

Center-Crossing Recurrent Neural Networks for the Evolution of Rhythmic Behavior

Boonyanit Mathayomchan

bxm40@po.cwru.edu

Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106, U.S.A.

Randall D. Beer

beer@eecs.cwru.edu

Departments of Electrical Engineering and Computer Science and of Biology, Case Western Reserve University, Cleveland, OH 44106, U.S.A.

A center-crossing recurrent neural network is one in which the null-(hyper)surfaces of each neuron intersect at their exact centers of symmetry, ensuring that each neuron's activation function is centered over the range of net inputs that it receives. We demonstrate that relative to a random initial population, seeding the initial population of an evolutionary search with center-crossing networks significantly improves both the frequency and the speed with which high-fitness oscillatory circuits evolve on a simple walking task. The improvement is especially striking at low mutation variances. Our results suggest that seeding with center-crossing networks may often be beneficial, since a wider range of dynamics is more likely to be easily accessible from a population of center-crossing networks than from a population of random networks.

1 Introduction ---

Dynamical recurrent neural networks are an ideal neural substrate for the generation and recognition of temporal patterns, but they are considerably more difficult and computationally expensive to train than feedforward networks. Two broad classes of training algorithms are typically used: recurrent extensions of traditional feedforward learning algorithms (Pearlmutter, 1990) and evolutionary algorithms (Beer & Gallagher, 1992; Cliff, Harvey, & Husbands, 1993; Nakahara & Doya, 1998; Di Paolo, 1998; Ijspeert, 2001). Due to the computational cost of these algorithms, general strategies for minimizing the computation required to achieve a given level of performance are of great practical interest. Here, we describe a general technique for improving the performance of evolutionary searches on continuous-time recurrent neural networks (CTRNNs).

In a previous analysis of the dynamical behavior of small CTRNNs, several strategies for improving their evolvability were suggested (Beer, 1995). These suggestions follow from the general dynamical structure of CTRNNs and thus are not specific to any particular application. In this article, we test the advantages of one such proposal: seeding evolutionary searches with center-crossing CTRNNs. A center-crossing CTRNN is one in which the null-(hyper)surfaces of individual neurons cross one another at their exact centers, ensuring that the range of inputs that each neuron receives is centered over the most sensitive region of its activation function (Beer, 1995). Here, we demonstrate that relative to a random initial population, seeding an evolutionary algorithm with center-crossing networks significantly improves both the reliability and speed with which high-fitness oscillatory circuits evolve on a simple walking task.

2 Center-Crossing Networks

Continuous-time recurrent neural networks are networks of model neurons of the following general form,

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^N w_{ji} \sigma(y_j + \theta_j) \quad i = 1, \dots, N \quad (2.1)$$

where y_i is the state of the i th neuron, \dot{y}_i denotes the time rate of change of this state, τ_i is the neuron's membrane time constant, w_{ji} is the strength of the connection from the j th to the i th neuron, θ_i is a bias term, and $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic output function. Although variations of this basic model neuron were studied much earlier, a restricted form of this model was popularized by Hopfield (1984) in his work on associative memories. Unlike Hopfield, we make no restrictions on the weight matrix, so that CTRNNs are capable of general dynamical behavior (Funahashi & Nakamura, 1993). The oscillatory behavior of dynamical recurrent neural networks has been studied by a number of authors, including Ermentrout and Cowan (1979), Atiya and Baldi (1989), and Blum and Wang (1992).

