Morphology and Gait Control Evolution of Legged Robots

Milton Roberto Heinen
UFRGS – Informatics Institute
CEP 91501-970, Porto Alegre, RS, Brazil
mrheinen@inf.ufrgs.br

Fernando Santos Osório
USP – ICMC
CEP 13560-970, São Carlos, SP, Brazil
fosorio@icmc.usp.br

Abstract

This paper describes our research and experiments with autonomous robots, in which were used genetic algorithms to evolve stable gaits of simulated legged robots in a physically based simulation environment. In our approach, the gait is defined using a finite state machine based on the joint angles of the robot legs, and the parameters are optimized using genetic algorithms. The proposed model also allows the evolution of the robot body morphology. The model validation was performed by several experiments and a valid statistical analysis, and the results show that it is possible to generate fast and stable gaits using genetic algorithms in an efficient manner.

1 Introduction

The autonomous mobile robots have been attracting the attention of a great number of researchers, due to the challenge that this research domain proposes: making these systems capable of intelligent reasoning and able to interact with the environment in which they are inserted in, through sensor perception (infrared, sonar, bumpers, gyro, etc) and motor action planning and execution [5]. The mobile robots are applied in different important tasks like: bomb disarming, exploration of hostile environments and automatic vehicle conduction[10, 20, 17]. At the present time, most of the mobile robots use wheels for locomotion, which makes this task easy to control and efficient in terms of energy consumption, but they have some important disadvantages since they have problems moving across irregular surfaces and crossing borders and edges, like stairs. So, in order to make mobile robots better adapted to human environments and to irregular surfaces, they must be able to walk or have a similar locomotion mechanism used by the humans and animals, that is, they should have legs [5, 1, 11].

However, the development of legged robots capable of moving on irregular surfaces is a quite difficult task, that needs the configuration of many gait parameters. The manual configuration of these parameters demands a lot of effort and time consuming of a human specialist, and the obtained results are usually suboptimal and specific for one robot architecture [4]. Thus, it would be useful to generate the robot gait configuration in an automatic manner, using machine learning techniques to perform this task.

One of these machine learning techniques that are most adapted for this specific task are the genetic algorithms (GA) [8, 24]. This is a reasonable choice because according to the Darwin's evolution theory, the locomotion mechanisms of several life forms resulted from the natural evolution, what makes the use of genetic algorithms a natural solution since they are biologically inspired and may generate biologically plausible solutions. From the computational point of view, the GA are also very well adapted for the automatic gait configuration of legged robots, because: (a) they use a multi-criterion optimization method to search solutions in the configuration space, that means in our specific case, they are capable to optimize not only the gait velocity, but also the stability and even other gait parameters; (b) they do not need local information for the error minimization, nor the gradient calculation, what is very important for the gait parameters generation and optimization, since it is very difficult to have available some a priory training data for supervised learning; (c) if correctly used, the GA is capable to avoid local minima [24].

In some previous work [14, 12, 13] we made a comparative study between robots with four (tetrapod) and six (hexapod) legs, and also about the use and the influence of different fitness functions used in GA robot control optimization. This paper extends these previous work including the evolution of the robot morphology at the same time that the evolution of the control parameters. This paper is structured as follows: Section 2 describes several concepts related to mobile robots simulation; Section 3 describes some related work in control of legged robots; Section 4 describes the genetic algorithms; Section 5 describes the proposed model, called LegGen; Section 6 describes the accomplished experiments and the obtained results; and Section 7 provides some final remarks and future perspectives.

2 Mobile robot simulation

In order to obtain a more realistic mobile robots simulation, several elements of the real world should be present in the simulated model, doing the simulated bodies to behave in a similar way related to the reality and also to interact with the environment they are inserted in. Especially, it is necessary that the robot suffers from instability and fall down if badly positioned and controlled, and also it should interact and collide against the environment objects in a realistic manner [25]. To accomplish that, it is necessary to model the physics laws in the simulation environment (e.g. gravity, inertia, friction, collision). Nowadays, several physics simulation tools exist used for the implementation of physics laws in simulations. After analyzing different possibilities, we chosen a widely adopted physics simulation library, called Open Dynamics Engine - ODE[1].

ODE is a software library for the simulation of articulated rigid bodies dynamics. With this software library, it is possible to make autonomous mobile and legged robots simulations with great physical realism. In ODE, several different rigid bodies can be created and connected using different types of joints. To move bodies using ODE, it is possible to apply forces or torques directly to the body, or it is possible to activate and control angular motors. An angular motor is a simulation element that can be connected to two articulated bodies, which have several control parameters like axis, angular velocity and maximum force. With these elements, it is possible to reproduce articulations present in real robots with a high precision level [25].

