Acquiring Efficient Locomotion in a Simulated Quadruped through Evolving Random and Predefined Neural Networks

Frank Veenstra[1], Alexander Struck[1], Matthias Krauledat[1]

[1]Department of Technology and Bionics
Rhine-Waal University of Applied Sciences Kleve, Germany

Abstract. The acquisition and optimization of *dynamically stable* locomotion is important to engender fast and energy efficient locomotion in animals. Conventional optimization strategies tend to have difficulties in acquiring dynamically stable gaits in legged robots. In this paper, an evolving neural network (ENN) was implemented with the aim to optimize the locomotive behavior of a four-legged simulated robot. In the initial generation, individuals had neural networks (NNs) that were either predefined or randomly initialized. Additional investigations show that the efficiency of applying additional sensors to the simulated quadruped improved the performance of the ENN slightly. Promising results were seen in the evolutionary runs where the initial predefined NNs of the population contributed to slight movements of the limbs. This paper shows how a predefined ENNs linked to bio-inspired sensors can optimize a locomotive strategy for a simulated quadruped.

Keywords. Bio-inspired artificial intelligence; Evolving neural networks; Legged locomotion; Quadruped evolution.

1. Introduction

Evolutionary pressure has driven species of animals to develop efficient locomotive behaviors by gradually changing their morphology and locomotive control. One of the evolved locomotive strategies includes terrestrial legged locomotion that is an efficient method for animals to traverse rough terrain making it an interesting feat to apply in robotics. Most conventional optimization strategies used for acquiring locomotive control are still inept to generate efficient stable locomotion and may be improved by using additional bio-inspired methods. Optimizing legged locomotion in robots is a difficult task as efficient legged locomotion is usually dynamically stable. Locomotion is considered dynamically stable when an agent's center of mass (COM) is only temporarily above the support area of the legs during locomotion [1].

In animals, a nervous system consisting of innate (inborn) and learned (acquired) types of behavior regulates locomotion [2]. Through implementing an evolutionary algorithm that alters a neural network (NN), this paper shows that applying predefined evolving neural networks (ENNs) to simulated and actual robots is a promising bio-inspired optimization strategy for the generation of dynamically stable locomotion.

A great diversity of neuroevolutionary strategies have been developed over the past two decades [3,4,5,6,7]. Changing the topology and weights of NNs is a commonly used strategy also known as Topology and Weight Evolving Artificial Neural Networks (TWEANNs) [8]. Based on TWEANNs, optimization strategies like NEuroevolution of Augmenting Topologies (NEAT) [9,8] and Evolution of Network Symmetry and mOdularity (ENSO) [10] were developed to increase the efficiency of TWEANNs. In this paper, no frameworks of other ENNs were used but instead an ENN similar to a TWEANN was implemented to control and optimize the locomotion of a simulated quadruped. Learning methods that adapt the synapses and weights of the NN are not implemented in the ENN of this paper as fixed NNs tend to evolve quicker [11]. Although there is evidence suggesting that NEAT and ENSO are more efficient strategies to apply compared to regular TWEANNs [9,10], these methods are not used as the aim of this paper is to analyze acquisition of efficient locomotion based on different initial NN states of the population. Comparative studies comparing the effectiveness of the devised ENN with other neuroelutionary strategies is out of the scope of this paper but can be done in future investigations.

Central Pattern Generators (CPGs) are also often implemented for mimicking animal locomotion [12]. In animals, CPGs provide rhythmic activation of muscles and do not necessarily require any sensory input to function [13]. Various strategies mimic the functionality of CPGs for the acquisition of gaits: Hopf oscillators, [14,15,16,17], cyclic genetic algorithms (CGAs) [18], continuous-time recurrent neural networks (CTRNNs) [19], compositional pattern producing networks (CPPNs) [20,21], and hypercube-based neuroevolution of augmenting topologies (hyperNEAT) [21]. CPGs applied to the ENN of this paper are simply defined by neurons that activate and deactivate based on an evolvable timer and outputs of the NN.

