8

Stable Walking Pattern Generation for a Biped Robot Using Reinforcement Learning

Jungho Lee
Department of Mechanical Engineering, KAIST,
335 Gwahangno Yuseong-gu, Daejeon, 305-701, Republic of Korea
Phone: +82-42-869-5223, Fax: +82-42-869-8900
E-mail: jungho77@kaist.ac.kr

Jun Ho Oh
Department of Mechanical Engineering, KAIST,
335 Gwahangno Yuseong-gu, Daejeon, 305-701, Republic of Korea

Abstract

In this research, a stable biped walking pattern is generated by using reinforcement learning. The biped walking pattern for forward direction is chosen as a simple third order polynomial and sinusoidal function is used for sideway direction. To complete the forward walking pattern, four boundary conditions are needed. In order to avoid jerk motion, initial position and velocity and final position and velocity of the joint are selected as boundary conditions. Also desired motion or posture can be achieved by using the initial and final position. The final velocity of the walking pattern is related to the stability but it is hard to choose proper value. So the final velocity of the walking pattern is chosen as a learning parameter. In order to find the proper boundary condition value, a reinforcement learning algorithm is used. For the sideway movement, a sway amount is selected as learning parameter and a reinforcement learning agent finds proper value for sideway movement. To test the algorithm, a three-dimensional simulator that takes into consideration the whole model of the robot and the environment is developed. The algorithm is verified through a simulation.

Keywords: Biped walking; Reinforcement learning; Robot learning; Humanoid robot

1. Introduction

In various research fields involving humanoid robots, this research centers on the mobility. For a humanoid robot, its method of locomotion is critical. Numerous movement methods have been considered, including the use of wheels and caterpillar-type motion, quadruped motion, and hexapod walking methods inspired from the motions of animals and insects. While these methods have respective strengths, it is preferable that humanoid robots to be

used in human society be capable of biped motion using two legs, resembling a human, as our environment is geared to biped walking. By employing biped motion, the humanoid robot will be able to navigate stairs easily and walk along common walkways. In addition, it will be more familiar to humans.

The realization of biped walking is, however, relatively difficult because a biped walking robot is a highly complex system that is inherently unstable. The realization of biped walking started with the idea of static walking. The robot known as WABOT-1, which was developed by Waseda University in the 1970s, required 45 seconds per step but was the first biped walking robot that utilized the concept of static walking [23]. Static walking is characterized by slow biped movement, and the effect of linear or angular momentum of the robot is neglected.

Dynamic walking was considered after the general idea known as ZMP (Zero Moment Point) was introduced to a biped walking robot by Vukobratovic [1]. The ZMP is a point on the ground plane at which the total moments due to ground reaction force becomes zero. If the ZMP is located in a support region, the robot will never fall down when the robot is walking. The first robot to which the idea of ZMP was applied successfully was the WL-10 series from Waseda University. This robot can walk with 1.3 seconds between each step. After the appearance of the first dynamic biped walking robot, researchers developed a variety of working algorithms based on the ZMP.

Since the first successful presentation of biped walking at Waseda University in the 1970s and 1980s, there have been numerous trials to realize stable biped walking robustly and efficiently, with a number of notable results [2][3][4][5][6]. The humanoid robots developed by these research groups can walk steadily on flat or inclined ground and can even run [7][8][9]. Many notable algorithms developed for stable walking and the methods can be categorized into four paradigms [10]: (a) the use of passive walking as a starting point for the design of active walkers; (b) the use of the "zero moment point" control; (c) the use of fixed control architecture and application of a parameter search to find the parameter settings that yield successful walking gaits; and (d) the development of feedback laws based upon insights into balance and locomotion.

HUBO, the first humanoid robot in Korea, was developed by Oh et al. at KAIST in 2004 [2] [11][12][13]. It is a child-sized (125 cm tall) biped walking robot with 41 DOF (Degree Of Freedom). This humanoid robot combines several biped walking methods for stable walking. For the walking strategy, a walking pattern specific to a given environment is initially designed and a ZMP (Zero Moment Point) feedback controller and other sub-controllers are then used to maintain stability for a dynamically changeable environment. Many researchers use only a ZMP feedback controller. While stable walking can be maintained in this manner, however, it is difficult to generate desired motions. Hence, HUBO uses the aforementioned (b) and (c) paradigms to overcome this problem.

But the key challenge with the existing method used by HUBO is the determination of the proper parameters for designing or generating a stable walking pattern. It is difficult to find proper parameters as they are influenced by many factors such as the posture of the robot and the ground conditions. The existing robot HUBO determines these parameters through many experiments along with an analysis of walking data using a real system. This process is, however, difficult and time-consuming. Furthermore, only an expert can tune these parameters because an unconfirmed walking pattern is tested using a real robot, there is an inherent risk of accidents. This is the starting point of the present research.

In order to overcome these problems, a HUBO simulator and an algorithm that automatically determines an appropriate walking pattern were developed. The HUBO simulator describes the dynamics of the entire system using a physics engine and includes interactions between the robot and its environment, such as reaction forces and collision analysis. The function of this simulator is to test walking patterns. Also reinforcement learning is used to find suitable walking pattern parameters automatically in order to ensure stable walking and tested on this simulator. Reinforcement learning is a learning method that mimics the human learning process (i.e., learning from experience). Furthermore, this control method is usable if the information or the model of the given system is unclear. With the exception of reinforcement learning, many other methods such as those utilizing a neural oscillator, neural network or fuzzy logic can be used to solve this problem. However, these methods are complex compared to reinforcement learning and require an expert or reliable data. Thus, reinforcement learning is used for the generation of stable walking patterns in this study.

Earlier research on the subject of biped walking using reinforcement learning focused primarily on stable walking. However, the posture of a robot is as important as stable walking. For example, posture is particularly important when the robot is climbing stairs or walking across stepping stones. In these cases, foot placement by the robot is very important. Each foot should be placed precisely or the robot can 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. Particularly, the Q-leaning algorithm is used as the learning method and CMAC (Cerebellar Model Articulation Controller) serves as the generalization method. The Q-learning algorithm is easy to implement and its convergence is not affected by the learning policy. Hence, it has been used in many applications.

2. Related work

Former studies concerning the realization of stable biped walking using reinforcement learning are categorized below.

(a) The use of reinforcement learning as a sub-controller to support the main controller
(b) The use of reinforcement learning as a main controller or a reference generator
In Case (a), reinforcement learning is normally used as a gain tuner of the main controller

or as a peripheral controller for stable walking. In Case (b), reinforcement learning is used directly to generate a stable walking pattern or as the main controller for stable walking.

Chew and Pratt [14][54] simulated their biped walking robot, Spring Flamingo (Fig. 2-1), in the planar plane (two-dimensional simulation). This seven-link planar bipedal robot weighed 12 kg, was 0.88m in height and had bird-like legs. A reinforcement leaning system was used as the main controller. The following states were chosen as follows: (a) velocity of the hip in the forward direction (x-coordinate); (b) the x-coordinate of an earlier swing ankle measured with reference to the hip; and (c) the step length. The goal was to enable the robot to walk with constant speed; thus, the learning system received '0' when the robot walked within the boundary speed or was given a negative value as a reward. The position of the swing foot was used as the action. Additionally, a torque controller in each ankle was used to control the ankle joint torque. The same type of torque controller was also used to maintain the velocity of the body. The ankle joint torque was limited to a certain stable value; hence, the robot could walk stably without considering the ZMP. However, because the goal was to realize walking with constant speed, the posture of the robot was not considered.

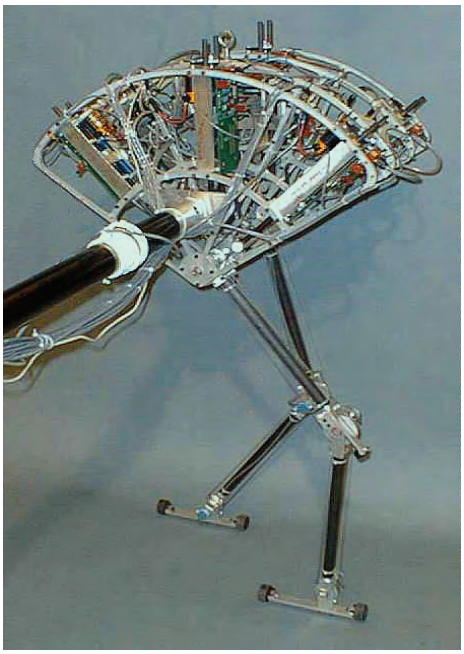

Fig. 1. Spring Flamingo

Benbrahim and Franklin [15] used a reinforcement learning system as both the main and sub-controllers. To achieve dynamic walking with their planar robot, central and other

peripheral controllers were used. The central controller used the experience of the peripheral controllers to learn an average control policy. Using several peripheral controllers, it was possible to generate various stable walking patterns. The main controller activated specific peripheral controllers in an approach that was suitable for specific situations. However, the architecture of the controller was complex, and this approach required many learning trials and a lengthy convergence time.