3 Related work

Control of locomotion in legged robots is a challenging multidimensional control problem [5, 1]. It requires the specification and coordination of motions in all robot legs while considering factors such as stability and surface friction [21]. This is a research area which has obvious ties with the control of animal locomotion, and it is a suitable task to use to explore this issue [29]. It has been a research area for a considerable period of time, from the first truly independent legged robots like Phony Pony [23], where each joint was controlled by a simple finite state machine, to the algorithmic control of bipeds and quadrupeds by Raibert [28].

Lewis [22] evolved controllers for a hexapod robot which learned to walk inspired on insect-like gaits. After a staged evolution, its behavior was shaped towards the final goal of walking. Bongard [2] evolved the parameters of a dynamic neural network to control various types of simulated robots. Busch [3] used genetic programming to evolve the control parameters of several robot types. Jacob [19], on the other hand, used reinforcement learning to control a simulated tetrapod robot. Reeve [29] evolved the parameters of various neural network models using GA.

In most of these approaches described above, the fitness function used was the distance traveled by the robot in a predefined amount of time. Although this fitness function has been largely used, it may hinder the evolution of more stable gaits [9]. In our approach, we use in the fitness function, beyond distance traveled, sensory information (gyroscope and bumpers) to allow stable and fast gaits [12, 15].

4 Genetic algorithms

Genetic algorithms are optimization methods of stochastic space state search based on the Darwin's Natural Evolution Theory, that were proposed in the 60s by John Holland [18]. They work with a population of initial solutions, called chromosomes, which are evolved by several operations during a certain number of generations, usually reaching a well optimized solution, and preserving the best individuals according to a specific evaluation criterion. In order to accomplish this, in each generation the chromosomes are individually evaluated using a function that measures its performance, called fitness function [24]. Usually the chromosomes with the best fitness values are selected to generate the next generation applying the crossover and mutation operations. Thus, each new generation tends to adapt and improve the quality of solutions, until we obtain a solution that satisfies a specific objective.

5 Proposed model

The LegGen simulator[2] [12, 11, 16, 15] was developed to accomplish the gait control of simulated legged robots in an automatic way. This simulator is composed of several modules, showed in Figure 1.

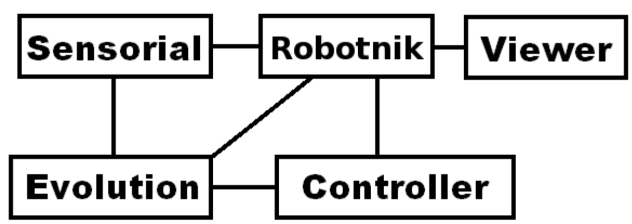


Figure 1. The LegGen modules

The *Robotnik* module is responsible for the robot and virtual environment creation using the ODE library. The *Evolution* module is responsible for the evolution of the control parameters using genetic algorithms. The *Sensorial* module is responsible for sensory information reading

[1] Open Dynamics Engine (ODE) – http://www.ode.org

[2] LegGen – http://www.inf.ufrgs.br/~mrheinen/leggen/

during simulation and fitness calculation for each individual. The *Viewer* module is responsible for the visualization of results in a three-dimensional graphic environment. The *Controller* module is responsible for the robot joints control. The LegGen prototype was implemented using the C++ programming language and the free software libraries ODE and GAlib. LegGen reads two configuration files, one describing the robot format and dimensions and the other file describing the simulation parameters.

LegGen works as follows: initially the file describing the robot is loaded, and the robot is created in the ODE environment according to file specifications. After this, the simulator parameters are loaded, and the genetic algorithm is initialized and executed until the number of generations is reached. The evaluation of each chromosome is accomplished in the following way:

- The robot is placed in the starting position;
- The genome is read and the control parameters are set;
- The physical simulation is executed during 30 seconds;
- Gait and sensor information are captured during each physical simulation;
- Fitness is calculated and returned to GA;

During the simulation, if all paws of the robot leave the ground at same time for more than one second, the simulation of this individual is immediately stopped, because this robot probably fell down, and therefore it is not necessary to continue the physical simulation of this individual.

