Evolved Controllers for Simulated Locomotion

Brian F. Allen and Petros Faloutsos

University of California, Los Angeles CA 90024, USA
vector@cs.ucla.edu

Abstract. We present a system for automatically evolving neural networks as physics-based locomotion controllers for humanoid characters. Our approach provides two key features: (a) the topology of the neural network controller gradually grows in size to allow increasingly complex behavior, and (b) the evolutionary process requires only the physical properties of the character model and a simple fitness function. No *a priori* knowledge of the appropriate cycles or patterns of motion is needed.

1 Introduction

Natural character motion is closely tied to the character's physical makeup and environment. As interactive entertainment rapidly adopts physical simulation as a core aspect, the problem of physical control for characters comes to the fore.

There are three main hurdles to the adoption of physical controllers in real-time systems. First, controllers, even for seemingly simple motions, have proven surprisingly difficult and time-consuming to engineer by-hand. Second, real-time environments allow for limited computational resources, and finally, of prime consideration for animation and games, the resulting motion should appear fluid and natural.

This work takes a significant step toward a long-standing goal in the graphics literature: "An ideal automated synthesis system would be able to design an efficient locomotion controller given only the mechanical structure of the creature (including actuators) and no other *a priori* information" [1]. By forgoing acquired real-world data (e.g., motion capture), our method becomes applicable to a much wider variety of characters. Acquiring motion data from animals at extreme scales, such as an ant or an elephant, is problematic and expensive. Further, imaginary characters, although quite common to game settings, are generally unavailable for motion capture sessions. We evaluate the method described in this paper using human-like characters because we believe human motion is generally held to the highest standard of quality of motion, and because bipedal locomotion is among the more difficult gaits to control.

Our approach creates controllers for physically simulated characters in a biologically inspired manner: by evolving networks of connected neurons. The resulting controllers are consistent with physical laws and provide fluid motion. Changes to the characters' size and shape results in changes to the motion trajectories. A sequence of frames from an animation of five morphologically distinct characters is shown in figure 1. The animation begins with all characters standing upright, feet together, and shows each character initiating a walk cycle.

A. Egges, R. Geraerts, and M. Overmars (Eds.): MIG 2009, LNCS 5884, pp. 219–230, 2009.

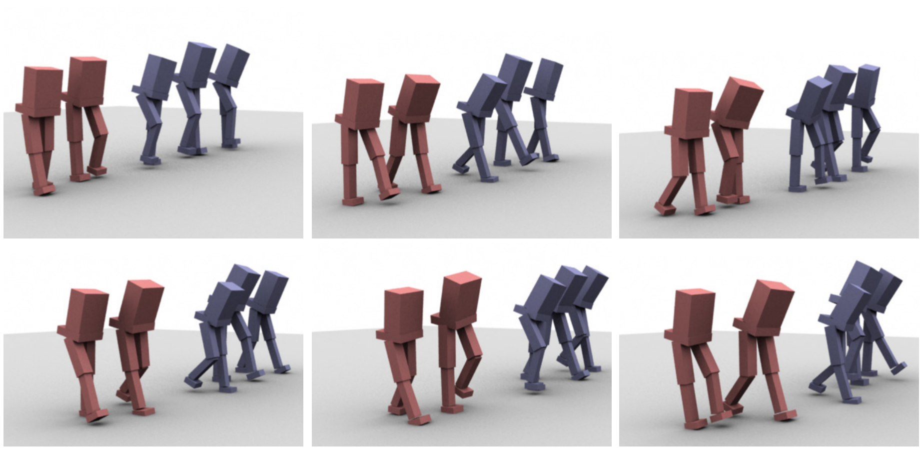

Fig. 1. Five human-sized characters with uniquely evolved walking controllers

2 Related Work

Controller design for physics-based articulated systems has been approached from a variety of perspectives. Several researchers have engineered controllers manually to perform a variety of ballistic or high-energy motions (such as running, vaulting and diving) [2, 3, 4]. Since low-energy motions, like walking, are poorly constrained by ballistics or energy minimization alone, controllers can rely on potentially unnatural or contrived higher-level constraints [5], or on generalizations from motion captured data [6]. Alternatively, the control problem can be simplified at the expense of realism in the physical simulation [7].