The aim of this paper is to apply a several predefined NNs and bio-inspired sensors to a simulated quadruped as neuroevolutionary optimization strategies in order to evolve efficient locomotive behavior. Three types of predefined NNs (NNs that were preprogrammed to have a certain morphology and thereby a distinct neural activation pattern) were used to initialize various populations. Bongard [22] has shown that initializing populations with behavior of robots that were formerly evolved using a simpler physical morphology led to more rapid acquisition of robust locomotive behavior compared to evolving the robot behavior of the more complex robot without implementing the evolved behaviors of simpler robots. Similar results are expected when rough estimations of simple predefined NNs are used to initialize a population.

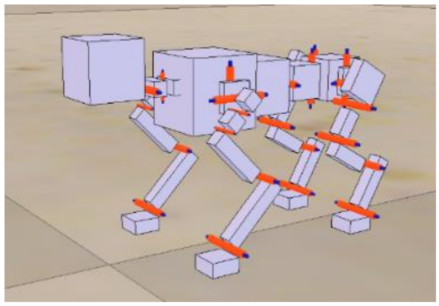

Figure 1. The 3D model of the simulated quadruped

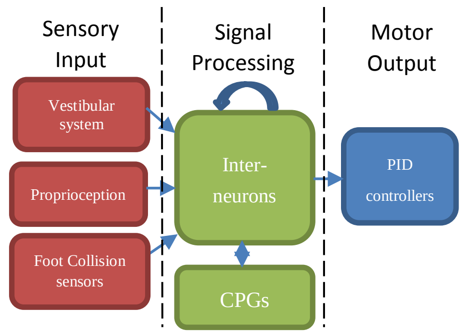

Figure 2. Overview of the artificial neural network. Arrows indicate where to neuron connectivity. Note that the sensory input and the CPGs can also directly connect to the Motor output layer.

2. Methods

The 3D robot model (Figure 1) was simulated in the robotics platform "Virtual Robot Experimentation Platform" (V-REP) [23]. The 3D model of the quadruped is based on feline morphology as cat-like quadrupeds are among the fastest animals alive. The length of the cat is around 0.5 meters. Spring-like properties were able to arise as PID controllers regulate the joint actuation. Notably, two spine joints mimic properties of a flexible spine, which is a valuable feature for the high performance locomotion of the Cheetah [16,24,25]. The open-source Bullet dynamics physics engine was used to simulate the physics of the simulation. Based on feline morphology, the maximum allowed angles of all 28 joints ranged from 30 to 180 degrees. Four types of sensors were applied to give the simulated quadruped some bio-inspired feedback. These sensors include proprioception, tactile feet sensors, an abstraction of the vestibular system (the balance organ), and CPGs.

Unlike feedforward perceptrons [26], the applied NN's hidden layer is recurrent. The applied NN consisted of a variable number of input neurons (depending on the sensors used) 150 interneurons in the hidden layer and 96 output neurons connected to PID controllers and CPGs (to alter the CPGs timers and thus altering activation speed). The equations below (Equation 1-5) explain how each layer in the NN is updated. The activation levels of the sensory, inter- and motor neurons are defined by B_i, C_j and D_k respectively. A_i and E_k define the sensory input and motor output respectively. There are four types of weights for each type of possible connection: weights from sensory neurons connected to interneurons (ϕ); weights from interneurons connected to interneurons (χ); weights from sensory neurons connected to motor neurons (ψ); weights from interneurons connected to motor neurons (ω). α_j represents the acquired activation levels of the interneurons. If α_j passes the value of the corresponding threshold level θ_j, the interneuron is activated. Finally, the decay factor δ decreases the acquired activation levels of both interneurons and motor neurons. In all equations, the operator ":=" represents an update of the left hand side variable with the term on the right hand side, as it is performed in each calculation for a new frame.

