

With or without motor noise, goal instability was most often seen in the final turn direction of the three-turn sequences learned. This final turn instability can be appreciated by noting that the control neuron activity tends to hover around the turn threshold near the end of the trial. As seen in the phase plot of Figure 3, this region is unstable.

Figure 5 shows an analysis of the movement sequences generated for the range of task neuron initial activities, $\gamma_{task}(0) \in [-10, 10]$, in the evolved controller of experiment 2. 441 sets of initial task neuron activities were tested and the resulting turn sequences recorded. The numbers in the figure correspond to movement sequences as labeled in the figure, e.g., LRL for a left, right, and left turn sequence. The sequence patterns are arranged in well-defined, topologically ordered clusters in the $\gamma_{task}(0)$

space. First, the $\gamma_{task}(0)$ space is grossly clustered based on the first turn direction, left or right, of the movement sequence, as shown by a thick solid line in Figure 5. Each of these two clusters is then further divided into sub-clusters, depending on the second turn direction of the movement sequence, as shown by a solid line. These sub-clusters are still further divided into smaller clusters, depending on the third turn as shown by the dashed lines.

This mapping of initial task neuron activity to particular sequences is an emergent property of the evolved controller. Different evolutionary runs yield different cluster patterns, but the general trend of distinct, topologically ordered sequence regions remains. This self-organized, topologically ordered mapping of sequences, with increasing initial sensitivity to task node activities as movement sequence complexity increases, is notable in such a small network, and is reminiscent of the fractal distribution of sequences mapped in the parameter space of [11]. Indeed, it would be interesting to see if fractal structure could be found in controllers branching out to larger numbers of goals.

4.3 Experiment 3: Sequence Retention in an Enlarged Environment

When six sets of task neuron initial activities ($\gamma_{task}(0)$), corresponding to six stable goals found in experiment 2, were loaded into the controller of the robot when placed in an enlarged environment, only two out of the six $\gamma_{task}(0)$ sets led to the robot reaching the same goal as in the smaller environment of experiment 2. These results indicate that the internal dynamics of the higher level neurons appear to proceed at their own rate, relatively independent of external inputs. For example, the Right-Right-Left (RRL) sequence turned into an RLL sequence after doubling the maze size since the second right turn was missed due to the longer corridor. The control neu-

rons' outputs then passed the turn threshold, leading to a premature left turn when the robot finally arrived at the intersection (Figure 6). This result is not surprising, since no direct sensory input reaches the higher level network. It would be interesting to explore in future research the capability of the control neurons, which are connected more closely to sensory signals via their connections with the bottom level network, to modify the higher level activity based on environmental changes.

5 Discussion/Conclusion

The work presented here describes a novel hierarchical model of behavioral sequence memory and generation. It recalls in general terms the hierarchical organization of movements in the primate spinal cord, brainstem, and cortical regions. Different types of dynamic structures self-organize in the lower and higher levels of the network. A parametric bifurcation in the control neurons' interaction with the lower level allows top-down behavioral switching of the primitives embedded in the lower level. Utilizing the initial sensitivity characteristics of nonlinear dynamic systems [17], a topologically ordered mapping of initial task neuron activity to particular behavior sequences self-organizes throughout the development of the network. The interplay of task-specific top-down and bottom-up processes allows the execution of complex navigation tasks.

One unique feature of the current model is the hierarchical organization of the network and its training. The bottom level network represents movement primitives, such as collision avoidance and turning at intersections. Since it must directly deal with quickly changing environmental stimuli, its time constants (τ) have become small through adaptation so that the neuronal activity of the output neurons ($\tau_0 = 1$, $\tau_1 = 1$) can change rapidly to drive the robot's movement in real time. In contrast, the higher level represents sequences of the lower level primitives over longer time spans. Accordingly, the task neuron time constants have adapted to be large ($\tau_{\text{task0}} = 70$, $\tau_{\text{task1}} = 52$) so that neuronal activity changes much more gradually and is less affected by short-term sensory changes.

The neurons of the higher level receive no direct sensory inputs, but are gradually influenced by them through the control neurons, which are fully connected to the input-receiving bottom level. This system is reminiscent of the organization of sequence generation in primates ([13], [18]). In [13], cellular activity in monkeys' supplementary motor area (SMA) was found to be selective for the sequential order of forthcoming movements, much as the task neurons' initial activities determine future movement order in the current model. In [18], distinct groups of cells in the lateral prefrontal cortices (LPFC) of monkeys were found to integrate the physical and temporal properties of sequentially reached objects, in a manner analogous to integration of higher level sequential information and lower level sensory input by the control neurons in the present model.

Although other models of sequence generation have been trained in a modular fashion because it was felt necessary to achieve the task [5], the current work begins by explicitly evolving simple movement primitives, such as straight movements, collision avoidance, and turning at corners. The next level of the hierarchy subsequently

develops to utilize the lower level primitives in complex movement sequences. One can envision further levels of complexity, with higher levels representing sequences of sequences for different sets of tasks, in a manner analogous to the “chunking” phenomenon observed in human memory of data sequences [19]. The beauty of this system is that the synaptic connections need not grow without bound as the number and complexity of sequences increases. As shown here, a *single* network can represent *multiple* complex movements through modulation of the activities of a small number of “task” neurons.