5.1 Gait control

The gait control is generated using a finite state machine (FSM), in which is defined for each state and for each robot joint their final expected angles configuration [2]. In this way, the controller needs to continually read the joint angles state, in order to check if the joint motor accomplished the task. Real robots do this using sensors (encoders) to control the actual angle attained by the joints [5, 1]. So, in this approach the gait control is accomplished in the following way: initially the controller verify if the joints have already reached the expected angles. The joints that do not have reached them are moved (activate motors), and when all the joints have reached their respective angles, the FSM passes to the following state.

To synchronize the movements, it is important that all joints could reach their respective angles at almost the same time. This is possible with the application of a specific joint angular velocity for each joint, calculated by the equation:

$$V_{ij} = Vr_i(\alpha_{ij} - \alpha_{ij-1}) \qquad (1)$$

where V_{ij} is the velocity applied to the motor joint i in the j state, α_{ij} is the joint angle i in the j state, α_{ij-1} is the joint angle i in $j-1$ state, and Vr_i is the reference velocity of the i state, used to control the set velocity. The reference velocity Vr is one parameter of the gait control that is also optimized by the genetic algorithm. The other parameters are the joint angles for each state. To reduce the search space, the GA generates values only between the maximum and minimum accepted values for each specific parameter.

5.2 Evolution

In our model, the control parameters are evolved using genetic algorithms. The GA implementation used in our simulator was based on the GAlib software library[3], developed by Matthew Wall of Massachusetts Institute of Technology (MIT). GAlib was selected as it is one of the most complete, efficient and well known libraries for genetic algorithms simulation, and also it is a free and open source C++ library. In LegGen, a genetic algorithm as described by Goldberg in his book [8] was used, and a floating point type genome was adopted. In order to reduce the search space, alleles were used to limit generated values only to possible values for each parameter. Table 1 shows the parameter values used by GA.

Table 1. Parameters of the LegGen simulator

Par-ID	Parameter	Value
1	One point crossover	0.80
2	Mutation rate	0.08
3	Population size	350
4	Number of generations	700

The fitness evaluation uses the following sensory information that must be calculated: (a) the distance $D = x_1 - x_0$ covered by the robot in the x axis, where x_0 is the x start position and x_1 is the end x position; (b) the instability measure G, calculated using the robot position variations in the x, y and z axis. These variations are collected during the physical simulation, simulating a gyroscope sensor, which is a sensor present in some modern robots [5]. The instability measure G (Gyro) is then calculated by [9]:

$$G = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x}_x)^2 + \sum_{i=1}^{N}(y_i - \overline{x}_y)^2 + \sum_{i=1}^{N}(z_i - \overline{x}_z)^2}{N}} \qquad (2)$$

where N is the number of sample readings, x_i, y_i and z_i are the data collected by the simulated gyroscope in the time i, and $\overline{x}_x$, $\overline{x}_y$ and $\overline{x}_z$ are the gyroscope reading means:

$$\overline{x}_x = \frac{\sum_{i=1}^{N} x_i}{N}, \quad \overline{x}_y = \frac{\sum_{i=1}^{N} y_i}{N}, \quad \overline{x}_z = \frac{\sum_{i=1}^{N} z_i}{N} \qquad (3)$$

[3] GAlib – http://www.lancet.mit.edu/ga/

After finishing the sensory information processing, the fitness function F is then calculated by the equation:

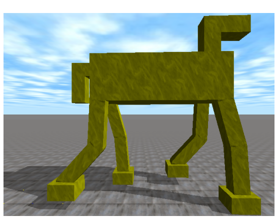
$$F = \frac{D}{1+G} \quad (4)$$

Analyzing this fitness function, we see that the individual better qualified will be the one that has the best relationship between velocity and stability, so the best solutions are those that move fast, but without losing the stability.

5.3 Modeled robot

According to the ODE documentation, computational complexity when using the ODE library is $O(n^2)$, where n is the amount of bodies present in the simulated physical world. Thus, in order to maintain the simulation speed in an acceptable rate, we should use few and simple objects. For this reason, all the simulated robots were modeled with simple objects, as rectangles and cylinders, and they have only the necessary articulations to perform the gait. Thus, body parts as the head and the tail are usually not present in the modeled robots. In order to keep our robot project simple, the joints used in the robots legs just move around the z axis of the robot (the same axis of our knees), and the simulations just used robots walking in a straight line. In the near future, we plan to extend our simulator to accept more complex robot models and joints.

Several robot types were developed and tested, before we defined the final main model, presented in Figure 2. The simulated robots dimensions are approximately the dimensions of a medium sized dog. The joint restrictions used in the simulated robot are similar to its biological equivalent, with the following values: Hip=[-60°;15°]; Knee=[0°;120°]; Ankle=[-90°;30°]. All legs have these same joint restrictions.