- Equation 1: Sensory neurons are always activated by sensory input and the output of these sensory neurons is transformed into the activation level (B_i) of the i^th^ sensory neuron. A_i is the output value of the sensor connected to the i^th^ sensory neuron.
- Equation 2: The acquired activation level (α_j) of the j^th^ interneuron is based on the weights (ϕ_{ji}) of the connected sensory neurons and the weights (χ_{jl}) of other connected interneurons. A decay factor, δ, decreases the acquired activation level of the neuron over time so that continuously activated neurons limit their maximum activation level.
- Equation 3: The interneurons are activated if the acquired activation level α_j of the j^th^ interneuron is higher than the threshold θ_j. Θ represents the Heaviside step function, i.e., $C_j = 0$ if $\alpha_j < \theta_j$ and $C_j = 1$ otherwise.
- Equation 4: The activation level of the motor neurons (D_k) is calculated similar to the acquired activation level of the interneurons. ψ_{jk} represents the weight of the j^th^ sensory neuron connected to the k^th^ motor neuron, ω_{kj} represents the weight of the j^th^ interneuron connected to the k^th^ motor neuron. However, motor neurons are, like sensory neurons, always active, meaning no threshold function needs to be applied. The decay factor, δ, limits the activation levels of the output neurons.
- Equation 5: The final equation describes how the factor (σ_k) scales the k^th^ motor neuron's activation level (D_j) to a motor output value (E_k).

$$B_i := A_i \tag{1}$$

$$\alpha_j := \frac{1}{\delta}(\alpha_j + \textstyle\sum_{i=1}^{m} \phi_{ji} B_i + \sum_{l=1}^{n} \chi_{jl} C_l) \tag{2}$$

$$C_j := \Theta(\alpha_j - \theta_j) \tag{3}$$

$$D_k := \frac{1}{\delta}(D_k + \textstyle\sum_{i=1}^{m} \psi_{ki} B_i + \sum_{j=1}^{n} \omega_{kj} C_j) \tag{4}$$

$$E_k := D_k * \sigma_k \tag{5}$$

The evolutionary algorithm alters the NN's genotype through specific mutations and a function mimicking chromosomal crossover [27]. The hereditary information of the quadruped's NN is stored in arrays containing all parameters of the NN. Similar to TWEANNs, the mutations altered parameters such as the connections of individual neurons, the weights attributed to these connections and the threshold values that need to be surpassed before an interneuron is activated. The mutation rate is variable depending on one of three types of mutations that can occur to enable both large and subtle changes of the NN. The crossover function combines two NNs of two individuals in a population by combining their interneurons before mutations occur (25% probability). The genome of the quadruped stores the values of each neuron in arrays that combine through crossover based on the specified assigned neuron number in the array. A maximum of five crossover events could take place to create a new interneuron layer for the offspring. Each neuron's output is connected to at most 10 other neurons restricting the network's topology. This limit is set as most neurons only have one or a few axons [27].

The population size used was 20 and parents were randomly chosen to produce offspring asexually with the potential for crossover. Newly formed individuals only replaced other individuals if their fitness was higher to the fitness value of a randomly chosen individual of the population. Subsequent generations thus always performed equally well or better than the previous ones. The fitness value was measured by calculating the distance the quadruped had moved in a forward direction in five seconds of simulation time.

Three predefined morphologies of NNs were used to initialize the population of different evolutionary runs. The three predefined morphologies consisted of two manually defined and one randomly defined morphology. One of the manually defined morphologies produced a behavior where the joints were kept stationary (predefined stationary) while the other produced slight movements in eight joints resembling a precursor of a two-beat diagonal trot gait (predefined walking). An additional fitness function was applied to the simulations running the predefined NNs to speed up the evolutionary runs. This additional fitness function reduced overall simulation time by resetting the simulation in occurrences of head to floor collisions. Simulation runs for type of initial NN population were performed both with enabled and disabled sensory input to evaluate the impact of sensors on the effectiveness of the ENN. For each experiment, 10 deterministic simulations ran each using a different random seed.

3. Results

Differences in the progression of each type of ENN was notable between different evolutionary runs (Figure 3). The evolutionary run of the predefined walking NNs using sensors developed the best locomotive strategy for the simulated quadruped in generation 1000 as its final generation moved significantly further ($p < 0.05$) than the other types of evolutionary runs. The evolution of the population of quadrupeds using the predefined stationary NN performed similar with and without using sensors. There were slight but significant differences between the evolutionary runs the randomly initialized NNs that did and did not use sensors ($p < 0.05$). No significant differences were seen between the predefined stationary NNs using sensors and not using sensors. Evolving the best evolved individual from generation 1000 of the predefined walking run further for an additional 5000 generations showed better performing NNs without any dramatic changes to the phenotype of the behavior [28].