A center-crossing CTRNN is one in which the null-(hyper)surfaces of individual neurons intersect at their exact centers of symmetry (see Figure 1), producing an equilibrium point for which the outputs of all neurons are one-half (Beer, 1995). Since $\sigma^{-1}(1/2) = 0 = y_i + \theta_i$, this occurs when each $y_i = \theta_i$. By substituting this constraint into equation 2.1, setting that equation equal to 0 to specify the equilibrium points, and solving for θ_i , we find that the center-crossing condition is given by

$$\theta_i^* = \frac{-\sum_{j=1}^N w_{ji}}{2}. \quad (2.2)$$

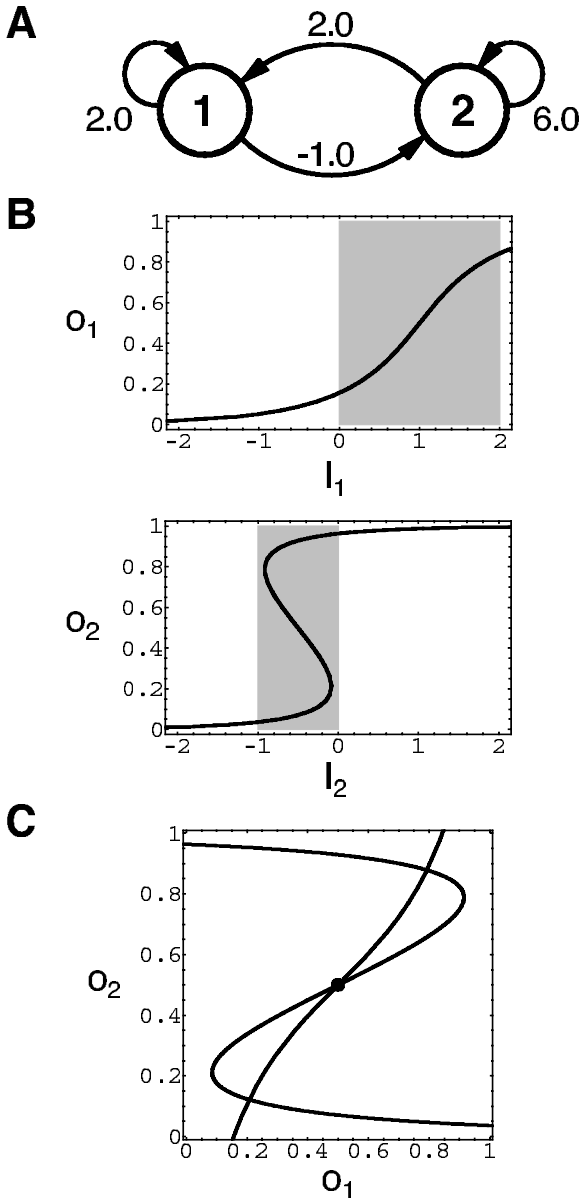


Figure 1: A simple illustration of the center-crossing condition in a two-neuron CTRNN. (A) The example circuit, with biases set as $\theta_1 = -(2 + 2)/2 = -2$ and $\theta_2 = -(6 - 1)/2 = -2.5$. (B) The steady-state input-output curve of each neuron superimposed over the range of synaptic inputs it receives from the other neuron. (C) The nullclines of the full two-neuron circuit, with the center-crossing point indicated by a black dot.

Center-crossing networks were originally studied because of their analytical tractability (Beer, 1995). However, it was conjectured that seeding the initial population of an evolutionary algorithm with center-crossing networks rather than random networks might improve the performance of evolutionary searches on CTRNNs (Beer, 1995). A center-crossing network has the property that the operating range of each neuron is centered about the most sensitive region of its activation function (see Figure 1). Due to the form of $\sigma(x)$, unless a neuron's bias is properly tuned to the range of inputs it receives, that neuron will simply saturate on or off and drop out of the dynamics. Thus, the richest dynamics should be found in the neighborhood of the center-crossing networks in parameter space, and one would expect that an evolutionary algorithm would benefit from focusing its search there.