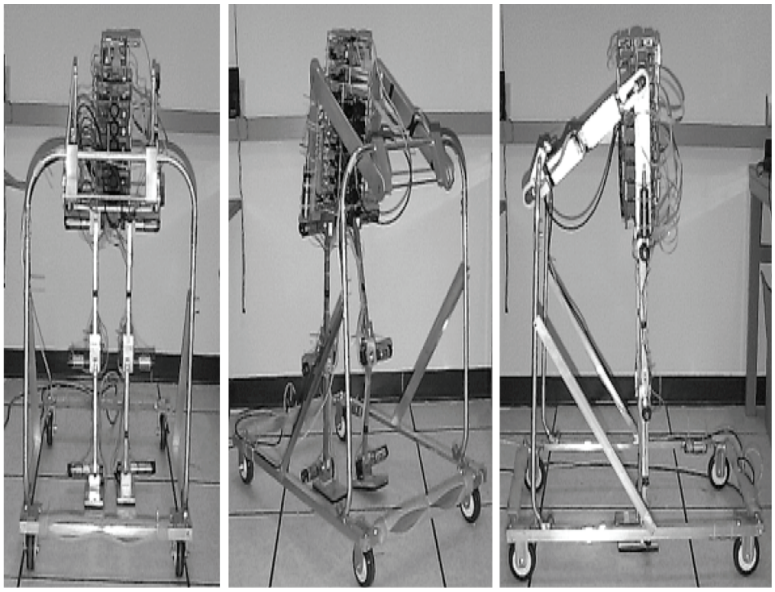

Fig. 2. Benbrahim's biped planar walking robot

Morimoto, Cheng, Atkeson, and Zeglin [16][60] (Fig. 2-3) used a simple five-link planar biped robot to test their reinforcement learning algorithm. The foot of each leg had a shape resembling a 'U', and no joints were used in the ankle. Consequently, it moved in the manner of a passive walker. The goal of the learning system was to walk with constant speed and the states were as follows: (a) velocity of the hip in the forward direction; and (b) forward direction distance between the hip and ankle. The reward was simply falling down or remaining upright, and the action was the angle of the knee joint. The hip joint trajectory was fixed but the step period could vary. If the swing leg touched the ground before the current step period, the next step period was decreased. In addition, if the swing leg touched the ground after the current step period, the step period was increased. This work concentrated only on stable walking; the posture of the robot was not considered.

Schuitema et al. [17] also used reinforcement learning to simulate their planar robot. Their robot, termed Meta, is a passive dynamic walking robot with two hip active joints. The goal of their learning system was to have the robot walk successfully for more than 16 steps.

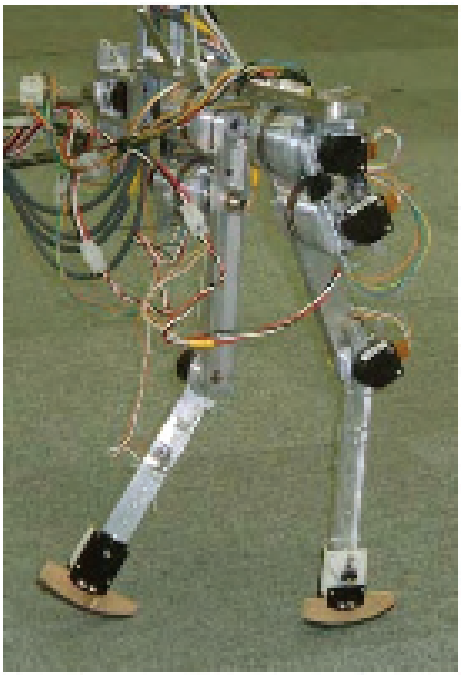

Fig. 3. Morimoto, Cheng, Atkeson, and Zeglin's five-link planar robot

The state space consisted of six dimensions: the angle and angular velocity of the upper stance leg, the upper swing leg, and the lower swing leg. To avoid conditions in which the system would not learn the same thing twice, symmetry between the left and right leg was implemented by mirroring left and right leg state information when the stance leg changed. There was one action dimension, the torque that was applied to the hip joint, which was given a range between -8 and 8 Nm. If the robot walked forward successfully, the learning system received a reward. Additionally, if the body of the robot moves backward, the learning system was penalized. Various experiments were simulated under various constraints and compared the results of each experiment.

Kim et al. [18] used a reinforcement learning system for ZMP compensation. In their research, two-mode Q-learning was used as ZMP compensation against the external distribution in a standing posture. The performance of the Q-learning system was improved using the failure experience of the learning system more effectively along with successful experiences. The roll angle and the roll angular velocity of the ankle were selected as the states. For the action, ankle rolling was given three discrete levels (±0.5°, 0°) during a period of 20 ms. If selecting an action in the opposite direction of the external force, the agent received a reward. If the angle and ZMP constraints were exceeded, the agent was penalized.

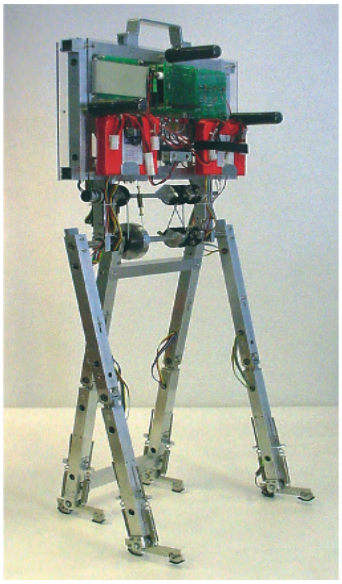

Fig. 4. Meta passive dynamic walking robot

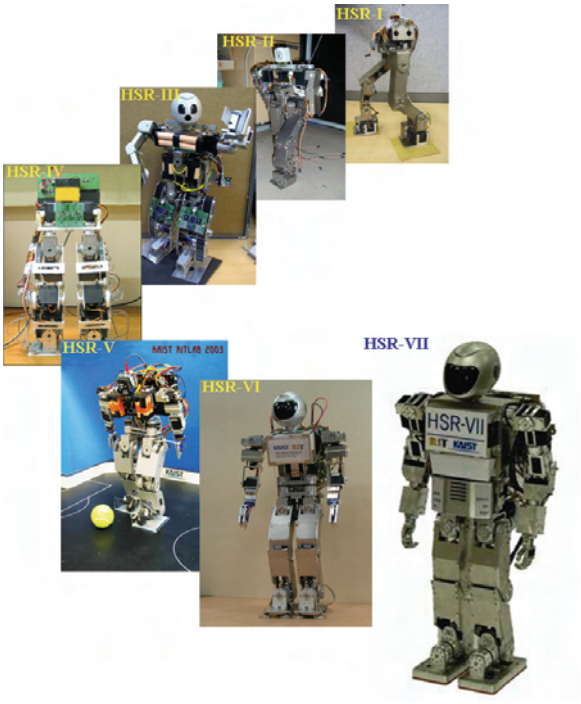

Fig. 5. HanSaRam Series

Other researchers have also proposed learning approaches for stable biped walking [19][20][21]. However, existing research in this area in which reinforcement learning is utilized concerns only stable walking while the posture of the robot is not considered. These researchers normally utilize the position of the swing leg for stable walking. It is, however, difficult to locate the foot in a desired position using only the swing leg. Moreover, it was necessary to use another controller or additional devices for the motion of the support leg. Therefore, this research focuses on the control of the support leg as opposed to control of the swing leg for stable and desired motion walking.

3. Walking pattern

3.1 Sagittal plane

There are several methods of designing a stable walking pattern. But recent researches can be categorized into two groups [22]. The first approach is the 'inverted pendulum model control method' [46][47]. In this method, a simple inverted pendulum model is used as a biped walking model. Based on this model, a proper ZMP reference is generated and a ZMP feedback controller is designed to follow this reference. As this method uses a simple inverted pendulum model, its control structure is very simple. Furthermore, because it follows the ZMP reference for stable walking, stability is always guaranteed. However, it requires a proper ZMP reference and it is difficult to define the relationship between the ZMP reference and the posture of the biped walking robot clearly and accurately. Therefore, it is difficult to select the proper ZMP reference if the posture of the biped walking robot and its walking stability is important. A pattern generator, which translates the ZMP reference to a walking pattern, is also required. Fig. 3-1 shows a block diagram of the 'inverted pendulum model control method'.

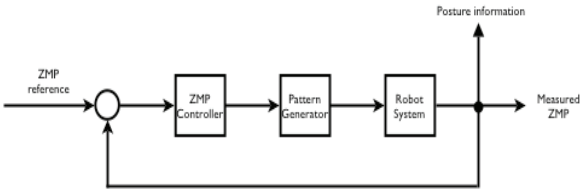

Fig. 6. Inverted pendulum model control method

A second method is known as the 'accuracy model method' [44][45][48][49]. This model requires an accurate model of the biped walking robot and its environment. In this method, a stable walking pattern is generated in advance based on the abovementioned accurate model and the biped walking robot follows this walking pattern without a ZMP feedback controller. One advantage of this method is that it allows control of the biped walking robot with a desired posture. Additionally, it does not require a ZMP controller. However, the generated walking pattern is not a generally functional walking pattern. For example, the walking pattern that is generated for flat ground is not suitable for inclined ground.