Although the initial sensitivity of the movement sequences generated to task neuron activations was an emergent feature of the system found by self-organization of network parameters through a genetic algorithm, the model architecture was predetermined, and the details of the network training influenced the specific functions that were assumed by different components of the architecture. Given that the current network architecture is loosely based upon the primate motor system’s hierarchical design, one might expect it to perform better than a less biologically plausible giant first-order network that encompasses both simple movement primitives as well as their combination into complex sequences. This assumption will be tested in future work.

As seen in the results of experiment 3, one limitation of the current model is its inability to respond effectively to environmental changes without further weight modification. When the length of the corridors was doubled, the robot usually failed to reproduce the turn sequence which had been learned in the smaller environment. The dynamics of the higher level network were essentially independent of the external environment, as if the robot were executing a learned sequence by rote at a fixed speed. When environmental changes prevented the robot from turning at the usual time, the top network activity continued to progress toward the next turn in the sequence, skipping a turn instead of merely delaying it until the next intersection.

The higher level network’s influence on the outputs of the lower level is disproportionately large compared to the lower level’s influence on it. The higher level’s relative isolation from the “real world’s” sensory input is in stark contrast to the rich flow of both physical and temporal sensory information which is integrated in the primate lateral prefrontal cortex during the learning of movement sequences [18]. Although both the model’s control neurons and primate LPFC neurons integrate both temporal sequence and physical sensory information, the monkey can modulate the speed of its sequence generation, whereas the current model cannot. Future work will therefore explore the possibility of better modulating the activity of the higher level through bottom-up connections in a way which reflects environmental changes.

References

1. Arbib, M. A.: Perceptual structures and distributed motor control. In: Brooks, V. B. (ed.): Handbook of Physiology, Section 2: The Nervous System, Vol. II, Motor Control, Part 1. American Physiological Society (1981) 1449-1480
2. Giszter, S. F., Mussa-Ivaldi, F. A., Bizzi, E.: Convergent force fields organized in the frog’s spinal cord. *Journal of Neuroscience*, 13(2) (1993) 467-491

3. Mussa-Ivaldi, F. A., Giszter, S. F., Bizzi, E.: Linear combination of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences, USA*, 91 (1994) 7535-7538
4. Thoroughman, K. A., Shadmehr, R.: Learning of action through combination of motor primitives. *Nature*, 407 (2000) 742-747
5. Yamauchi, B., Beer, R.D.: Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2(3) (1994) 219-246
6. Blynel, J., Floreano, D.: Levels of dynamics and adaptive behavior in evolutionary neural controllers. In: Hallam, B., Floreano, D., Hallam, J., Hayes, G., Meyer, J.A. (eds.): *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*. MIT Press, Bradford Books, Cambridge, MA (2002)
7. Tani, J.: Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks*, 16(1) (2003) 11-23
8. Aharonov-Barki, R., Beker, T., Ruppin, E.: Spontaneous Evolution of Command Neurons, Place Cells and Memory Mechanisms in Autonomous Agents. In: *Advances in Artificial Life, ECAL '99, Vol. 1674. Lecture Notes in Artificial Intelligence*, Springer Verlag (1999)
9. Edwards, D. H., Heitler, W. J., Krasne, F. B.: Fifty years of a command neuron: the neurobiology of escape behavior in the crayfish. *Trends in Neurosciences*, 22(4) (1999) 153-161
10. Teyke, T., Weiss, K. R., Kupfermann, I.: An identified neuron (CPR) evokes neuronal responses reflecting food arousal in *Aplysia*. *Science*, 247, (1990) 85-87
11. Nishimoto, R., Tani, J.: Learning to Generate Combinatorial Action Sequences Utilizing the Initial Sensitivity of Deterministic Dynamical Systems. *Proc. of the 7th International Work-Conference on Artificial and Natural Neural Networks (IWANN'03)*. Springer, (2003) 422-429
12. Blynel, J.: Evolving Reinforcement Learning-Like Abilities for Robots. In: Tyrrell, A, Haddow, P.C., and Torresen, J. (eds.): *Evolvable Systems: From Biology to Hardware: 5th International Conference, ICES (2003)*
13. Tanji, J., Shima, K.: Role for supplementary motor area cells in planning several movements ahead. *Nature*, 371 (1994) 413-416
14. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA (1998)
15. Goldberg, D.E.: *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Boston, MA (2002)
16. Floreano, D., Mondada, F.: Automatic creation of an autonomous agent: genetic evolution of a neural-network driven robot. In: Cliff, D., Husbands, P., Meyer, J., Wilson, S.W. (eds.): *From Animals to Animats 3: Proceedings of the Third Conference on Simulation of Adaptive Behavior*. MIT Press, Bradford Books, Cambridge, MA (1994)
17. Fan, J., Yao, Q., Tong, H.: Estimation of Densities and Sensitivity Measures in Nonlinear Dynamical Systems. *Biometrika*, 83(1) (1996) 189-206
18. Ninokura, Y., Mushiake, H., Tanji, J.: Representation of the Temporal Order of Visual Objects in the Primate Lateral Prefrontal Cortex. *Journal of Neurophysiology*, 89 (2003) 2868-2873
19. Sakai, K., Kitaguchi, K., Hikosaka, O.: Chunking during human visuomotor sequence learning. *Experimental Brain Research*, 152(2) (2003) 229-242