All simulations initialized with a population of individuals with predefined walking NNs evolved locomotive strategies that moved up to three times as far as the other evolved NNs. From these evolved behaviors, some evolved a walking motion, others included slight jumping movements and the best simulation evolved a crawling motion that made the quadruped move by seemingly only using its two forelimbs (Figure 4). The ENN also evolved useful strategies when not using sensors although the evolutionary progression was generally slower. The predefined stationary NNs did not evolve effective locomotion but rather evolved motions wherein the individual rolls on its side preventing the head from colliding with the floor. The randomly initialized

NNs did not evolve any efficient locomotive strategies either as they evolved behaviors consisting mostly of falling and rolling forward, and twitching.

4. Discussion

From evolving the different initial population states, the predefined walking NNs evolved effective locomotion the quickest (Figure 3). All other evolutionary runs evolved behaviors in which the locomotive phenotype of the fittest individual consisted of either falling or rolling forward. These results suggest that initializing a population with individuals displaying slight limb movements, resembling a desired movement, greatly accelerates the evolution of the simulated quadrupeds' NNs. The best performing individual did not evolve a legitimate dynamically stable gait as its hind-legs were not noticeable but were instead dragged across the floor (Figure 4). The application of sensors did slightly increase the speed at which desired behaviors in the randomly initialized and predefined sensing evolutionary runs.

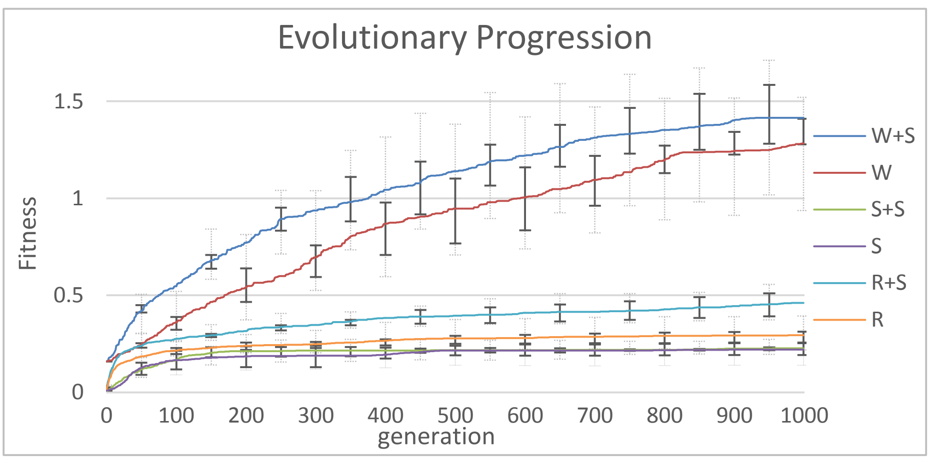

Figure 3: The graph shows the median of the performance of the ENN after 1000 generations. The fitness value indicates the average movement of the population of simulated quadrupeds. The six lines represent the median value of 10 evolutionary runs with different initial states. The initial states of each run were: predefined walking NN using sensors (W+S), predefined walking NN not using sensors (W): predefined stationary NN using sensors (S+S), predefined stationary NN without using sensors (S) random NN using sensors (R+S) and random NN without using sensors (R). Black error bars represent the 1st and 3rd quartiles and the dotted grey error bars represent the minimum and maximum values.

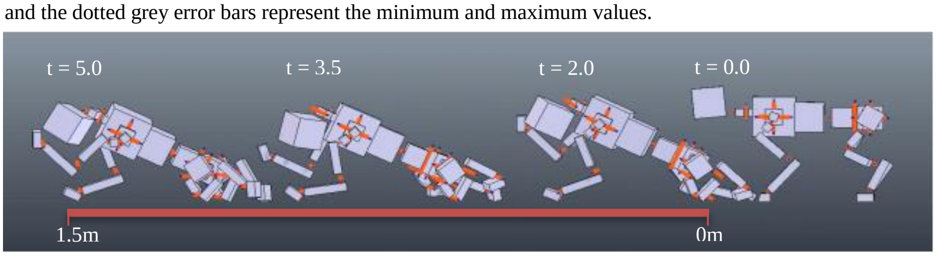

Figure 4. The locomotion of the best evolved NN from generation 1000. t represents the time it took (in seconds) for the quadruped to get to the particular location.

