Biped Walking Pattern Generation Using Reinforcement Learning

Jungho Lee and Jun Ho Oh

Abstract—**In this research, a stable biped walking pattern is generated. The walking pattern is a simple third order polynomial. To find the proper boundary condition, the reinforcement learning algorithm is used. The final velocity of the walking pattern is chosen as learning parameter. To test the algorithm, a simulator that includes the reaction between the foot of the robot and the ground was developed. The algorithm is verified through a simulation.**

***Index Terms*—Biped Walking, CMAC, Reinforcement Learning and Walking Pattern**

I. INTRODUCTION

GENERALLY, many control methods need a system model, based on these models, controllers are designed to perform desired motions. However, if the system is difficult to model, these control methods are useless. In these cases, control methods through reinforcement learning can serve as an alternative method. Reinforcement learning is a learning algorithm that mimics the human learning procedure from experience [9].

Recently many research groups have reported results concerning a biped walking robot [1, 2, 3, 4, 10]. These robots can walk stably over level ground and inclined ground, go upstairs and even run. These robots use commonly one of two methods for stable walking.

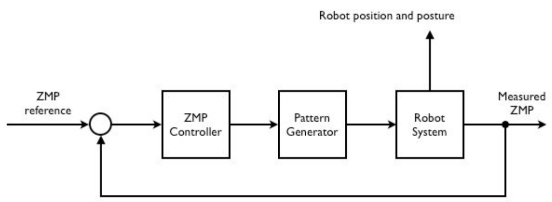

Fig. 1-1 Inverted pendulum model control method

Manuscript received July 28, 2007.
Jungho Lee is with the Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong Yuseong-gu, Daejeon, Republic of Korea (phone: +82-42-869-5223; fax: +82-42-869-8900; e-mail: jungho77@kaist.ac.kr).
Jun Ho Oh is with the Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong Yuseong-gu, Daejeon, Republic of Korea (e-mail: jhoh@kaist.ac.kr).

The first involves a simple inverted pendulum model. Based on this simple model, a feedback controller is designed and follows a ZMP reference.

The second method involves the use of an accuracy model of robot and environment. A stable walking pattern is generated before walking based on the accuracy model.

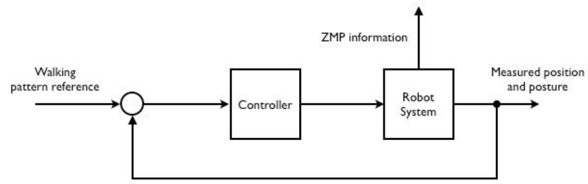

Fig. 1-2 Accuracy model method

Problems can arise with the use of the second method. If the environment changes, the generated walking pattern is likely to be useless. The walking pattern should be regenerated based on the changed model. An additional issue involves the difficulty with modeling an accurate model of the robot with the environment including such factors as the influence of the posture of the robot and the reaction force from the ground. Consequently, the generated walking pattern should be tuned by experiments.

This research begins to solve these problems using reinforcement learning. Numerous research results concerning biped walking using reinforcement learning have been announced, and a number of research group have had good results [5, 6, 7, 8, 23, 24]. Morimoto et al. determined a parameter value, the knee angle of the front leg, for stable and repeated walking in the sagittal plane using a simple actor-and-critic method. The robot involved with their study has a U-shaped foot. Chew et al. also used a parameter value, the foot placement for the front leg, to walk with a constant velocity using what is known as Q-learning. And a simple ankle torque controller is added for stable walking. In addition, Katic et al. and Benbrahim et al. use reinforcement leaning as a sub-control routine to determine the overall biped walking control gain and parameters.

Earlier research on the subject of biped walking using reinforcement learning mainly considers stable walking. However, the posture of the robot is as important as stable walking, for example, if considering climbing stairs or walking across over stepping stones. In these cases, the foot placement of the robot is very important. Each foot should be placed in the required position or the robot will collapse.

Thus, the main goal of this research is to determine a walking pattern that satisfies both stable walking and the required posture_(foot placement) using reinforcement learning. The Q-leaning algorithm is used as the learning method and CMAC(Cerebellar Model Articulation Controller) is used as the generalization method.

The remainder of this paper is organized as follows: Chapter 2 presents the walking pattern generation for stable walking. In Chapter 3, the reinforcement learning agent and simulator for training the reinforcement learning agent are represented. In Chapter 4, simulation results are presented. Conclusions and future works are presented in Chapter 5.