	Dimensions		
Part	**x**	**y**	**z**
Body	45.0cm	15.0cm	25.0cm
Thigh	5.0cm	15.0cm	5.0cm
Shin	5.0cm	15.0cm	5.0cm
Paw	8.0cm	5.0cm	9.0cm

Figure 2. Modeled robot

5.4 Morphology evolution

According to Pfeifer [27], in the nature the evolution of the control (nervous system) does not occurs independently of the body morphology evolution. Instead, this is a process that happens at same time. This strategy is very used in the artificial life area [30, 31, 6, 7]. In the previous section, the robot model used in our previous work [12, 15] was described. This robot was modeled in an empirical way, inspired in four leg animals, but with some simplifications. But when the morphology and the control parameters are evolved at same time, this makes it possible to discover new robot models, without a biological equivalent, but equally or more efficient [27, 26]. Thus, LegGen was extended to allow the evolution of the robot morphology at the same time that the evolution of the control parameters. To make this possible, new genes were included in the GA, which encodes the robot segments using three floating point values (x, y and z dimensions).

6 Results

This section describes the accomplished experiments and the achieved results. In order to evaluate the morphology evolution importance, were executed 10 different experiments (i) evolving just the control parameters and (ii) evolving the robot morphology and control parameters at the same time. Table 2 shows the results obtained in these experiments. The first column (**E**) describes the individual experiment index. The next columns show the values of the fitness function (F), distance (D) and gyro instability measure (G), respectively. The last two rows in Table 2 show the mean (μ) and the standard deviation (σ) computed over these 10 experiments.

Table 2. Experiment results

	Just control			Morphology & control		
E	F	D	G	F	D	G
1	16.265	29.19	0.079	18.802	38.03	0.101
2	16.635	28.31	0.070	17.903	32.96	0.075
3	16.991	27.85	0.063	19.839	39.52	0.099
4	16.678	27.92	0.067	17.801	37.86	0.112
5	16.157	28.20	0.074	20.093	27.41	0.031
6	15.965	31.13	0.093	15.902	32.80	0.105
7	17.335	29.63	0.070	18.869	41.13	0.117
8	16.654	29.04	0.074	18.498	36.22	0.095
9	16.289	30.15	0.085	19.078	39.16	0.105
10	16.227	29.81	0.083	15.572	37.40	0.140
μ	**16.520**	**29.12**	**0.076**	**18.235**	**36.25**	**0.098**
σ	**0.420**	**1.08**	**0.009**	**1.504**	**4.10**	**0.029**

We fixed the number of states in the automata to four. These parameters were defined after a careful preliminary study [12, 16] based on experiments. We spent a total of 149.22 hours processing the final experiments of Table 2. Figure 3 shows the box plot graph and the confidence interval (CI) of 95%, related to the fitness values obtained in the experiments presented in Table 2.

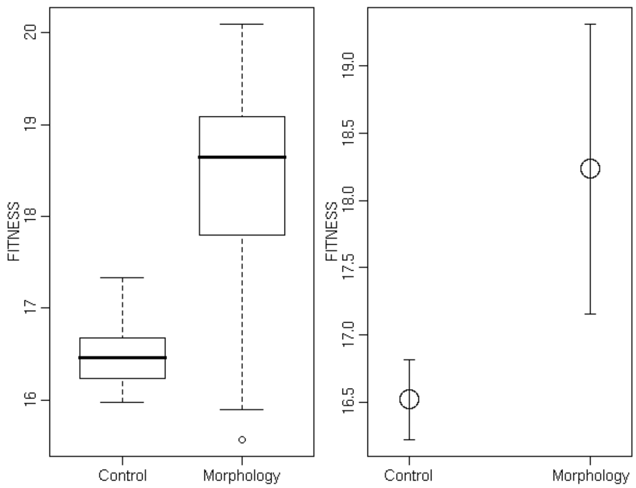


Figure 3. Boxplot and confidence interval

According to Figure 3, the results obtained by the morphology and control evolution are clearly superior to those obtained using just the control evolution, since the confidence intervals are not superposed. Figure 4 shows a walking accomplished by Figure 2 robot, and Figure 5 shows a walking accomplished by an evolved robot[4]. Figure 6