3 Methods

We test this conjecture by studying the evolution of CTRNN central pattern generators (CPGs) for walking in a simple legged body (Beer & Gallagher, 1992). The body consists of a single leg with a joint actuated by two opposing "muscles" and a foot. Details of the model and its analysis can be found in Beer, Chiel, and Gallagher (1999). We focus here on fully interconnected five-neuron CPGs having a total of 35 parameters (5 time constants, 5 biases, and 25 connection weights). Three of these neurons are motor neurons that control the two opposing muscles and the foot, while the remaining two neurons are interneurons with no preassigned function. The model was integrated for 220 time units using the forward Euler integration method with a step size of 0.1. Fitness was evaluated as the average velocity (total forward distance covered in 220 time units divided by 220). The highest average velocity achievable is known to be 0.627 from a previous analysis of the optimal controller for this task and body model (Beer et al., 1999).

A real-valued genetic algorithm was used to evolve CTRNN parameters. A population of 100 individuals was maintained, with each individual encoded as a vector of 35 real numbers. The best individual in the old population was simply copied to the new one (elitist selection), and the remaining children were generated by mutation of selected parents. Individuals were selected for mutation using a linear rank-based method, with the best individual producing an average of 1.1 offspring. A selected parent was mutated by adding to it a random displacement vector whose direction was uniformly distributed on the 35-dimensional hypersphere and whose magnitude was a gaussian random variable with 0 mean and variance σ^2 (Bäck, 1996). Searches were run for 250 generations. Connection weights and biases were constrained to lie in the range ± 16 , and time constants were constrained to the range [0.5, 10].

In the experiments described below, we compared the performance of evolutionary searches seeded with random center-crossing networks (seeded searches) to the performance of searches with completely random initial

populations (random searches). In the random searches, the initial population was generated by randomly assigning parameter values drawn from a uniform distribution over the allowed ranges. This is the typical way in which evolutionary algorithms are initialized. In the seeded searches, the initial population was generated by randomly assigning time constants and weights, but computing biases from the weights according to the center-crossing condition, equation 2.2. Aside from this difference in the initial population, the evolutionary search proceeded in exactly the same way for both random and seeded searches.

4 Results

4.1 Seeded Searches More Reliably Evolve High-Fitness Circuits. The benefits of seeding an evolutionary search with center-crossing networks are clearly shown in Figure 2A, where we plot the mean performance of seeded (black) and random (gray) searches as the mutation variance σ^2 is varied from 0.05 to 3.0. Each point represents the mean of the average velocity achieved by the best CPG evolved in each of 100 searches. Over the entire range of σ^2 shown, the mean best performance of seeded searches exceeds that of random searches. Since the very best circuits evolved by both random and seeded searches have comparable fitness, the observed differences in mean best performance primarily reflect differences in the number of successful searches. This difference becomes especially pronounced at small mutation variance, where the mean performance of the CPGs evolved by seeded searches is nearly twice that of random searches.

The qualitative forms of these two curves are quite different. While the performance of random searches begins low and rises to a plateau with increasing σ^2 , the performance of seeded searches begins high and falls to a plateau. The form of the random curve suggests that randomly generated parameter sets generally perform quite poorly as CPGs. When mutation variance is low, it is difficult for the search to progress very far from these low-performance initial circuits. At higher mutation variances, however, more searches are able to recover from poor initial conditions and still find their way to good walking controllers.

In contrast, the form of the seeded curve suggests that random center-crossing networks are generally much better CPGs, so that low mutation variances can often fine-tune them into very fit walking controllers. Larger mutation variances can easily destroy this initial advantage, causing the performance of the seeded searches to degenerate to nearly that of the random searches. Thus, the conditions necessary for oscillation must be somewhat delicate, and they must be closely related to the center-crossing constraint.

At large mutation variance, both curves flatten into a slow decrease of mean best performance with σ^2 . The flattening is probably due to our use of elitist selection, which ensures that any progress made by mutation is retained. Nevertheless, as the mutation variance increases, children