II. Walking Pattern

In this research, third order polynomial ankle and hip joint pattern for a support leg is designed. This pattern is from the moment one foot touches the ground to the moment the other foot touches the ground. It is shown in the Fig. 2-1. To make the body upright from the ground, the sum of the hip, knee, and ankle angles is zero. As the knee angle of the support leg is constant while walking, the hip angle is not independent from the pattern of the ankle for an upright. Thus, only the ankle joint pattern is required.

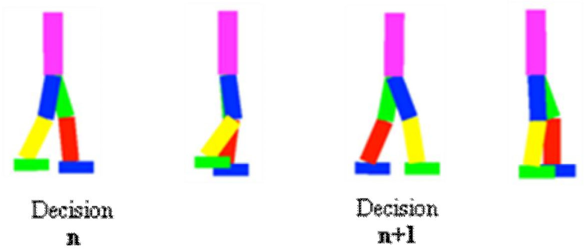

Fig. 2-1 Sequence of walking

To create third order walking pattern, four boundary conditions are needed. These boundary conditions were chosen with a number of factors taken into account. To avoid jerking motions, the pattern must be continuous. For this reason, the angle and angular velocity of the ankle joint at the moment of beginning of the pattern of support leg were chosen as the boundary conditions. Additionally, when the foot must be placed in a specific location, such as upstairs or on stepping stones, the final position of the walking pattern is important. This final position is related to the step length, and this value is defined by the user. Finally, the final velocity of the walking pattern is utilized. Using this final velocity, it is possible to modify the walking pattern shape without changing the final position [12].

However, it is difficult to choose the correct final velocity of the pattern. In addition, it requires numerous trials to tune the final velocity. Thus, in order to find a proper value of this parameter, the reinforcement leaning algorithm is used.

From these four boundary conditions, third order polynomial walking pattern can be generated. Fig. 2-2 shows this process.

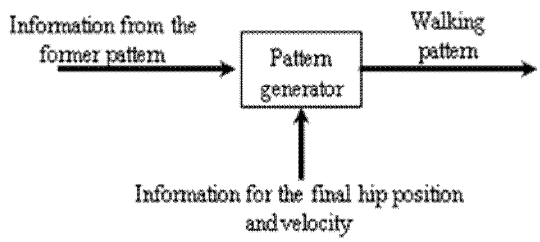

Fig. 2-2 Walking pattern generator

III. Reinforcement Learning

Because reinforcement learning is essentially based on trial-and-error, it is dangerous to apply in actual systems before sufficient training is performed. Therefore, a learning agent should be fully trained in a biped walking robot simulator and then applied to an actual robot. In addition, the biped walking robot simulator can be used to test the walking algorithm, and the walking pattern [4, 15, 16, 17, 18].

The simulator is used to train a reinforcement learning agent, hence, its model is very important. The model used for the simulator should take into account the robot dynamics and the interaction between the robot and its environment model. To build this model, the ODE(Open Dynamics Engine) [22] developed by Russel Smith is used. The ODE provides the dynamics and a collision analysis library. Many researchers use it as a physics library [19, 20, 21]. The ODE library is an open source program.

The reinforcement learning agent uses the Q-learning algorithm which in turn uses the Q-value. To store the various Q-value which represents actual experience or trained data, generalization methods are needed. Here, the CMAC(Cerebellar Model Articulation Controller) is used as a generalization method, as this algorithm is converged quickly and us easy to apply to a real system.

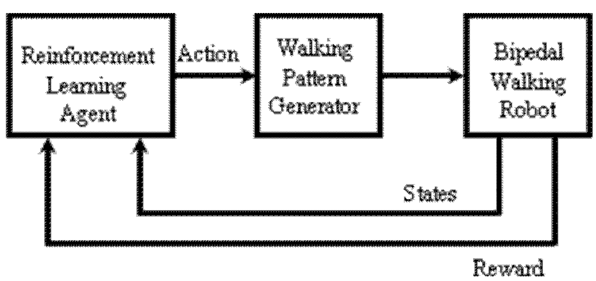

Fig. 3-1 The overall structure

Biped walking system and pattern generation processes involve a discrete system. Before the moment the biped walking pattern is started, the reinforcement learning agent measures the current states of the robot and calculates the action for the biped walking pattern. The robot walks based on this walking pattern, and this pattern does not change while the robot is walking. This procedure is repeated with every walking step. Fig. 3-1 shows this procedure. During the walking process, the robot can collapse or walk stably, this information is stored as the Q-value.