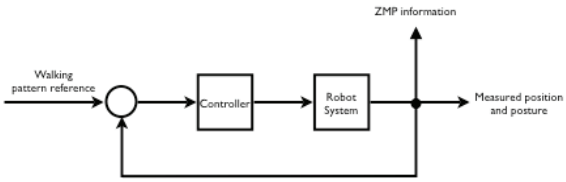

Fig. 7. Accuracy model method

Therefore, with different conditions (e.g., different ground conditions, step lengths, or step periods), new walking patterns should be generated. Fig. 3-2 shows the process of the 'accuracy model method'.

Compared to the 'inverted pendulum model control method', the 'accuracy model method' does not guarantee stability against disturbances; however, it has its own strengths. First, it is possible to control the motion of the biped robot using this method. The 'inverted pendulum model control method' only guarantees stability if the ZMP reference is correct. And it is not possible to control the motion. Second, this method is more intuitive compared to the 'inverted pendulum model control method'. Thus, it is easy to imply physical intuitions using this method. Third, a ZMP controller is not required. Hence, the overall control architecture is simpler with this method compared to the 'inverted pendulum model control method'.

However, an additional problem with the 'accuracy model method' involves difficulty in obtaining an accurate model of the robot and its environment, including such factors as the influence of the posture of the robot, the reaction force from the ground, and so on. Consequently, the generated walking pattern should be tuned by experiments. The generated walking pattern for a specific environment is sensitive to external forces, as this method does not include a ZMP controller. However, when the precise posture of the biped walking robot is required, for example, when moving upstairs or through a doorsill, the 'accuracy model method' is very powerful [9].

In an effort to address the aforementioned issues, the algorithm generating walking patterns based on the 'accuracy model method' was developed using reinforcement learning. To generate a walking pattern, initially, the structure of the walking pattern should be carefully selected. Selection of the type of structure is made based on such factors as polynomial equations and sine curves according to the requirements. The structure of the walking pattern is selected based on the following four considerations [24].

(a) The robot must be easy to operate. There should be minimal input from the operator in terms of the step time, stride, and mode (e.g. forward/backward, left/right) as well as commands such as start and stop.
(b) The walking patterns must have a simple form, must be smooth, and must have a continuum property. It is important that the walking patterns be clear and simple. The

trajectory of the walking patterns should have a simple analytic form and should be differentiable due to the velocity continuum. After the walking patterns are formulated, the parameters for every step are updated.
(c) The calculation must be easy to implement in an actual system. The calculation burden and memory usage should be small and the pattern modification process should be flexible.
(d) The number of factors and parameters that are to be tuned must be small. The complexity of the learning process for the walking patterns is increased exponentially as the number of factors and parameters is increased.

In this research, based on these considerations, a third-order polynomial pattern for the support leg was designed as the walking pattern. This pattern starts from the moment one foot touches the ground and ends the moment the other foot touches the ground (Fig. 3-3).

$$\begin{aligned} z(t) &= Z \\ x(t) &= at^3 + bt^2 + ct + d \end{aligned} \tag{1}$$

To create or complete the third-order forward walking pattern, as shown in Eq. 3-1, four boundary conditions are needed. These boundary conditions were chosen with a number of factors taken into account. First, to avoid jerking motions and formulate a smooth walking pattern, the walking pattern must be continuous. For this reason, the position and velocity of the hip at the moment of the beginning of the walking pattern for the support leg were chosen as the boundary conditions. Additionally, when the foot of the robot is to be placed in a specific location, for example traversing uneven terrain or walking across stepping stones, the final position of the walking pattern is important. This final position is related to the desired posture or step length, and this value is defined by the user. Hence, the final position of the hip can be an additional boundary condition. Lastly, the final velocity of the walking pattern is utilized as the boundary condition. Using this final velocity, it is possible to modify the walking pattern shape without changing the final position, enabling the stabilization of the walking pattern [24]. From these four boundary conditions, a third-order polynomial walking pattern can be generated.

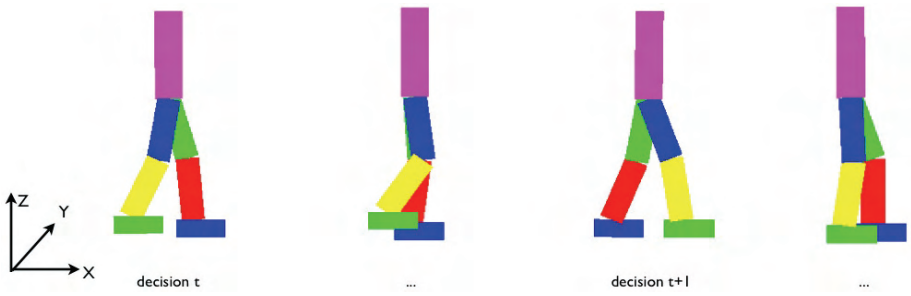

Fig. 8. Sequence of walking

However, it is difficult to choose the correct final velocity of the pattern, as exact models include the biped robot, ground and other environmental factors, are unknown. The existing HUBO robot uses a trial-and-error method to determine the proper final velocity parameter, but numerous trials and experiments are required to tune the final velocity. Thus, in order to find a proper value for this parameter, a reinforcement leaning algorithm is used.

Table 3-1 summarizes the parameters for the sagittal plane motion. And to make problem simpler z-direction movement is fixed as Z (Eq. 3-1).

Boundary condition	Reason
Initial velocity	To avoid jerk motion
Initial position	To avoid jerk motion and continuous motion
Final position	To make wanted posture
Final velocity	To make the walking pattern stable (Unkown parameter)

Table 1. Boundary conditions for the walking pattern

3.2 Coronal plane

Coronal plane movements are periodic motions; it the overall movement range of these movements is smaller than the sagittal plane motion, a simple sine curve is used. If the movement of the z direction is constant, the coronal plane motion can be described by Eq. 3-2, where Y is the sway amount and w is the step period.

$$\begin{aligned} z(t) &= Z \\ y(t) &= Y\sin(wt) \end{aligned} \tag{2}$$

From the simple inverted pendulum model, the ZMP equation can be approximated using Eq. 3-3, where l denotes the length from the ankle joint of the support leg to the mass center of the robot.

$$ZMP(t) = y(t) - \frac{l}{g}\ddot{y}(t) \tag{3}$$

From Eq. 3-2 and Eq. 3-3, the ZMP can be expressed using Eq. 3-4

$$ZMP(t) = Y(1 + \frac{l}{g} w^2) \sin(wt) \tag{4}$$

The length l and the step period w are given parameters and the acceleration of gravity g is known parameter. The only unknown parameter is the sway amount. The sway amount can be determined by considering the step period, the DSP (Double Support Phase) ratio and the support region. If the amplitude of the ZMP is located within the support region, the robot is stable. It is relatively easy to determine the unknown parameter (the sway amount) compared to the sagittal plane motion. However, it is unclear as to which parameter value is most suitable. The ZMP model is simplified and linearized, and no ZMP controller is used in this research. Thus, an incorrect parameter value may result from the analysis.

Therefore, using the reinforcement learning system, the optimal parameter value for stable walking using only low levels of energy can be determined.

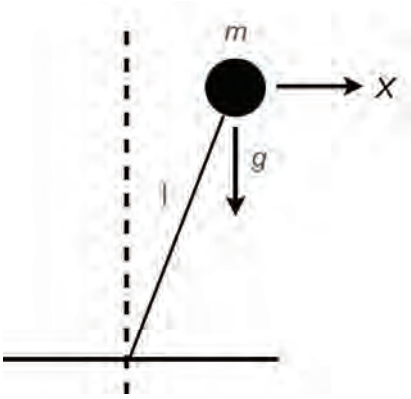

Fig. 9. Inverted pendulum model

4. Simulation

4.1 Simulator

4.1.1 Introduction

Reinforcement learning is based on trial-and-error methodology. It can be hazardous to apply a reinforcement learning system to an actual biped walking system before the learning system is trained sufficiently through many trials, as walking system likely has not been fully analyzed by the learning system. In particular, when such a system is inherently unstable, such as in the case of a biped walking robot, attention to detail is essential. Therefore, it is necessary to train a learning system sufficiently before applying it to a real system. For this reason, simulators are typically used.

A simulator can be used purposes other than for the training of a learning system. For example, simulators can be used for testing new control algorithms or new walking patterns. Various research groups investigating biped walking systems have developed simulators for their own purposes [32][33][34][35][36][37].