Our evolutionary approach is, at the core, an optimization problem. The space-time constraint approach also frames the problem of physically valid control in terms of search– minimizing a function of the energy expended by the virtual character [8,9]. This approach has been quite successful for finding motion paths in high-energy or ballistic domains, such as jumping or diving, however, low-energy motions with many physically correct solutions tend to be difficult to optimize and often yield unnatural motion.

More relevant to our approach are several attempts to produce locomotion cont rollers through stochastic optimization and genetic algorithms. Two early approaches were [10] and [11]. Sims referred to his control networks as "neural", but used a variety of functions favorable to cyclical output, such as sinusoids, in a manner more similar to genetic programming [12] than neuroevolution. However, as with our approach, evolving the topology of the control system was a key element in obtaining the complex behavior seen in Sims' virtual creatures. Gritz [13] also used genetic programming to create controllers, though with a focus on providing keyframe-like control of the resulting motion.

Laszlo [14] addressed the more constrained and difficult problem of bipedal walking. Their approach was to use limit-cycle control to reduce the problem to a search of parameters that would stabilize a walk cycle. This approach yielded stable and controllable walks for two different character models, one humanoid and one non-humanoid. However, the resulting motion was clearly robotic.

More recently, evolved neural networks have become a central approach to control problems in the nascent field of evolutionary robotics [15]. For example, [16, 17] use fixed-topology neural networks without sensory inputs to evolve biomimetic "central pattern generators" (CPG). The evolved artificial CPG's generate open-loop patterns that drive joint angles of a bipedal model to show very "natural motion" through a walk cycle. In these works, continuous-time neurons are arranged and connected in a human-designed, fixed topology designed to produce cyclical patterns suitable for a locomotion controller. The connection weights and neural parameters are evolved to generate walking motions. The design of the fixed topologies used in [17] were based on recent findings in neurobiology [18]. However, in the absence of guiding biological knowledge, choosing the best fixed topology for a given problem is difficult, sometimes requiring "expert experience and a tedious trial-and-error process" [19].

In contrast, our approach builds on recent advances in the evolution of the neural topology [20], in addition to evolving the connection weights. This both relieves the controller designer of the task of choosing the proper topology, and drastically expands the range of possible behaviors. This is an important advance in the pursuit of a generic controller-creator.

3 Overview of Our Approach

We propose an evolutionary approach that produces neural networks that are used to control physically simulated, humanoid characters. In the context of evolutionary processes, it is useful to appropriate analogous terms from biology. A *genome* in our system is information transmitted from parent to offspring, and is implemented as a weighted, directed graph. This graph is used to construct a neural network. The mapping of a genome to a neural network is the encoding scheme. We employ a simple, direct encoding: genome nodes become neurons and edges become interneuron connections.

A collection of such genomes makes up the population that undergoes an evolutionary process. Each genome is evaluated, and the best-performing genomes are then either duplicated with slight modification (*mutation*) or combined with other successful genomes to produce new genomes (*cross-over*). The genomes resulting from reproduction form the population of the next generation. This process is analogous to biological evolution and, over the course of many evaluate-select-reproduce generations, genomes better suited to performing the evaluation task are likely to emerge.

To perform the generational selection, each genome is evaluated by how well its generated neural network performs as the controller of a virtual character.

4 Artificial Neural Networks

Artificial neural networks(ANN's) are composed of (a) neurons with a single scalar internal state y, referred to as the neuron's activation level, and (b) weighted, directed connections between the neurons. The structure of the connections is known as the network's topology. Though simple, the ANN model is able to exhibit complex behavior due to the non-linear manner in which each neuron aggregates the inputs from its incoming connections.

Each neuron is updated once per cycle by linearly combining its incoming connections' activations and applying the activation function, $\sigma(x)$, to that sum to determine its activation level for the next cycle (as in Equation 1).

$$y_i = \sigma\left(\sum_{j=1}^{N} w_{ji} y_j\right), \qquad \sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

where σ is the sigmoid function and w_{ji} the weight of the connection from the jth neuron to the ith.