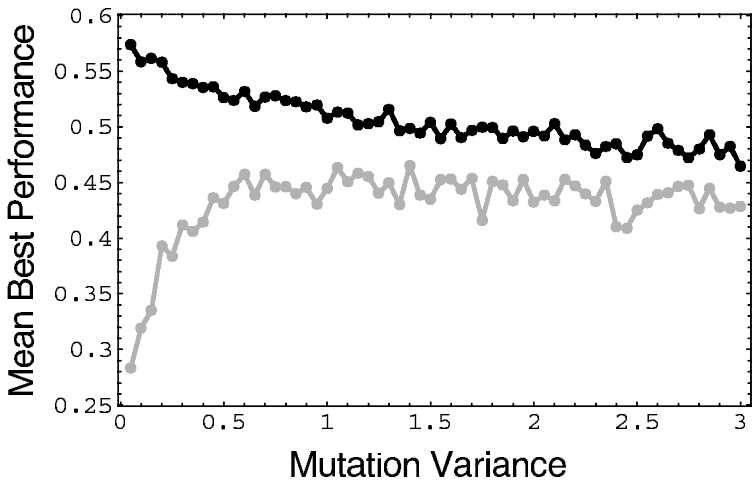
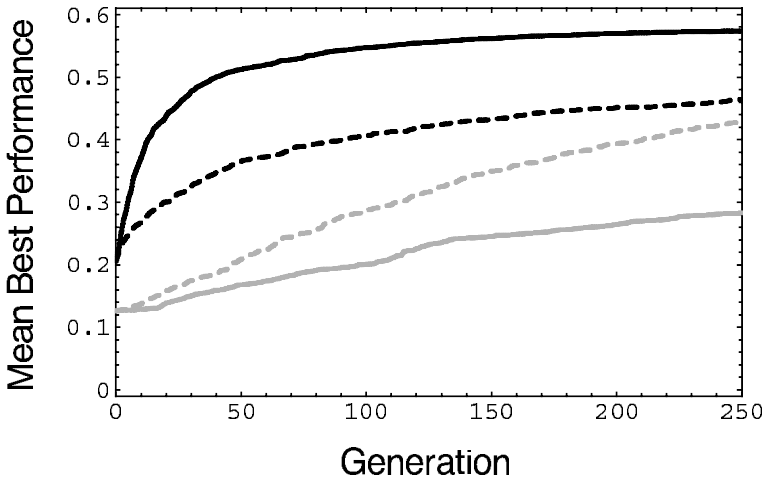
A**B**

Figure 2: Results. (A) Mean best performance as a function of mutation variance for seeded (black) and random (gray) initial populations. Each point represents the mean of 100 evolutionary searches at mutation variance intervals of 0.05, for a total of 12,000 experiments. (B) Mean best performance achieved as a function of generation number for seeded (black) and random (gray) initial populations. Results for two different mutation variances are shown: $\sigma^2 = 0.05$ (solid) and $\sigma^2 = 3.0$ (dashed). For comparison, the best fitness obtained from 25,000 completely random CTRNNs was 0.24, and the best fitness obtained from 25,000 random center-crossing CTRNNs was 0.40.

become decreasingly correlated with their parents, and eventually evolutionary search degrades to random sampling.

4.2 High-Fitness Circuits Evolve Faster in Seeded Searches. Not only do seeded searches evolve better CPGs on average, but they also evolve them faster, as shown in Figure 2B. Here, we plot the mean best performance achieved by seeded (black) and random (gray) searches as a function of generation number. Results for two different values of σ^2 are shown (solid lines are $\sigma^2 = 0.05$ and dashed lines are $\sigma^2 = 3.0$). It is clear from these plots that seeded searches achieve a given level of performance considerably faster than random searches. Of course, as is to be expected from Figure 2A, this advantage declines with increasing mutation variance. Nevertheless, even at $\sigma^2 = 3.0$, the mean performance of seeded searches begins at higher values and rises faster than for random searches.