The HUBO simulator, which was developed for this study, is composed of a learning system that is in charge of all leaning processes, a physics engine that models a biped robot and its environment, and utility functions to validate the simulation results. Fig. 4-1 shows these modules and the relationships between them. As shown in the figure, learning contents or data obtained from the reinforcement learning module are stored through generalization process. In this study, the CMAC algorithm is used as the generalization method; however, other generalization methods can be easily adapted. The dynamics module, which contains a physics engine, informs the reinforcement learning module of the current states of HUBO. It also receives the action (final velocity of the walking pattern and the sway amount) from the reinforcement learning module, generates a walking pattern, and returns a reward. For the visualization of the movement of a biped walking robot, the OpenGL library is used. Because all components of the HUBO simulator are modularized, it is easy to use with new algorithms or components without modification.

The HUBO simulator contains all of the components necessary for simulating and testing biped walking systems and control algorithms. In addition, all modules are open and can be modified and distributed without limitation. The HUBO simulator follows the GPL (GNU General Public License) scheme.

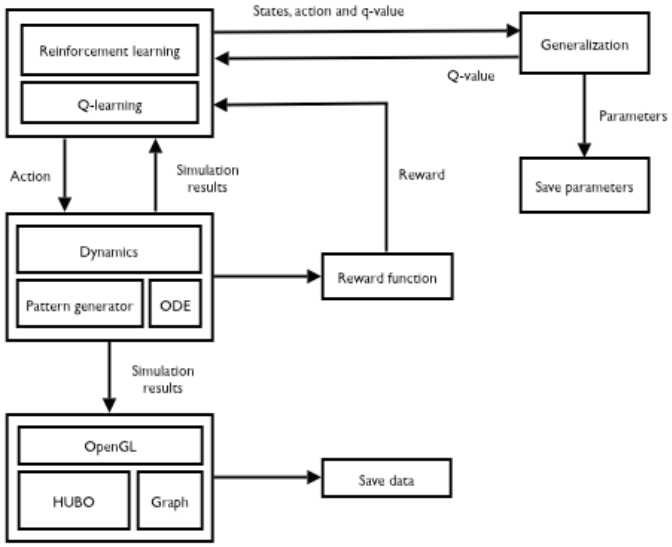

Fig. 10. Structure of the HUBO simulator

4.1.2 Physics Engine

To obtain viable simulation results, the dynamics model of a simulator is very important. If the dynamics model differs greatly from a real model, the result of the simulator is useless. Therefore, it is important to ensure that the simulation model resembles the actual model to the greatest extent possible. Essentially, the model of a biped walking system should contain a robot model as well as a model of the environment of the robot. Many researchers only consider the robot model itself and neglect a model of the environment, which is in actuality more important in a realistic simulation of a biped walking.

For this reason, a physics engine was used to build realistic dynamic model in this study. A physics engine is a tool or APIs (Application Program Interface) that is used for computer simulation programs. In this research, ODE (Open Dynamics Engine) [38] was used to develop the robot and environmental model in an effort to represent the actual condition of the robot accurately. ODE is a rigid body physics engine initially developed by Russell Smith. Its source code is open and is governed by the open source community. ODE provides libraries for dynamics analyses, including collision analyses. The performance of ODE has been validated by various research groups [37][39][40], and many commercial and engineering programs use ODE as a physics engine.

4.1.3 Learning System

The learning system of the HUBO simulator consists of a learning module and a generalization module. The reinforcement learning module uses the Q-learning algorithm, which uses the Q-value. To store the various Q-values that represent actual experience or trained data, generalization methods are needed. Various generalization methods can be used for this. In the present work, the CMAC (Cerebella Model Articulation Controller) algorithm is employed. This algorithm converges quickly and is readily applicable to real systems.

Setting up states and a reward function is the most important process in the efficient use of reinforcement learning. When setting up states, using physical meanings is optional; however, it is important that the most suitable states for achieving the goal are selected. Additionally, the reward function should describe the goal in order to ensure success. The reward function can represent the goal directly or indirectly. For example, if the goal for a biped walking robot is to walk stably, the learning agent receives the reward directly if the robot walks stably without falling down. Otherwise, it is penalized. In addition, the reward function describes the goal of stable walking indirectly, including such factors as the pitch or roll angle of the torso while walking and the walking speed. However, it is important that the reward should suitably describe the goal.

4.1.4 Layout

Fig. 4-2 shows the main window of the HUBO simulator. The motion of HUBO calculated using ODE is displayed in the center region of the HUBO simulator using OpenGL. Each step size or foot placement can be modified from the main window. Fig. 4-3 shows the

learning information window. This window shows information such as the current states and the reward associated with the learning module. In addition, the learning rate and the update rate can be modified from this window. Fig. 4-4 shows the body data window. This window shows the current position and orientation of each body. As lower body data is important for the system, only the data of the lower body is represented. Fig. 4-5 shows the joint angle of the lower body. The data of the force and torque for each ankle joint is shown in the force-torque data window in Fig. 4-6.

The HUBO simulator was developed using the COCOA[○,R] library under a Mac OS X [○,R] environment. As COCOA is based on the Object-C language and all structures are modulated, it is easy to translate to other platforms such as Linux[○,R] and Windows[○,R].

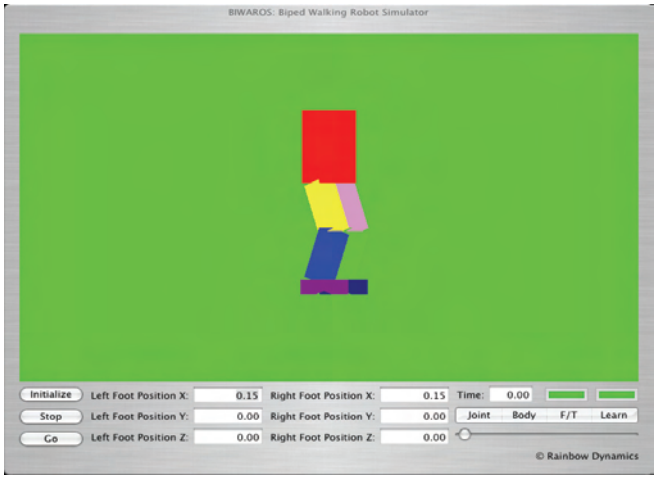

Fig. 11. Main window of the HUBO simulator

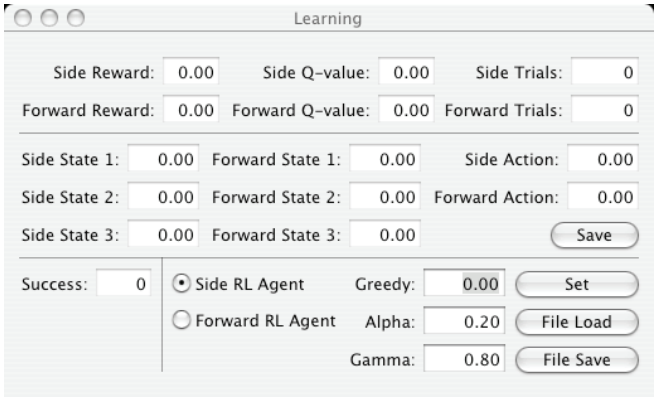

Fig. 12. Learning information window of the HUBO simulation

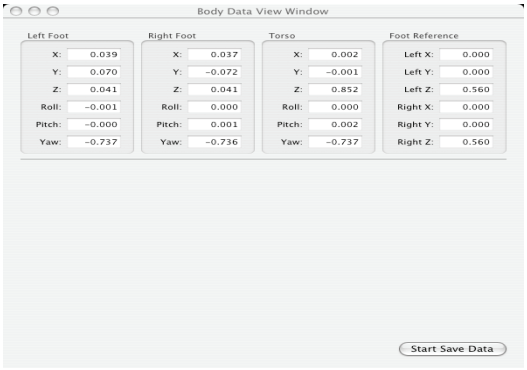

Fig. 13. Body data window of the HUBO simulator

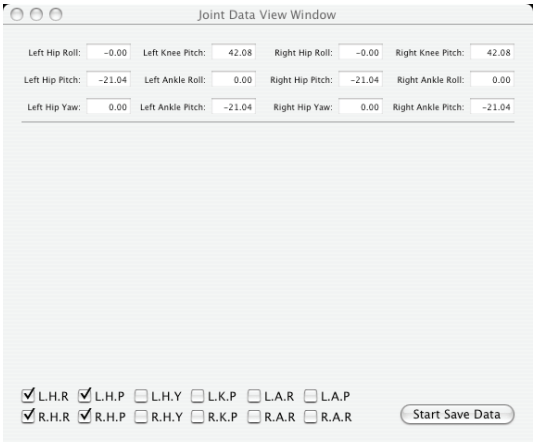

Fig. 14. Joint data window of the HUBO simulator

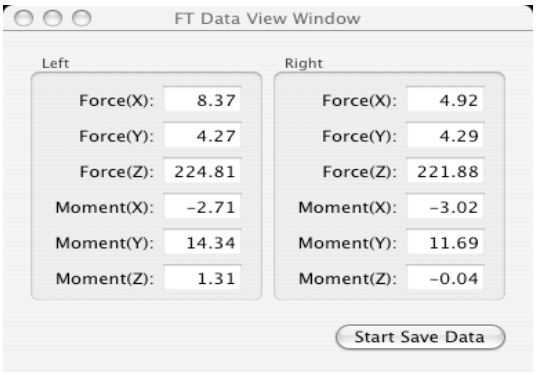

Fig. 15. Force-Torque data window of the HUBO simulator

4.2 States, action and reward

The biped walking pattern generation system can be viewed as a discrete system. Before a new walking step begins, the learning module receives the information of the current states and generates the walking pattern. The robot then follows the generated walking pattern. Following this, the walking pattern is finished and the process starts again. Therefore, this system can be viewed as a discrete system in which the time step is the walking pattern period or the step period. In this study, the walking pattern starts at the moment of the SSP (Single Support Phase) and ends at the moment of the next SSP. At the beginning of the SSP, the learning module receives the current states and calculates the action. Simultaneously, an evaluation of the former action is carried out by the learning module.