To achieve state and time dependent behavior, an artificial neural network must rely on cycles in its topology, in a manner analogous to cross-linked XOR gates forming an electronic flip-flop. Note that in the sigmoidal neurons used here, such cycles are the only way an ANN can store state internally. To manage complex timing-based behavior, as is needed for a motion controller, an ANN generally requires a complex topology of interneuron connections.

5 Neuroevolution

Evolving a fixed-topology neural network involves mapping the scalar connection weights to a one-dimensional vector. If the topology of the network is fixed, then the correspondence between position in the vector and the neural connection can be fixed. Then the evolutionary algorithm need only manipulate a fixed-length vector of scalars. To our knowledge, this general approach has been used by all evolved neural network motion controllers for 3D bipeds (e.g., [16, 17, 21, 22, 23, 24]).

Although this traditional approach has the advantage of simplicity, using a fixed topology neural network categorically limits the type of behavior that can be exhibited (e.g., a network lacking recurrent connections cannot maintain memory). In contrast, by allowing the network to evolve its structure as well as the connection weights, the space of possible solutions is unrestrained. Our approach evolves neural topology using a version of the NEAT (NeuroEvolution of Augmenting Topologies) algorithm, described in detail in [25].

Unfortunately, evolving neural topology is not a panacea. Direct encoding from genetic representation to neural network topology may have scalability problems as the size of the graph grows [26]. Several indirect encodings (i.e., a genetic representation that requires some procedure to produce the neural topology) have been proposed, such as cellular encoding [27], to alleviate these scalability concerns, though they are not without problems [28]. In the absence of either biological analogue to or accepted practice of an indirect encoding scheme, our approach uses direct encoding.

5.1 Mutation

Mutations to the directed-graph genotype are of two kinds: structural and non-structural. The latter alter the weight of a connection or a parameter of a neuron (see Section 4). Small alterations ($\mu = 0$, $\sigma^2 = 0.1$) are most likely, though occasionally entirely new values are chosen (at random uniformly in the range $(-0.1, 0.1)$) as replacements. In an effort to bias mutational changes toward newer structures, parameters that are relatively new features in the genome are more likely to be replaced than older ones.

Two types of structural mutation are allowed: splitting an existing edge into two edges and a node, and adding a directed edge between two previously unconnected nodes, or between a node and itself. All structural mutations increase the size of the genome, so any given genetic linage is monotonically increasing.

5.2 Crossover Using Historical Markers

The NEAT algorithm labels each node and edge in a genome with a unique *historical marker*. During reproduction, these markers are preserved and passed to the offspring. During sexual reproduction, these historical markers are used to determine genetic homology. The guiding assumption is that genes with the same historical origin (and therefore the same historical markers) will perform the same function in the phenotype. Although this is, in essence, an ad hoc approach whose assumption can, and may often, fail, it has been shown to be the best known neuro-evolutionary method for simple benchmark control problems such as double-pole balancing and predator-prey simulations [29].

5.3 Speciation to Protect Innovation

In addition to enabling a more productive cross-over operation, NEAT-style historical markers can also be used to estimate the chance of mating success between individuals. A distance metric $\delta = \frac{G}{N^k} + cW$ groups the individuals of a population into species, allowing sexual reproduction within a species containing similar individuals. In this expression, G is the number of genes without a corresponding historical marker in the genes of the other parent. The constant c is a normalization factor based on the magnitude of connection weights. N is the number of genes in the larger genome, and k is a term allowing a scaling of the effect of normalization based on the number of genes N. Such speciation greatly improves the likelihood that the next generation will be viable.

Note that previous implementations of the NEAT algorithm have used $k = 0$, while our implementation uses $k = 1$. The effect of $k = 1$ is to measure genomic distance by the ratio of differing genes to the total number of genes, in contrast with $k = 0$, which considers the absolute number. Comparing the ratio, rather than absolute number better supports genomic distance calculations for large networks.

New *structural* innovations will likely differ significantly (as measured by δ) from existing genomes, meriting classification in a new species. Individuals only compete directly with other members of their own species, providing protection to new innovations until they have time to optimize their structure. This process is further assisted by giving new species an artificial fitness bonus of 20% for their first few generations.