Why do populations seeded with center-crossing circuits evolve faster? Oscillations are essential to high-performance locomotion. Since the maximum forward distance that the body model can move in one step is 27.5 (Beer et al., 1999), the best performance that a nonoscillatory CTRNN can obtain is $27.5/220 = 0.125$. Random searches often plateau near this level for long periods of time. In order to exceed this limit, CTRNNs must oscillate so as to take multiple steps during the performance evaluation period. Thus, one way to interpret the results shown in Figure 2B is that seeded searches discover oscillatory CTRNNs faster than random searches. Indeed, since the mean best performance for seeded searches is greater than 0.125 even at generation 0, oscillatory CTRNNs must often exist in the initial center-crossing population. To test this hypothesis, we generated 10,000 random center-crossing CPGs and 10,000 completely random CPGs. We found that 26.6% of the center-crossing circuits produced oscillations, while only 1.2% of the random circuits did so. Once oscillations are discovered, the evolutionary algorithm can fine-tune them into highly fit CPGs by matching the amplitude, period, and phase of the oscillation to the characteristics of the model body.

5 Conclusion

We have demonstrated that seeding an evolutionary algorithm with center-crossing networks can significantly improve both the frequency and the speed with which high-fitness central pattern generators evolve. While the benefits of center-crossing networks decrease with increasing mutation variance, searches from random initial populations never achieve either the best performance or the rate of improvement exhibited by seeded searches. Since nothing about the center-crossing condition is specific to walking, it is likely that a similar improvement would be found on any oscillatory task. In fact, all other things being equal, our results suggest that seeding evolutionary searches with center-crossing networks may always be beneficial, since a

wider range of dynamics is more likely to be easily accessible from a population of center-crossing networks than from a random population. Thus, a considerable practical benefit can be obtained from general considerations of the dynamics of recurrent neural networks. By the same reasoning, it is likely that the optimization of integrate-and-fire or conductance-based models would also benefit from initialization with networks whose synaptic weights and thresholds ensured that the neurons were centered about their most sensitive operating regions. Indeed, there is growing evidence that nerve cells actively regulate their ionic conductances so as to maintain a given level of excitability (Turrigiano, Abbott, & Marder, 1994).

Acknowledgments

We thank Hillel Chiel for his feedback on a draft of this article. This research was supported in part by grant N0014-99-1-0378 from the Office of Naval Research and grant DMI97-20309 from the National Science Foundation.

References

- Atiya, A., & Baldi, P. (1989). Oscillations and synchronization in neural networks: An exploration of the labeling hypothesis. *Int. J. Neural Systems*, *1*(2), 103–124.
- Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. New York: Oxford University Press.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, *3*, 469–509.
- Beer, R. D., Chiel, H. J., & Gallagher, J. C. (1999). Evolution and analysis of model CPGs for walking II. General principles and individual variability. *J. Computational Neuroscience*, *7*(2), 119–147.
- Beer, R. D., & Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, *1*, 91–122.
- Blum, E. K., & Wang, X. (1992). Stability of fixed points and periodic orbits and bifurcations in analog neural networks. *Neural Networks*, *5*, 577–587.
- Cliff, D., Harvey, I., & Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, *2*(1), 73–110.
- Di Paolo, E. A. (1998). An investigation into the evolution of communication. *Adaptive Behavior*, *6*(2), 285–324.
- Ermentrout, G. B., & Cowan, J. D. (1979). Temporal oscillations in neuronal nets. *J. Mathematical Biology*, *7*, 265–280.
- Funahashi, K., & Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, *6*, 801–806.
- Hopfield, J. J. (1984). Neurons with graded response properties have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, *81*, 3088–3092.
- Ijspeert, A. J. (2001). A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, *84*, 331–348.

- Nakahara, H., & Doya, K. (1998). Near-saddle-node bifurcation behavior as dynamics in working memory for goal-directed behavior. *Neural Computation*, *10*, 113–132.
- Pearlmutter, B. A. (1990). *Dynamic recurrent neural networks* (Tech. Rep. No. CMU-CS-90-196). Pittsburgh, PA: Carnegie Mellon University School of Computer Science.
- Turrigiano, G., Abbott, L. F., & Marder, E. (1994). Activity-dependent changes in the intrinsic properties of cultured neurons. *Science*, *264*, 974–976.

Received August 7, 2001; accepted March 21, 2002.