4.2.1 Sagittal plane

To set up proper states for the sagittal plane motion, a simple inverted model is used. From the linearized inverted pendulum model, the ZMP equation can be formulated, as shown in Eq. 4-1.

$$ZMP(t) = x(t) - \frac{l}{g}\ddot{x}(t) \tag{5}$$

From Eq. 4-1, the position and acceleration of the mass center is directly related to the ZMP. As the ZMP is related to the stability of the biped walking system, it is feasible to select the position and acceleration of the mass center as states. In addition, to walk stably with minimal energy consumption, the robot should preserve energy, implying that the robot should utilize its momentum (angular or linear). The momentum reflects current and future states; it is related to the velocity of the mass center. Therefore, the velocity of the mass center was chosen as the state in this study. Selected states and the reasons for their selection are summarized in Table 4-1.

All states are normalized to -1.0 ~ 1.0. However, the reinforcement learning agent has no data regarding the maximum values of the states. It receives this data during the training and updates it automatically. First, these maximum values are set to be sufficiently small; in this research, the value is 0.1. The reinforcement learning agent then updates the maximum value at every step if the current values are larger than the maximum values.

State	Reason
The position of the mass center with respect to the support foot	Relation between the position of the mass center and ZMP and the body posture
The velocity of the mass center	Angular or linear momentum
The acceleration of the mass center	Relation between the position of the mass center and ZMP

Table 2. States for the sagittal plane motion

The learning parameter learnt through reinforcement learning is the final velocity. It is an

unknown parameter in the initial design of the walking pattern. The boundary conditions of the walking pattern were discussed in Chapter 3. Eq. 4-2 shows these conditions again.

$$
\begin{aligned}
&X(t) = at^3 + bt^2 + ct + d && \text{Walking pattern} \\
&\text{When } t = 0 \\
&X = \textit{current position} && \text{Condition 1} \\
&\dot{X} = \textit{current velocity} && \text{Condition 2} \\
&\text{When } t = T \\
&X = \textit{final position} && \text{Condition 3} \\
&\dot{X} = \textit{final velocity} && \text{Unknown parameter}
\end{aligned}
\tag{6}
$$

From Eq. 6, Conditions 1 and 2 are determined from the former walking pattern and Condition 3 is the given parameter (the desired step size) from the user. However, only the final velocity is unknown, and it is difficult to determine this value without precise analysis. Hence, the action of the reinforcement learning system is this final velocity (Table 4-2).

Action	Reason
Final velocity of the walking pattern	Only the final velocity is unknown parameter and it is related to the stable walking

Table 3. Action for the sagittal plane motion

The reward function should be the correct criterion of the current action. It also represents the goal of the reinforcement learning agent. The reinforcement learning agent should learn to determine a viable parameter value for the generation of the walking pattern with the goal of stable walking by the robot. Accordingly, in this research, the reward is 'fall down or remain upright' and 'How good is it?' Many candidates exist for this purpose, but the body rotation angle (Fig. 4-7) was finally chosen based on trial and error. Table 4-3 shows the reward and associated reasons. If the robot falls down, the reinforcement learning agent then gives a high negative value as a reward; in other cases, the robot receives positive values according to the body rotation angle. The pitch angle of the torso represents the feasibility of the posture of the robot.

Reward	Reason
Fall down	This denotes the stability of the robot(or absence of stability)
Pitch angle of the torso	It represents how good it is for stable dynamic walking

Table 4. Reward for the sagittal plane motion

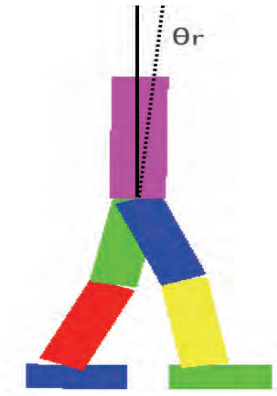

Fig. 16. Pitch angle of the torso (This angle is used as the reward in the sagittal motion)

Fig. 4-8 shows the overall structure of the generation of the walking pattern using reinforcement learning. The reinforcement learning system receives the current states, calculates the proper action, and the walking pattern generator generates the walking pattern based on this action. The reinforcement learning system learns the suitability of the action from its result, and this process is repeated until the reinforcement learning system shows reasonable performance.

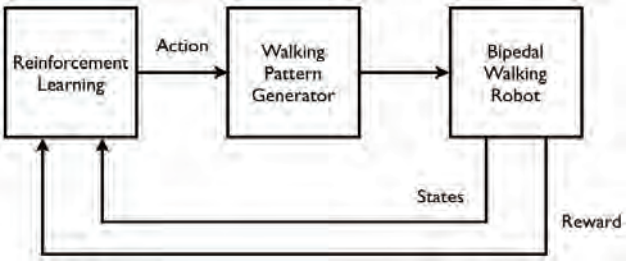

Fig. 17. Diagram of the biped walking pattern generation system

4.2.2 Coronal plane

The coronal plane motion is assumed to be weakly coupled to the sagittal plane motion. Thus, states for the sagittal plane motion are not considered in the coronal plane motion learning implementation. Regarding the inverted pendulum model, the linear dynamic equation for the inverted pendulum can be described as follows:

$$\ddot{y} = \frac{g}{l} y \tag{7}$$

Eq. 4-3 can be integrated to show the relationship between $\dot{y}$ and y :

$$\frac{\dot{y}^2}{2} = \frac{gy^2}{2l} + C \tag{8}$$

Here, C is the integration constant, which is termed the orbital energy [41]. Given the velocity $\dot{y}$ and the position y of the system at any instant, C can be computed. Eq. 4-4 defines the relationship between $\dot{y}$ and y for all other times before the next support exchange event. When C is greater than zero, the mass approaching the vertical plane that passes through the pivoting point will be able to travel across the plane. When C is less than zero, the mass will not be able to travel across the vertical plane. Instead, it reverts to its original direction of travel at some instant. If $\dot{y}$ and y are given, it is possible to predict stability using C. Thus, $\dot{y}$ (the velocity of the torso) and y (the position of the torso with respect to the support foot) are used as states.

As a simple sine function is used in the walking pattern of the coronal plane motion and because the step period is given, the only unknown parameter is the amplitude (sway amount) of the sine function. Thus, the amplitude is used as the action.

The reinforcement learning system for the coronal plane motion adopts a reward that only issues a punishment value when a failure state is encountered:

$$r = \begin{cases} 0 & for\ \Theta_l < \theta < \Theta_u \\ R & failure(otherwise) \end{cases} \tag{9}$$

Here, θ is the roll angle of the torso, which is bounded by Θ_l and Θ_u and R is a negative constant (punishment). It is important to note that the failure condition based on θ is checked at all times.

5. Experiment

To test the logic of the reinforcement learning algorithm, several experiments were carried out. It was assumed that the sagittal plane and the coronal plane motion are weakly related; for this reason, the coronal plane motion was considered first. The reinforcement learning agent for the coronal plane motion learns the stable walking pattern parameter. Based on this learning agent, the reinforcement learning agent for the sagittal plane motion is added. First, step walking was tested, and forward walking tests were performed after these tests.

5.1 Step walking

As step walking does not contain forward motion, it is similar to coronal plane motions.

Hence, the reinforcement learning agent for coronal plane motions learns stable parameters first through step walking experiments. Following this, the learning agent for the sagittal plane motion is added. To determine the suitable parameters, the learning agent for the coronal plane motion learns the sway amount. The update rate α and learning rate γ are set to 0.2 and 0.8, respectively, and the e-greedy value is set to 0.2 initially. The e-greedy value converges to 0 as the learning agent finds the successful parameter.
The experimental conditions are shown in Table 5-1. As shown in Fig. 5-1, step walking converged in 17 of the trails. The parameter for step walking (the sway amount) converges to 0.055m; Fig. 5-2 shows the sideways (y) direction movement of the torso. The movement of the torso is very stable with the period motion. The roll angle of the torso is within 0.4 degree, except for the initial movement, and is very stable, as shown in Fig. 5-3. Fig. 5-4 shows the z-direction movement of the foot, and shows that the robot walks stably without falling down. Additionally, it shows that the DSP time is set to 10% of the step period.