6 Neural Networks for Control

Artificial neural networks (ANN's) have two attributes that suggest their applicability to dynamic controllers. First, for even large feedback-based tasks, neural networks require little memory and can be efficiently evaluated. Second, neural networks produce output behaviors that tend to be smooth and natural. For animation applications (as opposed to, e.g., robotics), the quality of the resulting motion is of great importance. Many approaches to physical animation control that originated in robotics have yielded stable motion, but have lacked fluidity and naturalness. Although admittedly difficult to quantify, evolved neural networks tend to yield smooth, natural-looking motion, especially in comparison with techniques based on discrete state-spaces.

7 Anthropomorphic Character Model

Our evolutionary system is not provided with any information specifying the particular gait to use. This approach improves generality, but means the character morphology plays an significant role in the quality and kind of resulting locomotive gait. Since we are interested in controllers that provide human-like motion, we use character models whose physical parameters mimic human morphology. In addition, the generality of the approach is an important measure of success. To test the generality of our method in finding human-style gaits, we use a sample of humanoid character models from the normal human range of height, weight and bi-iliac (hip) breadth (BIB) covering the 5^{th} to 95^{th} percentile of each men and women, which encompasses the variation of slightly less than 95% of the total population.

Body segment sizes and weights are scaled linearly according to the three measures (height, weight and BIB) obtained from anthropometric data aggregated from over 30,000 individuals [30]. The relative proportions of segment sizes and weights are shown in Table 2.

7.1 Sensors

The controller is provided a set of eleven sensory inputs. Proprioceptive sensors from each actuated joint angle are provided, as well as haptic sensors for each foot that register -1 when the foot is touching the ground plane, and $+1$ at any other time. The X and Z components (ranging from -1 to $+1$) of the torso's normalized up-vector are also given.

7.2 Actuation

The controller specifies, at each control time-step, the target angle the joint will be driven toward. The corresponding actuator applies torque to drive the joint to the desired angle θ_d. The output signal of the controller ranges within $[0, 1]$, of which the central 0.8 range is linearly scaled to the full range of the joint, leaving 0.1 units at each extreme clamped to 0.0 and 1.0, respectively. The applied torque τ is computed from the total moment of inertia at the joint I using a proportional-derivative (PD) controller as

$$\tau = I(k_p(\theta - \theta_d) - k_d\dot{\theta}). \tag{2}$$

Table 1. The allowed ranges for each joint of the agent. The Control column indicates if the joint's target angle is set by the controller. If not, the joint applies torque to center itself in its range.

	Range		Neural Control	
Joint, axis	High	Low	Walk	Balance
Spine, transverse	$-\frac{\pi}{12}$	$\frac{\pi}{12}$	Yes	No
Spine, coronal	$-\frac{\pi}{16}$	$\frac{\pi}{16}$	No	No
Spine, sagittal	$-\frac{\pi}{16}$	$\frac{\pi}{16}$	No	No
Hip, transverse	0	0	No	No
Hip, coronal	$-\frac{\pi}{48}$	$\frac{\pi}{24}$	Yes	Yes
Hip, sagittal	$-\frac{\pi}{16}$	$\frac{\pi}{16}$	Yes	Yes
Knee	0	$\frac{3\pi}{4}$	Yes	Yes
Ankle, transverse	$-\frac{\pi}{12}$	$\frac{\pi}{12}$	No	No
Ankle, coronal	$-\frac{\pi}{12}$	$\frac{\pi}{12}$	No	Yes
Ankle, sagittal	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	No	Yes

Table 2. The distribution of the physical proportions of the agent's body. The mass proportion of bilaterally repeated segments shows the summed mass proportion of both sides.

Body Segment	Height	Mass
Trunk	31.2%	51.6%
Waist	6.2%	10.3%
Thigh (2)	31.0%	19.3%
Shank (2)	25.3%	9.3%
Foot (2)	6.2%	9.3%

7.3 Bilateral Symmetry

The character models used are bilaterally symmetric both in their body shape and in their neural controllers. Using simple, fixed-topology reactive neural controllers, [21] demonstrated that two identical, uncoupled networks could be used, instead of one monolithic neural network, to decrease the size of the evolutionary search space. Our system uses a similar approach by building two identical neural networks from a single evolved system.