To choose the proper states, the linear inverted pendulum model normally used to model a biped walking robot is used. If the third order polynomial is used as the walking pattern as mentioned previously, the ZMP equation can be written as shown in Fig. 3-2.

As shown in Fig. 3-2, the body position and body acceleration are related to the ZMP position. If the ZMP position is located in support region of the robot, the robot is than dynamically stable. Therefore, choosing body position and body acceleration as states is acceptable. In terms of energy efficiency, conserving the angular and linear momentum is important. The body velocity shows the direction of the movement of the body. Therefore, the body velocity can be another state. Table 3-1 shows the selected states and related reasons behind each state.

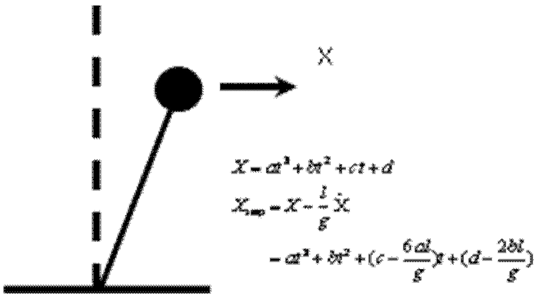

Fig. 3-2 The ZMP position of the inverted pendulum

All states are normalized to -1.0~1.0. However, the reinforcement learning agent has no data regarding the maximum values of the states; the reinforcement learning agent receives this data during the training.

Table 3-1 States

State	Reason
Body position respect to the support foot	Relationship between the C.G. position and the ZMP and the body posture
Body velocity	Angular and linear momentum
Body velocity	Relationship between the C.G. position and the ZMP

First these maximum values are set to be small, in this case 0.1, the reinforcement learning agent then updates the maximum value at every step if the current values are larger than the maximum values.

To create third order polynomial walking pattern, the final velocity is needed, as discussed in the Chapter 2. Hence, the final velocity is used as an action and other conditions are determined by the user. Table 3-2 shows the action and its reason. The maximum value of the action is limited to 0.3m/s. This maximum value is based on the physical motor specification.

Table 3-2 Action

Action	Reason
Final velocity of the walking pattern	Only the final velocity is the unknown parameter. It is related to stable walking [13].

The reward function should be the correct criterion of the current action and also represents the goal of the reinforcement learning agent. The reinforcement learning agent should learn to determine a viable parameter value for the walking pattern generation; its goal is to have the robot to walk stably. The reward is thus divided as 'fall down or not' and 'looking good or not' in this paper. Many candidates exist for this purpose, and the body rotation angle was finally chosen based on trial and error. Table 3-3 shows the reward and reasons. If the robot is falling down, the reinforcement learning agent then gives high negative value as the reward; in the other cases, the robot receives positive values according to body rotation angle. The body rotation angle represents the feasibility of the posture of the robot.

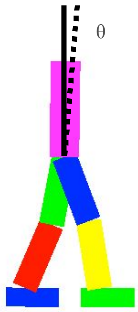

Fig. 3-3 Body rotation angle

Table 3-3 Reward function

Reward	Reason
Fall down or remain upright	This denotes the stability of the robot(or absence of stability)
Body rotation angle respect to support foot	It represents how good it is for stable dynamic walking

IV. Simulation

To test the reinforcement learning agent, a target motion is used. As shown in Table 4-1, the step length is 0.358m and the step period is 0.9 sec. The average speed in this case is 1.432km/h. The HUBO biped walking robot developed by KAIST [4] was used in this simulation. The specifications of this robot are shown in the appendix.

Table 4-1 Simulation conditions

Step period	Step length	
0.9 sec	0.179 + 0.179 = 0.358m	
	Target motion of the front leg	Hip: -0.4 rad
		Knee: 0.2 rad
		Ankle: 0.2 rad
	Target motion of the rear leg	Hip: 0.2 rad
		Knee: 0.2 rad
		Ankle: -0.4 rad

The reinforcement learning agent uses ε-greedy method to explore the learning space. ε-greedy value is initially set to 0.5 and during the training, and this value is decreased to zero gradually. From Fig. 4-1, the reinforcement learning agent converges after the 19th trial. After the 19th trial the robot walks over 400 steps and 120m. In the 10th trial, the robot succeeds in walking 38 steps but this is the local minimum.