5.2 Forward walking - 15 cm

Based on the learning agent for coronal plane motions that were fully trained through the step walking experiment, a forward walking experiment of 15cm was performed to find a stable forward walking pattern. The experiment conditions are shown in Table 5-2.

As shown in Fig. 5-5, the learning agent learns the proper parameters within 14 trials. The converged final velocity of the walking pattern is 0.3m/sec. The stable walking pattern is identical to that of the forward (x) direction movement of the torso; Fig. 5-6 shows that the walking pattern is very stable. Moreover, the pitch angle of the torso does not exceed 2 degrees (failure condition), as shown in Fig. 5-7. Earlier research used the motion of the swing leg for stable walking while in this research the walking pattern for the support leg is considered. Therefore, it is possible to place the foot in the desired position. Figs 5-8 and 5-9 represent the movement of the foot. As shown in Fig. 5-9, the foot is located in the desired position (0.15m).

5.3 Forward walking - 20 cm

To test the robustness of the reinforcement learning agent for sagittal plane motions, an additional forward walking test was performed. In this test, the learning agent determined a stable parameter within 21 trials, and the pitch angle of the torso was within 1.3 degrees while walking. Fig. 5-14 shows that the foot is placed in the desired position.

Step period	1.0 sec
Step length	0.0 m
Lift-up	0.06 m
DSP time	0.1 sec
Update rate	0.2
Learning rate	0.8
Initial e-greedy	0.2

Table 5. Experiment condition for step walking

Step period	1.0 sec
Step length	0.15 m
Lift-up	0.06 m
DSP time	0.1 sec
Update rate	0.2
Learning rate	0.7
Initial e-greedy	0.1

Table 6. Experiment condition for forward walking (15 cm)

Step period	1.0 sec
Step length	0.2 m
Lift-up	0.06 m
DSP time	0.1 sec
Update rate	0.2
Learning rate	0.8
Initial e-greedy	0.1

Table 7. Experiment condition for forward walking (20 cm)

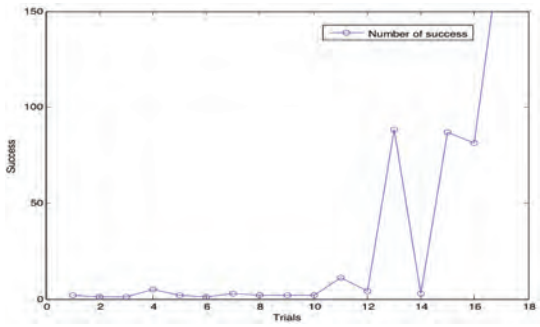

Fig. 18. Iteration and success in the Coronal plane motion

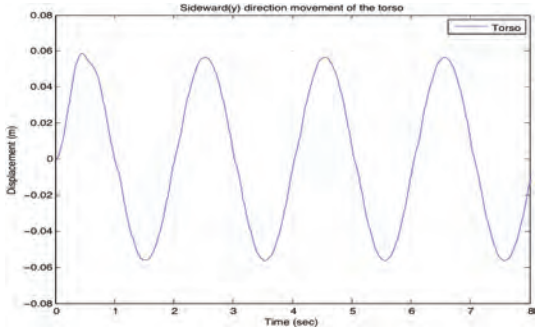

Fig. 19. Sideward(y) direction movement of the torso

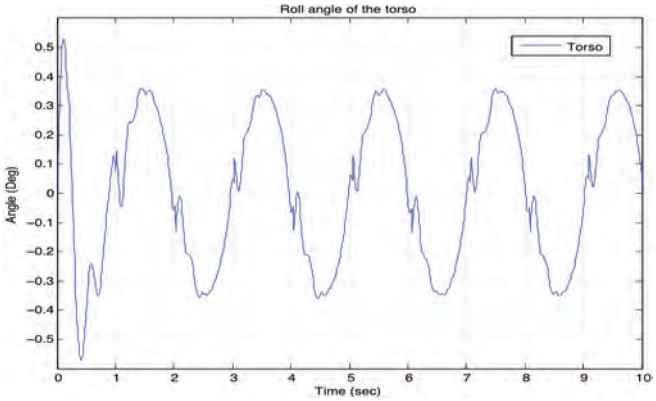

Fig. 20. Roll angle of the torso

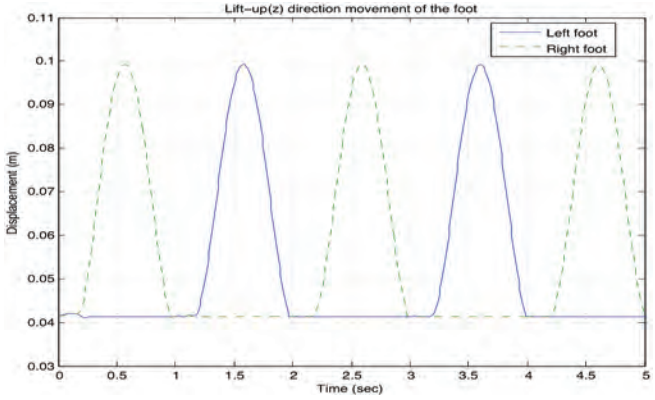

Fig. 21. Lift-up(z) direction movement of the foot

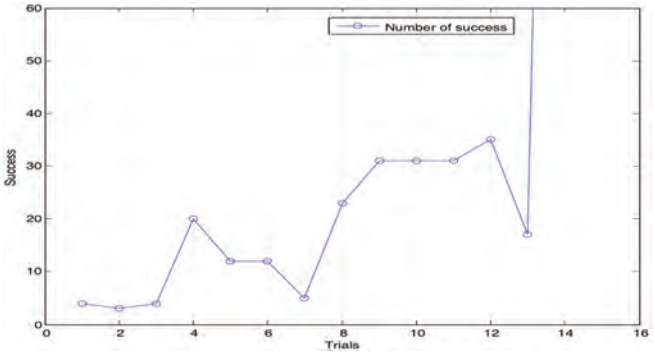

Fig. 22. Iteration and success (15cm forward walking)

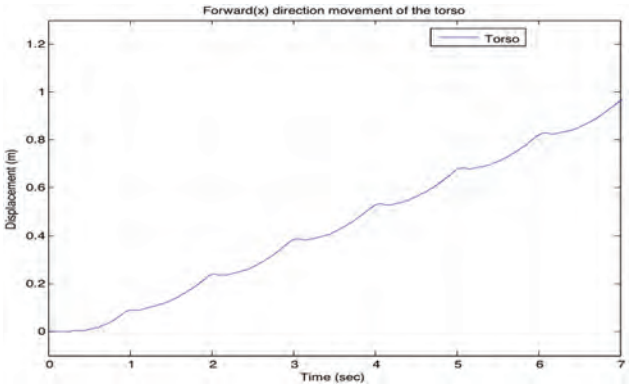

Fig. 23. Forward(x) direction movement of the torso (15cm forward walking)

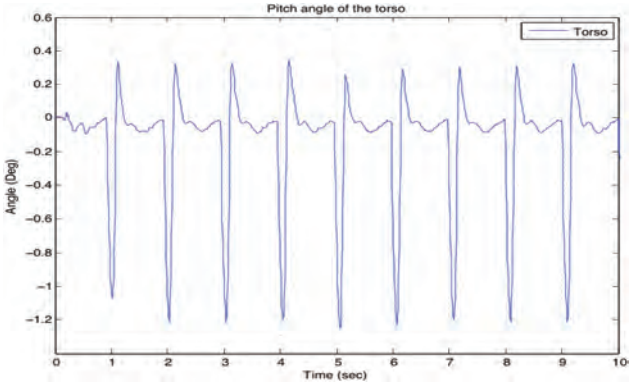

Fig. 24. Pitch angle of the torso (15cm forward walking)

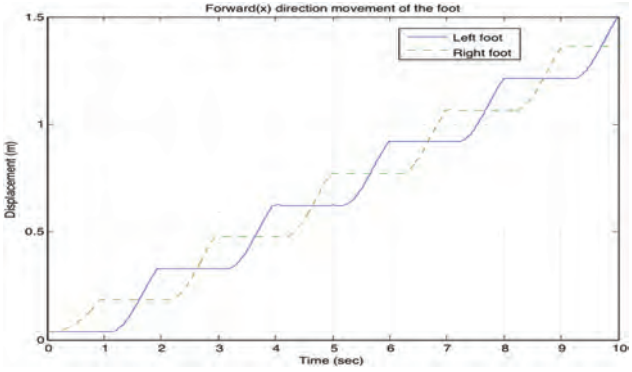

Fig. 25. Forward(x) direction movement of the foot (15cm forward walking)