Human locomotion is approximately bilaterally symmetric (although asymmetry has been shown to be important to human perception of motion as realistic [31]). It is reasonable then to use the same controller duplicated to each side of the agent. This approach was suggested for future work in [16] and [22] used simple, fixed-topology reactive neural controllers to demonstrate that two identical, uncoupled networks could successfully decrease the size of the evolutionary search space compared to evolving a single, monolithic controller.

In this work, we evolve a single controller genome, which is then used to build two initially identical neural networks, one for each side. The outputs of each network drive their side's actuated joints, with the central waist joint taken as the average of the two corresponding output nodes' activations. The inputs of each network are likewise set from per-side information.

Each controller is given two bias nodes, one has a constant activation of $+1$, while the other has an activation of $+1$ for the right controller and -1 for the left. This per-side bias (along with asymmetric sensory input) allows for asymmetrical behavior. In addition, two haptic foot-contact sensors yield $+1$ for the same-side foot, and -1 for the opposite foot.

8 Implementation

Evolutionary runs use a population of 512 individual genomes, which are clustered into thirty species based on the genetic similarity metric δ (see Section 5.3). Each generation, every genome is evaluated by creating two neural networks, each set to control one side of the character.

These neural networks are supplied with sensory data and updated once every 0.07 s of simulated time. This delay was chosen to be within the observed range of spinal reflex response times in humans [32]. Additionally, in the context of animation, Zordan [33] described similar response delays as subjectively believable.

8.1 Fitness Measure

Evolutionary selection is based on a simple fitness measure f calculated as

$$f = c_d \max(\|\mathrm{proj}_{\boldsymbol{j}}\, \boldsymbol{d}\|,\ \epsilon) + f_b, \tag{3}$$

where $\boldsymbol{d}$ is the vector from the starting position to the hindmost foot, and $\boldsymbol{j}$ is a fixed unit vector pointing in the direction the animator wishes the character to walk. Throughout this work, $\boldsymbol{j}$ is fixed to the character's starting facing direction, rewarding simple forward walking. c_d is a constant scaling factor, and ϵ a fixed small positive value. f_b is proportional to the time spent upright, $f_b = c_t t$, where t is the elapsed time for the trial and c_t is a constant scale factor. We also experimented with more complex fitness functions for locomotion, however Equation 3 was the most effective and is used for the results presented.

Early Termination. The evaluation of an individual controller halts if an early termination criteria is met. This is a useful way to improve the overall speed of the evolutionary process, and also provides a powerful means to shape the resulting behavior by specifying outcomes the animator deems undesirable.

Waist height. If the z-coordinate of the waist segment's center of mass (CoM) falls below a minimum height (70% of the waist segment's starting height), the simulation is terminated and the genome's final fitness is that computed using the state of the previous time-step. By forcing the CoM to maintain a minimum height, locomotion gaits such as crawling and walking on the knees (instead of the feet) are prohibited.

Instability. If significant numerical instability or joint divergence is detected, the simulation is terminated and the genome's final fitness is set to the minimal allowed value, ϵ.

Stagnation. A trial is terminated if the fitness fails to improve by $\epsilon + \frac{c_t n}{\Delta t}$ over the n seconds of simulated time since the previous improvement. This forces the system to gain fitness by some means other than time, preventing, for example, simply balancing in place for the course of the run.

8.2 Support Harness

During the first 100 generations, a virtual harness provides lateral, posterior and vertical support to the character. A similar harness was described in [17], though our variation offers no resistance to forward or upward motion. Although not found to be strictly necessary, a supporting harness was helpful in reducing the total number of evaluations needed to find a walking controller.

9 Results

Bipedal locomotion is a difficult control problem, partly due to the inherent instability. However, demanding natural-looking motion compounds the difficulty because, as a system, a walking humanoid is relatively unconstrained by either physical limitations (i.e., there are many physically feasible forms of moving forward besides the normal human gait, e.g., hopping, jumping or skipping) or energy minimization (though the human walk is characteristically low-energy). Our method explores an aesthetic subset of these possible motions due to two key factors. First, we use smoothly varying neural networks, and second, we use human-scale models with human-scale actuators.

Example joint angles for a typical successful walk controller are shown in Figure 2. This particular individual is of medium female height, weight and BIB.