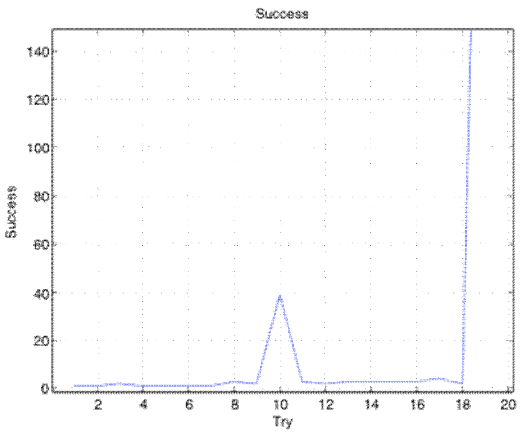

Fig. 4-1 Iteration and number of success

Fig. 4-2 and Fig. 4-3 show the body movements of the robot after the 19th trial. From these figures, the robot walks stably and the walking sequence is repeated.

The body moves up and down as knee angle of the support leg is fixed during walking. This motion is similar to passive walking.

Fig. 4-4 shows the body rotation angle. The maximum value of the body rotation angle is 1.28 degrees, and this occurred when the support leg changed as the dynamic model changed in this case.

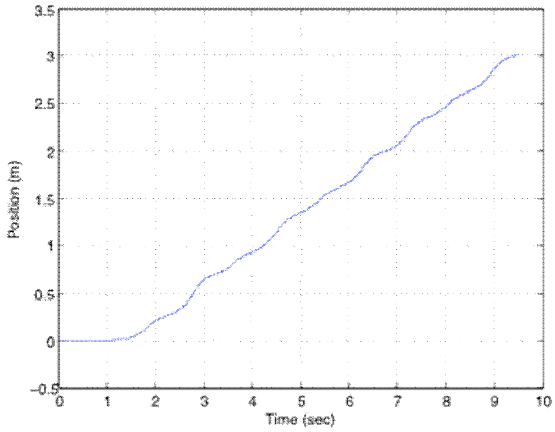

Fig. 4-2 Body movement (x-direction)

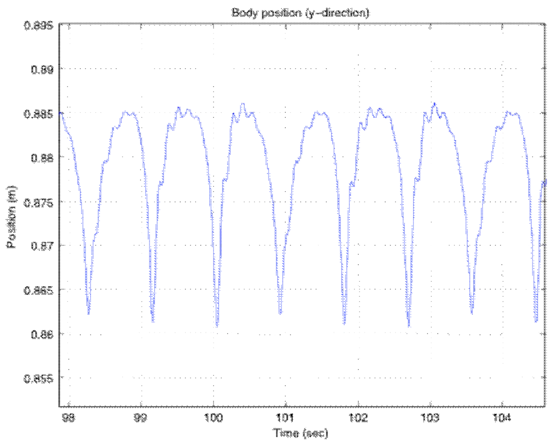

Fig. 4-3 Body movement (y-direction)

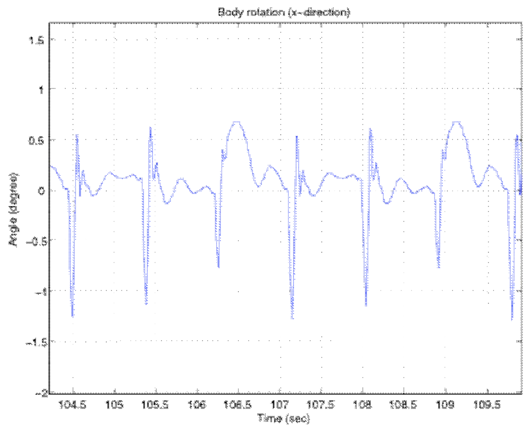

Fig. 4-4 Body rotation angle

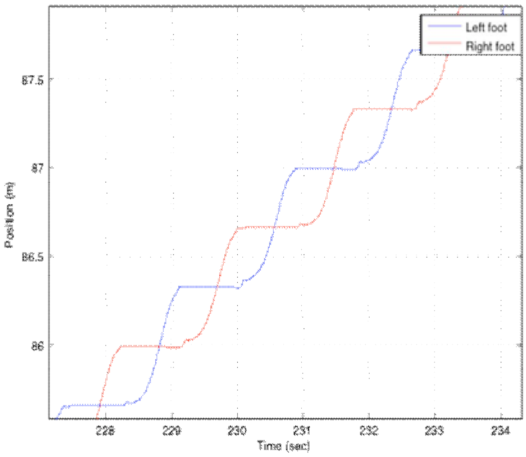

Fig. 4-5 Foot position (x-direction)