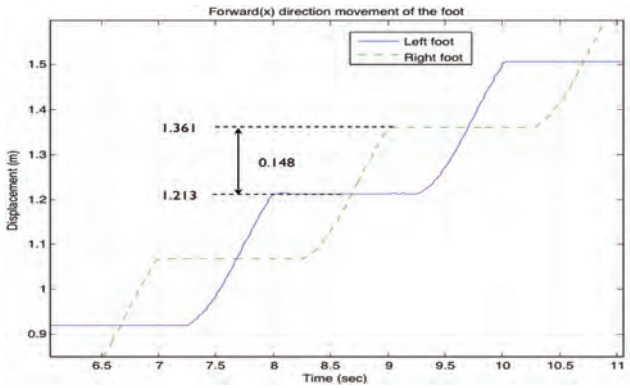

Fig. 26. Position of the foot (15cm forward walking)

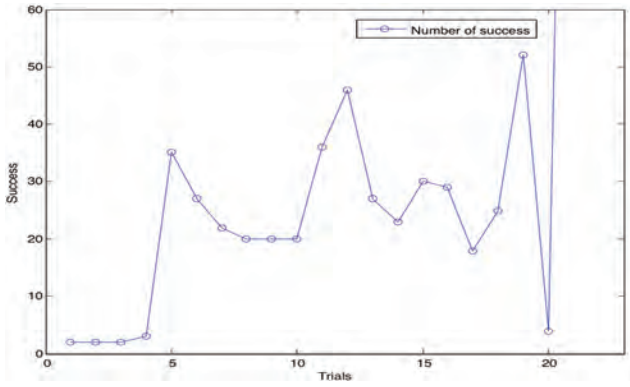

Fig. 27. Iteration and success (20cm forward walking)

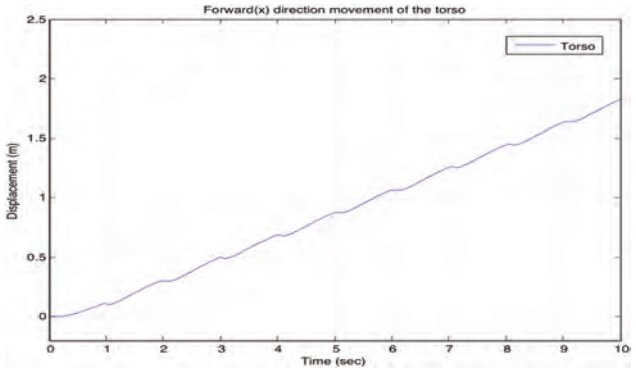

Fig. 28. Forward(x) direction movement of the torso (20cm forward walking)

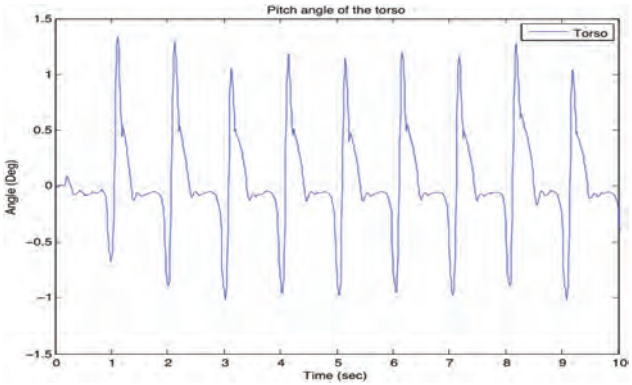

Fig. 29. Pitch angle of the torso (20cm forward walking)

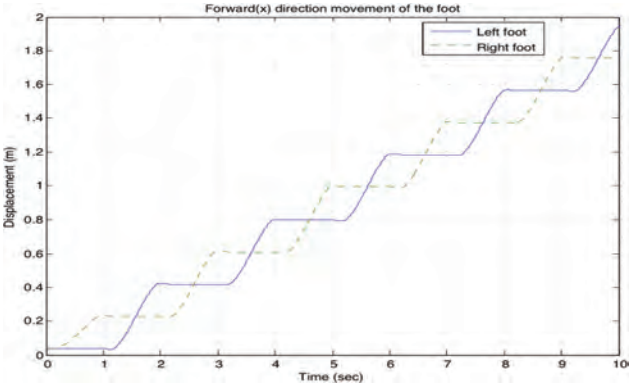

Fig. 30. Forward(x) direction movement of the foot (20cm forward walking)

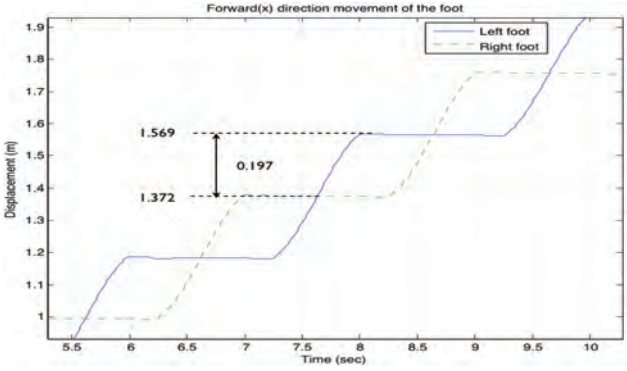

Fig. 31. Position of the foot (20cm forward walking)

7. Conclusion

The main purpose of this research is to generate the stable walking pattern. There are many methods about stable biped walking but these methods can be categorized into two groups. One is the 'inverted model based control method'. In this method, a simple inverted pendulum model is used as a biped walking model. Based on this model, a proper ZMP reference is generated and a ZMP feedback controller is designed to follow this reference. A second method is known as the 'accuracy model method'. This model requires an accurate model of the biped walking robot and its environment. In this method, a stable walking pattern is generated in advance based on the abovementioned accurate model and the biped walking robot follows this walking pattern without a ZMP feedback controller. One advantage of this method is that it allows control of the biped walking robot with a desired posture. Additionally, it does not require a ZMP controller. Each method has its own strength and weakness but in this research, the 'accuracy model method' is used.

However, a problem with the 'accuracy model method' involves difficulty in obtaining an accurate model of the robot and its environment, including such factors as the influence of the posture of the robot, the reaction force from the ground, and so on. Consequently, the generated walking pattern should be tuned by experiments. The generated walking pattern for a specific environment is sensitive to external forces, as this method does not include a ZMP controller. Also the tuning process takes much time and needs an expert.

In this research, reinforcement learning is used to solve this problem. Reinforcement learning is different from supervised learning, the kind of learning in most current research in machine learning, statistical pattern recognition, and artificial neural network. In point of view treating nonlinear problems which are finding optimal solution under given environment, supervised learning is similar to reinforcement learning. But this process is only possible to finding good solution if and if only there are good examples provided by external supervisor. But reinforcement learning learns this process without the external supervisor so if the system model is unknown or partially known, reinforcement learning will be good choice.

Reinforcement learning is based on trial-and-error methodology. It can be hazardous to apply a reinforcement learning system to an actual biped walking system before the learning system is trained sufficiently through many trials, as walking system likely has not been fully analyzed by the learning system. In particular, when such a system is inherently unstable, such as in the case of a biped walking robot, attention to detail is essential. Therefore, it is necessary to train a learning system sufficiently before applying it to a real system. For this reason, the HUBO simulator is developed. The HUBO simulator contains the physics engine, the learning system, the generalization module and other utility modules. The physics engine contains the environment model not only the robot model. So it is possible to simulate the interaction between the robot and its environment such as the ground. Its structure is modulated and simple, it is easy to update or add its functions or algorithm.

In this research, it is developed the stable biped walking pattern generation algorithm using reinforcement learning based on the HUBO simulator. Unlike former researches, the walking pattern of the support leg is considered. Existing researches use the motion of the swing leg for stable biped walking and extra controllers are needed for controlling the motion of the support leg. But because the algorithm developed in this research is for the walking pattern of the support leg and the motion of the swing is determined by the given specifications, extra controllers are not needed and overall control structure is very simple. Also this algorithm generates the stable biped walking pattern automatically, supervisors or experts are not needed.

Also algorithms developed by former researches were limited to the planar planed robot system, but the algorithm developed through this research considers the sagittal and the coronal plane motion. Former researches considered the sagittal plane motion only and the coronal plane motion was neglected. In this research, it is assumed that the sagittal and the coronal plane motion are weakly coupled. So the reinforcement learning systems for the each plane are trained separately and after the sufficient learning process, two learning systems are combined.

Through several experiments, it is validated the performance of the stable biped walking pattern generation algorithm. Several experiments are accomplished using the HUBO simulator and proper states and reward function for stable biped walking are founded. The algorithm is converged stably and its performance is superior and convergence time is faster than existing researches.

Although this research does not contain the experiment that contains the real system, the logic of the algorithm is tested and verified using the HUBO simulator. And the performance of the algorithm is distinguishable compare to the existing researches. But it is necessary to test or apply the algorithm to the real system and following research will contain this. Also the algorithm is tested to forward walking only but in the following research various motions such as side walking and asymmetric biped walking will be tested.

Contributions	Future work
The HUBO simulator is developed which contains the physics engine, the learning system, the generalization module and other utility modules.	It is needed to apply and test the algorithm to the real system.
Proper states and reward function are founded for the reinforcement learning system.	It is necessary to test various motions such as side walking and asymmetric biped walking.
The third order polynomial walking pattern for the motion of support leg is generated using reinforcement learning.	
In contrast to former researches, the sagittal and coronal plane motion is considered.	