9.1 Reliability of Walking Controller Generation

Although evolutionary processes are stochastic by nature, it is desirable that the system be able to reliably generate quality controllers. Over the entire range

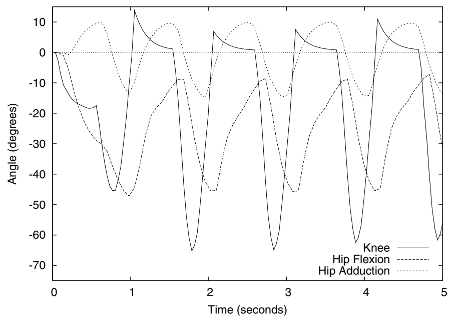

Fig. 2. Joint angles observed for average (50^{th} percentile) female model's walk cycle, starting from a standing position

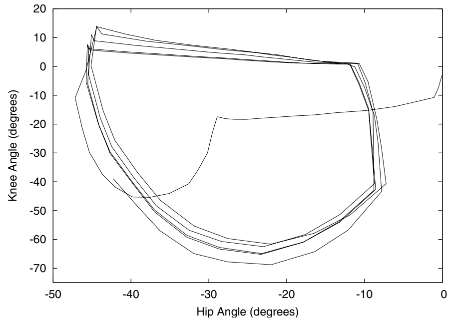

Fig. 3. Phase-space diagram of the hip and knee joints during the gait initiation and subsequent walking cycle

of human body shapes evaluated at walking, 93% of the runs found controllers capable of walking upright for more than two meters (28 out of 30 runs). Both failures were with heavy (95^{th} percentile weight) characters with short stature (5^{th} percentile height), once for a male character and once for a female. Successful walking controllers were found by simply repeating the runs with the same parameters. It is worth commenting that humans with such opposite extremes in body size and weight are likely well outside the normal range.

10 Conclusions

In this work, we have shown that the proposed system can produce walking controllers for a wide range of anthropomorphic bodies. The system is able to do so without the need for *a priori* knowledge of the correct neural topology. The resulting controllers show smooth and believable human-like motion, without the stiffness and phase artifacts associated with methods based on robotics techniques. The controllers are closed-loop, using proprioceptive, tactile and vestibular sensors to maintain balance and improve their performance.

However, the evolutionary process is by nature unpredictable, and may not result in useful controllers for any given character morphology. Also, none of the evolved controllers were capable of sustained walking– generally walks were stable for 5-10 meters before toppling. For the neural networks to evolve sufficient complexity to initiate locomotion and walk for a few meters, they have genome sizes in the several hundred real parameters. Mutations that only effect the stability in the circumstances of the final step are likely to be quite rare, leaving the system fragile and unable to improve. Constraining to or enforcing for stable cycles in neural output may be possible, but may also introduce unwanted visual or aesthetic artifacts in the motion.

A promising next step is to build on the task-specific controllers evolved with our system by composing many together, as described with hand-engineered controllers in [4].