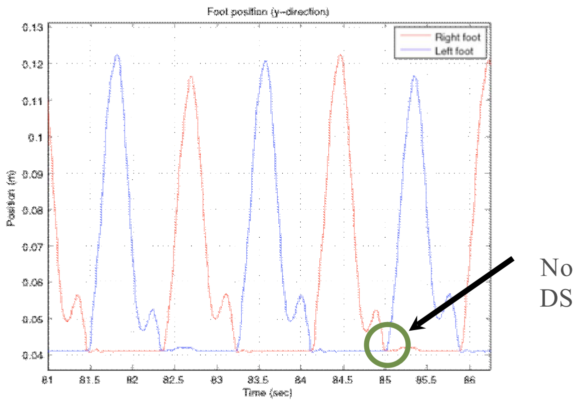

Fig. 4-6 Foot position (y-direction)

Fig. 4-5 and Fig. 4-6 show position of the foot during the stable walking process. From Fig. 4-5, it is shown that the robot follows the given condition mentioned in Table 4-1. Its step length is 0.382m and the step period is 0.9 sec. This implies that the robot can walk stably and will place its foot in the desired position.

V. Conclusion And Future Works

In this research, the learning system for a biped walking robot is developed. Using a reinforcement learning agent, a stable walking pattern is generated which is able to place the foot of the robot in a specific position. This pattern was tested and verified using a simulator. Although the motion of the robot is limited to the sagittal plane at present, this system will be extended to 3-dimensional motion in the future. Also more complicated motions will be tested in the real system.

Appendix

The model used in the simulation is from the real HUBO model. The physical parameters are calculated using 3D CAD software.

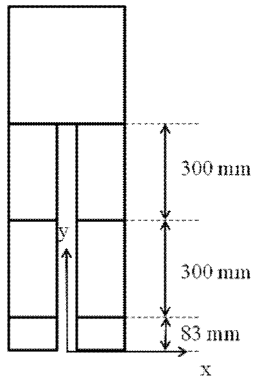

Fig. A-1 HUBO model

Table A-1 Inertia of momentum

Body number	x-x	y-y	z-z
1	0.76285 kgm^2	0.16358 kgm^2	0.74398 kgm^2
2	0.066 kgm^2	0.01146 kgm^2	0.06255 kgm^2
3	0.02164 kgm^2	0.0045 kgm^2	0.01991 kgm^2
4	0.00593 kgm^2	0.0046 kgm^2	0.00848 kgm^2
5	0.066 kgm^2	0.01146 kgm^2	0.06255 kgm^2
6	0.02164 kgm^2	0.0045 kgm^2	0.01991 kgm^2
7	0.00593 kgm^2	0.0046 kgm^2	0.00848 kgm^2

Table A-2 Mass

Body number	mass
1	32.56 kg
2	4.55 kg
3	1.80 kg
4	2.14 kg
5	4.55 kg
6	1.80 kg
7	2.14 kg

References

[1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki and K. Fujimura, *'The Intelligent ASIMO: System Overview and Integration'*, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, p2478-p2483, 2002.

[2] K. Kaneko, S. Kajita, F. Kanehiro, K. Yokoi, K. Fujiwara, H. Hirukawa, T. Kawasaki, M. Hirata and T. Isozumi, *'Design of Advanced Leg Module for Humanoid Robot Project of METI'*, Proc. IEEE International Conference on Robotics and Automation, p38-p45, 2002.

[3] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh and J. Yamaguchi, *'Integrated Motion Control for Walking, Jumping and Running on a Small Bipedal Entertainment Robot'*, Proc. IEEE International Conference on Robotics and Automation, p648-p653, 2004.

[4] Ill-Woo Park, Jung-Yup Kim, Jungho Lee, and Jun-Ho Oh, *'Mechanical Design of the Humanoid Robot Platform, HUBO'*, Journal of Advanced Robotics, Vol. 21, No. 11, 2007.