Table 8. Contributions and future work

8. Reference

A. A. Frank and M. Vokobratovic, *'On the Gait Stability of Biped Machine'*, IEEE Transactions on Automatic Control, December, 1970.

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.

K. Hirai, *'Current and Future Perspective of Honda Humanoid Robot'*, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, p500-p508, 1997.

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.

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.

A. Takanishi, M. Ishida, M. Yamazaki and I. Kato, *'The Realization of Dynamic Walking by the Biped Walking Robot WL-10RD'*, ICRA 1985, 1985.

Jung-Yup Kim, Ill-Woo Park, and Jun-Ho Oh, *'Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor'*, Journal of Intelligent and Robotic Systems, Accepted, 2006.

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.

Jung-Yup Kim, *'On the Stable Dynamic Walking of Biped Humanoid Robots'*, Ph. D Thesis, Korea Advanced Institute of Science and Technology, 2006.

KangKang Yin, Kevin Loken and Michiel van de Panne, *'SIMBICON: Simple Biped Locomotion Control'*, ACM SIGGRAPH 2007, 2007.

Jung-Yup Kim, Jungho Lee and Jun Ho Oh, *'Experimental Realization of Dynamic Walking for the Human-Riding Biped Robot, HUBO FX-1'*, Advanced Robotics, Volume 21, No. 3-4, p461-p484, 2007.

Jun-Ho Oh, David Hanson, Won-Sup Kim, Il Young Han, Jung-Yup Kim, and Ill-Woo Park, *'Design of Android type Humanoid Robot Albert HUBO'*, in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijin, China, 2006.

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.

Chew-Meng Chew and Gill A. Pratt, *'Dynamic Bipedal Walking Assisted by Learning'*, Robotica, Volume 20, p477-p491, 2002.

Hamid Benbrahim and Judy A. Franklin, *'Biped Dynamic Walking Using Reinforcement Learning'*, Robotics and Autonomous Systems, Volume 22, p283-p302, 1997.

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.

E. Schuitema, D. G. E. Hobbelen, P. P. Jonker, M. Wisse and J. G. D. Karssen, *'Using a ControllerBased on Reinforcement Learning for a Passive Dynamic Walking Robot'*, Proc. of IEEE-RAS International Conference on Humanoid Robots, Tsukuba Japan, 2005.

Jong-Hwan Kim, Kui-Hong Park, Jun-Su Jang, Yong-Duk Kim, Bum-Joo Lee and Ki-Pyo Kim, *'Humanoid Robot HanSaRam: Schemes for ZMP compensation'*, Proc. of International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore, 2003.

Dusko Katic and Miomir Vukobratovic, *'Control Algorithm for Biped Walking Using Reinforcement Learning'*, 2nd Serbian-Hungarian Joint Symposium on Intelligent Systems, 2004.

Naoto Shiraga, Seiichi Ozawa and Shigeo Abe, *'A Reinforcement Leaning Algorithm for Neural Networks with Incremental Learning Ability'*, Proceedings of International Conference Neural Information Processing, 2002.

William Donard Smart, *'Making Reinforcement Learning Work on Real robots'*, Ph. D. Thesis, Brown University, 2002.

Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi and Hirohisa Hiukawa, *'Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point'*, Proceedings of the 2003 IEEE International Conference on Robotics & Automation, p1620-p1626, 2003.

I. Kato, S. Ohteru, H. Kobayashi, K. Shirai and A. Uchiyama, 'Information Power Machine with Senses and Limbs', First CISM-IFToMM Symposium on Theory and Practice of Robots and manipulators, 1974.

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.

Richard S. Sutton and Andrew G. Barto, *'Reinforcement Learning: An Introduction'*, The MIT Press, 1998.

Lee, Y., Jung, T., *'Reinforcement Learning'*, Kyunghee Univ., 2005.

Watkins, C. J. C. H., *'Learning from Delayed Rewards'*, Doctoral Thesis, Cambridge University, 1989.

William Donald Smart, *'Making Reinforcement Learning Work on Real Robots'*, Ph .D Thesis, Brown University, 2002.

J. S. Albus, *'Theoretical and experimental aspects of a cerebellar model'*, PhD. Dissertation, University of Maryland, 1972.

J. S. Albus, *'Data storage in the cerebellar model articulation controller'*, Journal of Dynamic Systems, Measurement and Control, pp. 228-233, 1975.

J. S. Albus, *'Brains, behavior, and robotics'*, Peterborough, N.H.: Byte Books/McGraw-Hill, 1981.

Hirohisa Hirukawa, Fumio Kanehiro and Shuuji Kajita, *'OpenHRP: Open Architecture Humanoid Robotics Platform'*, Robotics Research: The Tenth International Symposium, Volume 6, p99-p112, 2003.

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.

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.

C. Angulo, R. Tellez and D. Pardo, *'Emergent Walking Behaviour in an Aibo Robot'*, ERCIM News 64, p38-p39, 2006.

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.

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.

Russel Smith, *'www.ode.org/ode.html'*, 2007.

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.

Olivier Michel, *'Webots: Professional Mobile Robot Simulation'*, International Journal of Advanced Robotic System, Volume 1, Number 1, p39-p42, 2004.

Shuuji Kajita, Kazuo Tani and Akira Kobayashi, *'Dynamic walk control of a biped robot along the potential energy conserving orbit'*, IEEE International Conference on Intelligent Robots and Systems, p789-p794, 1990.

J. K. Hodgins, M. H. Raibert, *'Biped Gymnastics'*, The International Journal of Robotics Research, Volume 9, No. 2, 1990.

M. H. Raibert, *'Hopping in Legged Systems-Modeling and Simulation for the Two Dimensional One Legged Case'*, IEEE Transactions on Systems, Man and Cybernetics, Volume SMC-14, No. 3, 1984.

J. Yamaguchi , E. Soga, S. Inoue and A. Takanishi, *'Development of a Bipedal Humanoid Robot-Control Method of Whole Body Cooperative Dynamic Biped Walking'*, Proc. of the ICRA 1999, p368-p374, 1999.

M. Gienger, *'Toward the Design of a Biped Jogging Robot'*, Proc. of ICRA 2001, p4140-p4145, 2001.

T. Sugihara, Y. Nakamura and H. Inoue, *'Realtime Humanoid Motion Generation through Manipulation based on Inverted Pendulum Control'*, Proc. of the ICRA 2002, p1404-p1409, 2002.

J. Pratt, P. Dilworth and G. Pratt, *'Virtual Model Control of a Bipedal Walking Robot'*, Proc. of the ICRA 1997, p1476-p1481, 1997.

A. Takanishi, M. Tochizawa, H. Karaki and I. Kato, *'Dynamic Biped Walking Stabilized With Optimal Trunk and Waist Motion'*, Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems, p187-p192, 1989.

A. Takanishi, H. Lim, M. Tsuda and I. Kato, *'Realization of Dynamic Biped Walking Stabilized by Trunk Motion on a Sagittally Uneven Surface'*, Proc. IEEE International Workshop on Intelligent Robots and Systems, p323-p330, 1990.

Q. Li, A. Takanishi and I. Kato, *'A Biped Walking Robot Having a ZMP Measurement System Using Universal Force-Moment Sensors'*, Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems, p1568-p1573, 1991.

Y. Ogura, Y. Sugahara, Y. Kaneshima, N. Hieda, H. Lim and A. Takanishi, *'Interactive Biped Locomotion Based on Visual / Auditory Information'*, Proc. IEEE International Workshop on Robot and Human Interactive Communication, p253-p258, 2002.

N. Kanehira, T. Kawasaki, S. Ohta, T. Isozumi, T. Kawada, F. Kanehiro, S. Kajita and K. Kaneko, *'Design and Experiments of Advanced Leg Module(HRP-2L) for Humanoid Robot(HRP-2) Development'*, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, p2455-p2460, 2002.

G. Pratt and M. M. Williamson, *'Series Elastic Actuators'*, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, p399-p406, 1995.

Jerry E. Pratt, *'Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots'*, Ph. D. Thesis, Massachusetts Institute of Technology, 2000.

T. Ishida, Y. Kuroki and T. Takahashi, *'Analysis of Motion of a small Biped Entertainment Robot'*, Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, p144-p147, 2004.

Claude F. Touzet, *'Neural Network and Q-Learning for Robotics'*, Proceedings of International Joint Conference on Neural Network, 1999.

Richard S Sutton, *'Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding'*, Advances in Neural Information Processing System, Volume 8, p1038-p1044, 1996.

R. Matthew Kretchmar and Charles W. Anderson, *'Comparison of CMACs and Radial Basis Functions for Local Function Approximation in Reinforcement Learning'*, Proceedings of International Conference on Neural Network, 1997.

Juan Carlos Santamaria, Richard S. Sutton and Ashwin Ram, *'Experiments with Reinforcement Learning in Problems with Continuous State and Action Space'*, COINS, p96-p88, 1996

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.