The phenotypic change of an increase in fitness did not reveal any drastic changes indicating that the evolved phenotypes were rather enhancements of resembling previous phenotypes. Although the evolution of randomly initialized NNs did not lead to the acquisition of efficient types of locomotion, the average fitness value was ever increasing nonetheless. Even after 5000 generations, populations still evolved into a better performing population [28]. However, evolving randomly initialized NNs with the neuroevolution strategy described in this paper is a lot less efficient than evolving predefined walking NNs.

Despite the limited amount of evolutionary runs and simulation time, promising locomotive patterns arose when evolving predefined walking NNs. It may be interesting to see what behavior arises when more generations, larger populations or island populations are implemented. Moreover, competitive co-evolution [29,30], morphological change [22], additional learning algorithms, genetic drift [31], evolving the evolvability of agents [32], the application of incremental evolutionary methods [33] are a few features that may enhance the performance of the ENN for acquiring dynamically stable locomotion. As tendons play a huge role for animals to achieve effective locomotion through the reuse of kinetic energy [34] applying abstractions of tendons may prove useful.

The ENN presented in this paper is able to evolve dynamically stable gaits in simulated quadrupeds the quickest when the initial population consists of predefined NNs. Future research could indicate whether standardized predefined or evolved NNs could be used for the rapid acquisition of efficient locomotive behavior in different types of simulated and actual robots.

5. Conclusion

The designed ENN evolved effective locomotive gaits when predefined walking NNs were used in the initial population. The ENN did not evolve particularly efficient locomotion when using other initial states and evolved less efficient without the implementation of sensors. The results thus indicate that predefining NNs greatly increases the speed of the neuroevolutionary optimization processes for the acquisition of dynamically stable gaits. Through improving the presented ENN and comparing it to other neuroevolution strategies, the ENN discussed in this paper may serve as a promising bio-inspired framework for the acquisition of dynamically stable locomotive gaits in simulated robots.

References

1. Luksch, T.: Human-like Control of Dynamically Walking Bipedal Robots. (2009)
2. Bolhuis, J., Giraldeau, L.: The Behavior of Animals: mechanisms function and evolution. Blackwell Publishing (2005)
3. Sims, K.: Evolving Virtual Creatures. Computer Graphics SIGGRAPH ‘94 Proceedings, 15-24 (1994)
4. Han, M., Fan, J., Han, B.: Evolving feedforward neural networks. International Joint Conference on Neural Networks, 1083-1089 (2009)
5. Angeline, P., Saunders, G., Pollack, j.: An Evolutionary Algorithm that Constructs Recurrent Neural Networks. IEEE Transactions on Neural Networks 5, 54-65 (1993)
6. Dasgupta, D., McGregor, D.: Designing Application-Specific Neural Networks using the Structured Genetic Algorithm. Combinations of Genetic Algorithms and Neural Networks, 87-96 (1992)