References

1. van de Panne, M., Kim, R., Fiume, E.: Virtual wind-up toys for animation. In: Proceedings of Graphics Interface 1994, Banff, Alberta, Canada, pp. 208–215 (1994)
2. Raibert, M.H., Hodgins, J.K.: Animation of dynamic legged locomotion. In: SIGGRAPH 1991: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, pp. 349–358. ACM Press, New York (1991)
3. Hodgins, J.K., Wooten, W.L., Brogan, D.C., O'Brien, J.F.: Animating human athletics. In: SIGGRAPH 1995: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp. 71–78. ACM Press, New York (1995)
4. Faloutsos, P., van de Panne, M., Terzopoulos, D.: The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. Computers and Graphics 25(6), 933–953 (2001)
5. Yin, K., Loken, K., van de Panne, M.: Simbicon: Simple biped locomotion control. In: Proceedings of the 2007 SIGGRAPH conference, vol. 26. ACM, New York (2007)
6. da Silva, M., Abe, Y., Popović, J.: Interactive simulation of stylized human locomotion. In: International Conference on Computer Graphics and Interactive Techniques. ACM, New York (2008)
7. Shapiro, A., Chu, D., Allen, B., Faloutsos, P.: A dynamic controller toolkit. In: Sandbox 2007: Proceedings of the 2007 ACM SIGGRAPH Symposium on Video Games, pp. 15–20. ACM, New York (2007)
8. Witkin, A., Kass, M.: Spacetime constraints. Computer Graphics 22(4), 159–168 (1988)
9. Liu, C.K., Hertzmann, A., Popovic, Z.: Learning physics-based motion style with nonlinear inverse optimization. ACM Transactions on Graphics (TOG) 24(3), 1071–1081 (2005)
10. van de Panne, M.: Sensor-actuator networks. In: SIGGRAPH 1993: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp. 335–342. ACM Press, New York (1993)
11. Sims, K.: Evolving virtual creatures. In: SIGGRAPH 1994: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp. 15–22. ACM Press, New York (1994)
12. Koza, J.: Genetic Programming: On the Programming of Computers by means of Natural Selection. MIT Press, Cambridge (1992)
13. Gritz, L.: Evolutionary Controller Synthesis for 3-D Character Animation. PhD thesis, George Washington University (1999)
14. Laszlo, J., van de Panne, M., Fiume, E.: Limit cycle control and its application to the animation of balancing and walking. In: Computer Graphics. Annual Conference Series, vol. 30, pp. 155–162 (1996)
15. Nolfi, S., Floreano, D.: Evolutionary Robotics. In: Intelligent Robots and Autonomous Agents. MIT Press, Cambridge (2000)
16. Reil, T., Husbands, P.: Evolution of central pattern generators for bipedal walking in a real-time physics environment. IEEE Trans. Evolutionary Computation 6(2), 159–168 (2002)
17. Reil, T., Massey, C.: Morpho-Functional Machines: The New Species: Designing Embodied Intelligence. Springer, Heidelberg (2003)
18. Golubitsky, M., Stewart, I., Buono, P.L., Collins, J.: Symmetry in locomotor central pattern generators and animal gaits. Nature 401, 693–695 (1999)

19. Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87, 1423–1447 (1999)
20. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)
21. Paul, C.: Bilateral decoupling in the neural control of biped locomotion. In: Proc. 2nd International Symposium on Adaptive Motion of Animals and Machines (2003)
22. Paul, C.: Sensorimotor control of biped locomotion based on contact information. In: Proc. International Symposium on Intelligent Signal Processing and Robotics (2004)
23. McHale, G., Husbands, P.: From Animals to Animats 8. In: Proceedings of the 8th International Conference on Simulation of Adaptive Behavior. MIT Press, Cambridge (2005)
24. Vaughan, E.D., Paolo, E.D., Harvey, I.R.: The evolution of control and adaptation in a 3d powered passive dynamic walker. In: Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems (Alife9). MIT Press, Cambridge (2004)
25. Stanley, K.: Efficient Evolution of Neural Networks Through Complexification. PhD thesis, University of Texas, Austin (2004)
26. Hornby, G.S., Pollack, J.B.: Creating high-level components with a generative representation for body-brain evolution. Artificial Life 8(3), 223–246 (2002)
27. Gruau, F.: Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm. PhD thesis, Laboratoire de l'Informatique du Parallilisme, Ecole Normale Supirieure de Lyon, France (1994)
28. Hornby, G.S.: Shortcomings with tree-structured edge encodings for neural networks. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 495–506. Springer, Heidelberg (2004)
29. Stanley, K.O., Miikkulainen, R.: Competitive coevolution through evolutionary complexification. Journal of Artificial Intelligence Research (21), 63–100 (2004)
30. Department of Defense, U.S.o.A.: Anthropometry of U.S. Military Personnel (DOD-HDBK-743A). Department of Defense, United States of America (1991)
31. Bodenheimer, B., Shleyfman, A., Hodgins, J.: The effects of noise on the perception of animated human running. Computer Animation and Simulation (1999)
32. DJ, D.: Neuromuscular control system. IEEE Transactions on Biomedical Engineering 3, 167–171 (1967)
33. Zordan, V.B., Majkowska, A., Chiu, B., Fast, M.: Dynamic response for motion capture animation. ACM Trans. Graph. 24(3), 697–701 (2005)