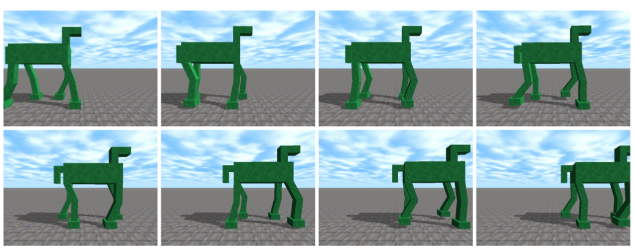


Figure 4. Gait of the modeled robot

shows the morphologies evolved in Table 2 experiments (the numbers in the top-left corner refers to the experiment number in Table 2). Observing Figure 6, it is noticed that the large state space allows the evolution of different solutions, even so efficient, in a similar manner that was occurred in the natural evolution.

7 Conclusions and perspectives

The main goal of this paper was to describe our research and experiments with autonomous robots, in which were used genetic algorithms to evolve stable gaits of simulated legged robots in a physically based simulation environment. The GA evolves parameters used to control the robot actuators and also the robot morphology, and this evolution was

[4]Some demonstration videos are available in LegGen website.

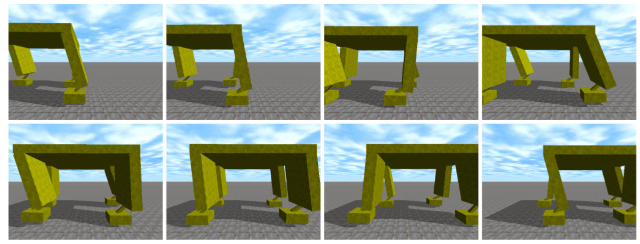


Figure 5. Robot evolved in 6th experiment

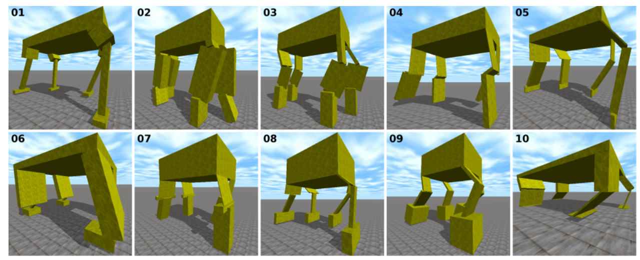


Figure 6. morphologies evolved

tested into a virtual environment using the ODE rigid body dynamics simulation tool. The accomplished experiments demonstrate that the morphology evolution is superior to the evolution of the control parameters only.

Some future work includes improving the robot gait in order to walk on irregular surfaces and to go upstairs or downstairs, as well as, to implement in hardware the simulated robot, once we have now acquired sufficient experience in order to design, implement and fine tune the control of the legged robots.

Acknowledgments

This work was supported by Fapergs, CAPES and CNPq.

References

[1] G. A. Bekey. *Autonomous Robots: From Biological Inspiration to Implementation and Control*. The MIT Press, Cambridge, MA, USA, 2005.

[2] J. C. Bongard and R. Pfeifer. A method for isolating morphological effects on evolved behaviour. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour (SAB)*, pages 305–311, Edinburgh, UK, Aug. 2002. The MIT Press.

[3] J. Busch, J. Ziegler, C. Aue, A. Ross, D. Sawitzki, and W. Banzhaf. Automatic generation of control programs for walking robots using genetic programming. In E. Lutton, J. A. Foster, J. Miller, C. Ryan, and A. G. B. Tettamanzi, editors, *Proceedings of the 5th European Conference on Genetic Programming (EuroGP)*, volume 2278 of *Lecture*

Notes in Computer Science, pages 258–267, Kinsale, Ireland, Apr. 2002. EvoNet, Springer-Verlag.

[4] S. Chernova and M. Veloso. An evolutionary approach to gait learning for four-legged robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, Sept. 2004. IEEE Press.

[5] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, Cambridge, UK, 2000.

[6] P. Eggenberger. Cell interactions as a control tool of developmental processes for evolutionary robotics. In *From Animal to Animats: Proceedings of the 4th International Conference on Simulation of Adaptative Behaviour*, pages 440–448, Cambridge, MA, USA, 1996. The MIT Press.

[7] P. Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. In P. Husbands and I. Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life (ECAL'97)*, pages 205–213, Cambridge, MA, USA, 1997. The MIT Press.

[8] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, USA, 1989.

[9] D. Golubovic and H. Hu. Ga-based gait generation of sony quadruped robots. In *Proc. 3th IASTED International Conference on Artificial Intelligence and Applications (AIA)*, Benalmadena, Spain, Sept. 2003. International Association of Science and Technology for Development (IASTED).