7. Auerbach, J., Bongard, J.: On the Relationship Between Environmental and Morphological Complexity in Evolved Robots. Proceedings of the 14th annual conference on Genetic and evolutionary computation 14, 521-528 (2012)
8. Stanley, K., Miikulainen, R.: Efficient Evolution of Neural Network Topologies. Proceedings of the 2002 Congress on Evolutionary Computation (2002)
9. Stanley, K., Miikkulainen, R.: Evolving Neural Networks through Augmenting Topologies. Evolutionary Computation 10(2), 99 - 127 (2002)
10. Valsalam, V., Miikkulainen, R.: Evolving Symmetry for Modular System Design. IEEE Transactions on Evolutionary Computation 15(3), 368-386 (2011)
11. Stanley, K., Bryant, B., Miikkulainen, R.: Evolving Adaptive Neural Networks with and without Adaptive Synapses. Proceedings of the 2003 IEEE Congress on Evolutionary Computation (2003)
12. Still, S., Hepp, K., Douglas, R.: Neuromorphic Walking Gait Control. IEEE Transactions on Neural Networks 17, 496-508 (2006)
13. Rossignol, S.: Plasticity of connections underlying locomotor recovery after central and/or peripheral lesions in the adult mammals. Phil. Trans. R. Soc. B 361, 1647-1671 (2006)
14. Barron-Zambrano, J., Torres-Huitzil, C.: Two-phase GA parameter tunning method of CPGs for quadruped gaits. Proceedings of International Joint Conference on Neural Networks, 1767-1774 (2011)
15. Wu, X., Shao, X., Wang, W.: Gait Planning of Crossing Planar Obstacles for A Quadruped Robot. Robotics and Biomimetics (ROBIO), 692-697 (2013)
16. Kani, M., Ahmadabadi, M.: Comparing effects of rigid, flexible, and actuated series-elastic spines on bounding gait of quadruped robots. First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), 282-287 (2013)
17. Ajallooeian, M., Puya, S., Sproewitz, A., Ijspeert, A.: Central Pattern Generators Augmented with Virtual Model Control for Quadruped Rough Terrain Locomotion. IEEE International Conference on Robotics and Automation (ICRA), 3321-3328 (2013)
18. Parker, G., Tarimo, W., Cantor, M.: Quadruped Gait Learning Using Cyclic Genetic Algorithms. IEEE Congress on Evolutionary Computation (CEC), 1529-1534 (2011)
19. Beer, R.: On the dynamics of small continuous-time recurrent neural networks. Adaptive Behavior 3(4), 469-509 (1995)
20. Stanley, K.: Compositional Pattern Producing Networks: A Novel Abstraction of Development. Genetic Programming and Evolvable Machines Special Issue on Developmental Systems 8(2), 131-162 (2007)
21. Clune, J., Beckmann, B., Ofria, C., Pennock, R.: Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding. IEEE Congress on Evolutionary Computation, 2764 - 2771 (2009)
22. Bongard, J.: Morphological change in machines accelerates the evolution of robust behavior. Proc.Natl.Acad.Sci.U.S.A. 108, 1234–1239.doi:10.1073/pnas. (2010)
23. Rohmer, E., Singh, S., Freese, M.: V-REP: a Versatile and Scalable Robot Simulation Framework. In : Proc. of The International Conference on Intelligent Robots and Systems (IROS), Tokyo, pp.1321 - 1326 (2013)
24. Khoramshahi, M., Sprowitz, A., Tuleu, A., Ahmadabadi, M., Ijspeert, A.: Benefits of an Active Spine Supported Bounding Locomotion With a Small Compliant Quadruped Robot. IEEE International Conference on Robotics and Automation (ICRA), 3329-3334 (2013)
25. Folkertsma, G., Kim, S., Stramigioli, S.: Parallel stiffness in a bounding quadruped with flexible spine. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2210-2215 (2012)
26. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review 65, 386-408 (1958)
27. Purves, D., Augustine, G., Fitzpatrick, D., Hall, W., LaMantia, A.-S., McNamara, J., Williams, S.: Neuroscience 3rd edn. Sinauer Associates, Inc, Massachusetts (2004)
28. Veenstra, F.: Evolving Dynamically Stable Locomotion in a Simulated Quadruped using Bio-Inspired Artificial Intelligence, MSc. thesis. Rhine-Waal University of Applied Sciences, Kleve, Germany (2014)
29. Nolfi, S., Floreano, D.: Evolutionary Robotics. The MIT Press (2000)
30. Sims, K.: Evolving 3D Morphology and Behavior by Competition. Artificial Life IV Proceedings, 28-39 (1994)
31. Campbell, N., Reece, J.: Biology Eight edn. Pearson Education, San Francisco (2008)
32. Crombach, A., Hogeweg, P.: Evolution of Evolvability in Gene Regulatory Networks. Theoretical Biology and Bioinformatics Group 4(7), 1-13 (2008)
33. Bongard, J.: Behavior Chaining: Incremental Behavior Integration for Evolutionary Robotics. Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (2008)
34. Biewener, A.: Animal Locomotion. Oxford University Press., New York (2003)