Maze Navigation and Memory with Physical Reservoir Computers

Chris Johnson[1], Andrew Philippides[1] and Philip Husbands[1]

[1]Centre for Computational Neuroscience and Robotics
University of Sussex
c.a.johnson@sussex.ac.uk

Abstract

The extent to which an organism's morphology may shape its behaviour is increasingly studied, but still not well understood (McGeer, 1990; Pfeifer and Bongard, 2007; Nakajima et al., 2015; Caluwaerts et al., 2012; Zhao et al., 2013). Hauser et al. (2011, 2012) introduced mass-spring-damper (MSD) reservoir networks as morphologically computing abstracted bodies. As these networks are abstracted from biological bodies, the two will share some properties and capabilities, and studying the former may give us useful clues about the latter. We have previously applied small MSD network pairs to the production of reactive behaviour often referred to as 'minimally cognitive' (Johnson et al., 2014, 2015). Here we go on to use similar controllers to solve a target-seeking problem for a mobile agent in a maze, which necessitates memory, over a finite but extended period. If MSD networks with relatively few elements but still high dynamic complexity can solve navigation problems requiring this kind of short term memory, then we may speculate that simple organisms can also.

MSD networks

In simulated MSD networks, point masses are connected by links which each consist of a spring and a damper in parallel (see Fig. (2)). In a true reservoir computing (RC) approach, the stiffness and damping coefficients of these elements, respectively, are randomised when the network is generated and do not change. Networks receive inputs in the form of forces applied to the point masses. The network forms a kernel, which projects its input streams into its higher-dimensional, and highly nonlinear, state space. In keeping with reservoir computing, the output from the network is a linear weighted sum of network states, and only these weights are trained.

Methods and Results

In an experiment based on that of Blynel and Floreano (2003), where a CTRNN controller was used, a group of 4 small MSD networks control a simulated ePuck robot which must navigate a T-maze and locate a target zone which is placed at the end of either the left or the right corridor (see Figures 1, 2 and 3). When the target is located, the robot is

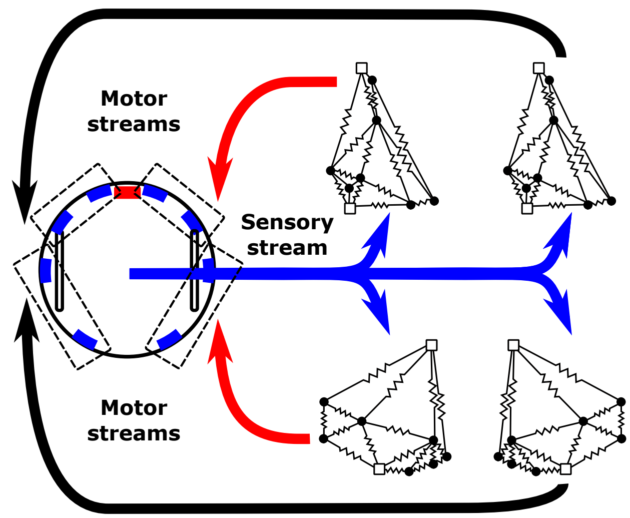

Figure 1: The ePuck and controller configuration. Camera (unused here) shown in red, on front of robot. Infrared (IR) distance sensors shown in blue. The rectangles with dashed outlines show how the sensors are grouped in pairs. Stimuli from paired sensors are summed before being input to the networks, so that the networks receive 5 inputs: 4 IR and one ground sensor to detect the target zone. The robot's wheels are shown as rounded open rectangles. Each of the robot's motors is controlled by an antagonistic pair of MSD networks. The back networks are identical to one another, but formed into a symmetric pair by connecting their inputs in reverse orders. In the case of the front networks, they are paired in the same way as the back ones, but symmetry is broken as they do not have the same weights as each other in the linear readout.

moved back to the start position and must navigate directly to the target zone, without searching other locations. In order to solve this problem, the robot must have memory of the target location at least until reaching the maze junction. Evolutionary methods are used to determine valid network parameters. In a break with the standard RC approach, stiffness and damping coefficients as well as readout weights

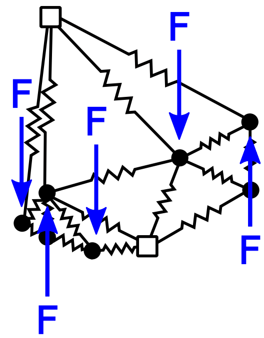

Figure 2: An MSD network from a valid controller. Square nodes are fixed, and round nodes move freely. Nodes are connected by links which consist of a parallel combination of spring and damper (dampers are omitted from this diagram for clarity). Inputs to the network are applied as forces on a subset of the free nodes. In this case, there are 9 nodes in total, and only 5 of the free 7 are driven.