[10] F. J. Heinen and F. S. Osório. HyCAR - a robust hybrid control architecture for autonomous robots. In *Proc. of Hybrid Intelligent Systems (HIS)*, volume 87, pages 830–840, Santiago, Chile, 2002. Soft Computing Systems, IOS Press.

[11] M. R. Heinen. Controle inteligente do caminhar de robôs móveis simulados. Master's thesis - applied computing, Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS, Brazil, Feb. 2007.

[12] M. R. Heinen and F. S. Osório. Applying genetic algorithms to control gait of physically based simulated robots. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, July 2006. IEEE World Congress on Computational Intelligence (WCCI), IEEE.

[13] M. R. Heinen and F. S. Osório. Gait control generation for physically based simulated robots using genetic algorithms. In *Proceedings of the International Joint Conference 2006, 10th Ibero-American Conference on AI (IBERAMIA), 18th Brazilian Symposium on AI (SBIA)*, Lecture Notes in Computer Science, Ribeirão Preto - SP, Brazil, Oct. 2006. Springer-Verlag.

[14] M. R. Heinen and F. S. Osório. Neural networks applied to gait control of physically based simulated robots. In A. M. P. Canuto, M. C. P. de Souto, and A. C. R. da Silva, editors, *Proceedings of the International Joint Conference 2006, 9th Brazilian Neural Networks Symposium (SBRN)*, Ribeirão Preto, SP, Brazil, Oct. 2006. International Joint Conference IBERAMIA/SBIA/SBRN 2006, IEEE Press.

[15] M. R. Heinen and F. S. Osório. Applying genetic algorithms to control gait of simulated robots. In *Proceedings of the IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA) 2007*, pages 500–505, Cuernavaca, Morelos, Mexico, Sept. 2007. IEEE Press.

[16] M. R. Heinen and F. S. Osório. Evolving gait control of physically based simulated robots. *Revista de Informática Teórica e Aplicada (RITA)*, XVI(1):119–134, 2007.

[17] M. R. Heinen, F. S. Osório, F. J. Heinen, and C. Kelber. SEVA3D: Using artificial neural networks to autonomous vehicle parking control. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, July 2006. IEEE World Congress on Computational Intelligence (WCCI), IEEE Press.

[18] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA, 1975.

[19] D. Jacob, D. Polani, and C. L. Nehaniv. Legs than can walk: Embodiment-based modular reinforcement learning applied. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 365–372, Espoo, Finland, June 2005. IEEE Press.

[20] C. Kelber, C. R. Jung, F. S. Osório, and F. J. Heinen. Electrical drives in intelligent vehicles: Basis for active driver assistance systems. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, volume 4, pages 1623–1628, Dubrovnik, Croatia, 2005. IEEE Press.

[21] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2619–2624, New Orleans, LA, USA, Apr. 2004. IEEE Press.

[22] M. A. Lewis, A. H. Fagg, and A. Solidum. Genetic programming approach to the construction of a neural network for control of a walking robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2618–2623, Nice, France, 1992. IEEE Press.

[23] R. B. McGhee. Robot locomotion. *Neural Control of Locomotion*, pages 237–264, 1976.

[24] M. Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, MA, USA, 1996.

[25] F. S. Osório, S. R. Musse, R. Vieira, M. R. Heinen, and D. C. Paiva. *Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation*, volume 2 of *Research in Interactive Design – Proceedings of the Virtual Concept Conference 2006*, pages 1–45. Springer-Verlag, Berlin, Germany, 2006.

[26] R. Pfeifer, F. Iida, and J. Bongard. New robotics: Design principles for intelligent systems. *Artificial Life*, 11(1-2):99–120, Jan. 2005.

[27] R. Pfeifer and C. Scheier. *Understanding Intelligence*. The MIT Press, Cambridge, MA, USA, 1999.

[28] M. H. Raibert. *Legged Robots That Balance*. The MIT Press, Cambridge, MA, USA, 1986.

[29] R. Reeve and J. Hallam. An analysis of neural models for walking control. *IEEE Transactions on Neural Networks*, 16(3):733–742, May 2005.

[30] K. Sims. Evolving 3d morphology and behavior by competition. In R. A. Brooks and P. Maes, editors, *Artificial Life IV Proceedings*, pages 28–39, Cambridge, MA, USA, 1994. The MIT Press.

[31] K. Sims. Evolving virtual creatures. *Computer Graphics*, 28:15–24, July 1994.