[5] Jun Morimoto, Gordon Cheng, Christopher Atkeson and Garth Zeglin, *'A Simple Reinforcement Learning Algorithm for Biped Walking'*, Proc. of the 2004 International Conference on Robotics & Automation, p3030-p3035, 2004.
[6] Jun Morimoto, Jun Nakanishi, Gen Endo, Gordon Cheng, Christopher G. Atkeson and Garth Zeglin, *'Poincare-Map based Reinforcement Learning for Biped Walking'*, Proc. of the 2005 International Conference on Robotics & Automation, 2005.
[7] Chew-Meng Chew and Gill A. Pratt, *'Dynamic Bipedal Walking Assisted by Learning'*, Robotica, Volume 20, p477-p491, 2002
[8] Hamid Benbrahim and Judy A. Franklin, *'Biped Dynamic Walking Using Reinforcement Learning'*, Robotics And Autonomous Systems, Volume 22, p283-p302, 1997.
[9] Richard S. Sutton and Andrew G. Barto, *'Reinforcement Learning: An Introduction'*, The MIT Press, 1998.
[10] Jungho Lee, Jung-Yup Kim, Ill-Woo Park, Baek-Kyu Cho, Min-Su Kim, Inhyeok Kim and Jun Ho Oh, *'Development of a Human-Riding Humanoid Robot HUBO FX-1'*, SICE-ICCAS 2006, 2006.
[11] Jung-Yup Kim, *'On the Stable Dynamic Walking of Biped Humanoid Robots'*, Ph. D Thesis, Korea Advanced Institute of Science and Technology, 2006.
[12] Ill-Woo Park, Jung-Yup Kim and Jun-Ho Oh, *'Online Walking Pattern Generation and Its Application to a Biped Humanoid Robot-KHR-3(HUBO)'*, Journal of Advanced Robotics, 2007.
[13] Ill-Woo Park, Jung-Yup Kim, Jung Ho Lee, and Jun-Ho Oh, *'Online Biped Walking Pattern Generation for Humanoid Robot KHR-3(KAIST Humanoid Robot - 3: HUBO)'*, Humanoids 2006, 2006
[14] Hirohisa Hirukawa, Fumio Kanehiro and Shuuji Kajita, *'OpenHRP: Open Architecture Humanoid Robotics Platform'*, Robotics Research: The Tenth International Symposium, Volume 6, p99-p112, 2003.
[15] Rawichote Chalodhorn, David B. Grimes, Gabriel Maganis and Rajesh P. N. Rao, *'Learning Dynamic Humanoid Motion using Predictive Control in Low Dimensional Subspace'*, IEEE-RAS International Conference on Humanoid Robots Humanoids2005, 2005.
[16] Rawichote Chalodhorn, David B. Grimes, Gabriel Maganis, Rajesh P. N. Rao and Minoru Asada, *'Learning Humanoid Motion Dynamics through Sensory-Motor Mapping in Reduced Dimensional Spaces'*, 2006 IEEE International Conference on Robotics and Automation, 2006.
[17] C. Angulo, R. Tellez and D. Pardo, *'Emergent Walking Behaviour in an Aibo Robot'*, ERCIM News 64, p38-p39, 2006.
[18] L.Holh, R. Tellez, O. Michel and A. Ijspeert, *'Aibo and Webots: simulation, wireless remote control and controller transfer'*, Robotics and Autonomous Systems, Volume 54, Issue 6, p472-p485, 2006.
[19] Wolff, K., and Nordin, P., *'Learning Biped Locomotion from First Principles on a Simulated Humanoid Robot using Linear Genetic Programming'*, Genetic and Evolutionary Computation GECCO 2003, 2003.
[20] Wolff, K., and Nordin, P., *'Evolutionary Learning from First Principles of Biped Walking on a Simulated Humanoid Robot',* The Advanced Simulation Technologies Conference 2003 ASTC'03, 2003.
[21] Olivier Michel, *'Webots: Professional Mobile Robot Simulation'*, International Journal of Advanced Robotic System, Volume 1, Number 1, p39-p42, 2004.
[22] Russel Smith, *'www.ode.org/ode.html'*, 2007.
[23] Dusko Katic and Miomir Vukobratovic, *'Control Algorithm for Biped Walking Using Reinforcement Learning'*, 2nd Serbian-Hungarian Joint Symposium on Intelligent Systems, 2004.
[24] H. Benbrahim and J. Franklin, *'Biped dynamic walking using reinforcement learning'*, Robotics and Autonomous Systems Journal, 1997