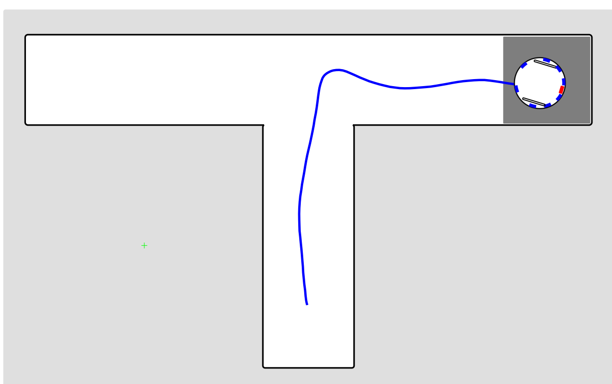

Figure 3: The ePuck in the T-maze. The target zone is shown in dark grey. In half of the evaluation trials, the target is on the right-hand side, as shown. In the other half it is on the left-hand side.

are evolved. The evolutionary algorithms and the MSD networks are implemented in the MATLAB IDE. The robot is simulated using the Enki 2D robot simulator C++ libraries (Magnenat et al., 2013).

Blynel and Floreano (2003) found that their CTRNNs stored the location of the target zone in the state of a network node. However, due to the 'fading memory' property, the MSD networks used here cannot store information indefinitely without the addition of a feedback loop (Hauser et al., 2012). In order to provide the possibility of memory in our controllers we added a feedback loop to a single network. Our first main discovery was that successful evolved controllers did not show signs of persistent memory. That being the case, we also tried evolving controllers without feedback loops. In both cases, it appears that transients in the network responses to sensory input are surviving for approximately 45s (and perhaps longer), such that the robot makes the correct turn when it reaches the maze junction.

Conclusion

Without the addition of a feedback loop (Hauser et al., 2012), the MSD networks used here are limited by the 'fading memory' property. Practically, this means that the effect of any sensory input to the networks will have finite duration. To succeed in this target-seeking challenge, agents must have networks with dynamics rich enough to encompass quick response to sensors for steering and long-term transients for memory. The networks used here have been kept small to illustrate that networks with relatively simple structures can still exhibit complex dynamics which can be tuned by evolution to solve non-trivial behavioural problems. The linear readout of an MSD network can be interpreted as analogous to a primitive or peripheral nervous system. If these controllers can solve the problem of target-seeking navigation requiring short-term memory, then perhaps similarly primitive organisms can perform similar tasks.

Acknowledgements

Chris Johnson was funded by a University of Sussex graduate teaching assistantship.

References

Blynel, J. and Floreano, D. (2003). Exploring the t-maze: Evolving learning-like robot behaviors using ctrnns. In *Applications of Evolutionary Computing*, pages 593–604. Springer, Berlin Heidelberg.

Caluwaerts, K., D'Haene, M., Verstraeten, D., and Schrauwen, B. (2012). Locomotion without a brain: Physical reservoir computing in tensegrity structures. *Artificial Life*, 19(1):35–66.

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., and Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biological cybernetics*, 105(5-6):355–370.

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., and Maass, W. (2012). The role of feedback in morphological computation with compliant bodies. *Biological cybernetics*, 106(10):595–613.

Johnson, C., Philippides, A., and Husbands, P. (2014). Active shape discrimination with physical reservoir computers. In Sayama, H., Rieffel, J., Risi, S., Doursat, R., and Lipson, H., editors, *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, pages 176–183, Cambridge, MA. MIT Press.

Johnson, C., Philippides, A., and Husbands, P. (2015). Active shape discrimination with compliant bodies as reservoir computers. *Artificial Life*. (in press).

Magnenat, S., Waibel, M., and Beyeler, A. (2013). Enki - an open source fast 2d robot simulator (v1.0). http://home.gna.org/

enki/. stephane.magnenat@epfl.ch, markus.waibel@epfl.ch, antoine.beyeler@epfl.ch.

McGeer, T. (1990). Passive dynamic walking. *The international journal of robotics research*, 9(2):62–82.

Nakajima, K., Hauser, H., Li, T., and Pfeifer, R. (2015). Information processing via physical soft body. *Scientific reports*, 5.

Pfeifer, R. and Bongard, J. (2007). *How the body shapes the way we think: a new view of intelligence*. MIT press.

Zhao, Q., Nakajima, K., Sumioka, H., Hauser, H., and Pfeifer, R. (2013). Spine dynamics as a computational resource in spine-driven quadruped locomotion. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1445–1451. IEEE.