DOCTORAL THESIS PROPOSAL

# Biped Locomotion: Augmenting an Intuitive Control Algorithm with Learning

By

CHEE-MENG CHEW
Rm NE43-006

Submitted to the

Department of Mechanical Engineering
Massachusetts Institute of Technology

NOVEMBER 9, 1999

Thesis Committee:
Prof. Jean-Jacques E. Slotine, Chairman, Rm 3-338
Prof. Gill A. Pratt, Rm NE43-006
Prof. Tomaso Poggio, Rm E25-218

# Doctoral Thesis Proposal

# Biped Locomotion: Augmenting an Intuitive Control Algorithm with Learning

Chee-Meng Chew
Department of Mechanical Engineering, Massachusetts Institute of Technology
Cambridge, MA 02139

November 9, 1999

**Abstract**

*Foot placement is a key determinant for the stabilization of walking speed and lateral motion of a biped. However, there is no closed form expression for the foot placement parameters in term of the walking speed or other gait parameters. A simple and intuitive control algorithm (called "Turkey Walking") based on Virtual Model Control (VMC) was successfully applied to planar bipedal walking. However, it has deficiencies that pertain to the foot placement problem. I propose augmenting the algorithm with reinforcement learning (RL) algorithms to overcome the deficiencies. No dynamic model is required for the overall control architecture. The RL algorithms are used to learn the leg swing parameters, whereas the "Turkey Walking" (TW) algorithm generates the desired joints' torque based on the gaits' parameters. The control architecture is first tested on planar bipeds. After that, it is extended to a 3D biped. The research emphasizes on establishing the detailed structure of the architecture and illustrating the generality of the approach by simulation analysis.*

## 1 Background

It is a great challenge for scientists and engineers to build a bipedal robot that can have the similar agility or mobility of a human counterpart. The complexity of bipedal robot control is due to the nonlinear dynamics, unknown environment interaction and limited torque at the stance ankle.

Pratt, et al. (1997) have proposed a simple and low computation algorithm called *Turkey Walking* (TW) for planar bipedal walking which is based on *Virtual Model Control* (VMC). One distinct feature of this algorithm is that no dynamic equations are used to control the robot. Together with a simple finite state machine, the algorithm was successfully implemented to control a planar biped for steady walking task on level ground in real-time. Chew (1998) has extended the algorithm for rough terrain locomotion.

However, there are two main deficiencies of the *Turkey Walking* algorithm. Firstly, the walking speed is controlled or modulated only during the double support phase. This mode of speed control is effective only if the double support phase has a significant duty factor. To have a significant duty factor, the swing leg needs to move quickly to the next touchdown position. This demands high bandwidth and maximum torque output of the actuators. Hence, for a given actuator bandwidth and maximum torque output, the biped's walking speed will not be able to reach the upper bound, which is achievable by a biped that walks without requiring any double support phase for speed control.

To achieve a stable gait in the sagittal plane without using the double support phase control, swing leg control becomes critical. Since bipedal locomotion has the inherent property of having unpowered d.o.f. in the single support phase (Vukobratovich, et al. 1990) which cause the bipedal system to have restricted controllability in the phase, we are not able to command the joints of the bipedal robot to track any prescribed trajectory as in the manipulator control. Thus, it is difficult to generate a swing leg trajectory or strategy that results in stable walking cycle.

The approach to solve the problem is usually by indirect approaches. For example, one may resort to intuition based on simple models. A common approach is to assume an inverted pendulum model and plan the swing leg trajectory independently. However, this model does not consider the coupling effect between the motion of the main body and the swing leg dynamics. The coupling effect increases when the walking speed is increased (especially if the legs are massive). Therefore, such an approach is not applicable for a wide range of walking speed.

The second deficiency is that since the Turkey Walking algorithm was designed for a planar biped, it only provides the sagittal plane motion control solution. It did not deal with lateral balancing problem that is faced by a general three-dimensional biped. Thus, the Turkey Walking algorithm is insufficient to enable three-dimensional bipedal walking.

Although the Turkey Walking algorithm has these deficiencies, it has nice properties like simplicity and low computation. It is thus critical to supplement the algorithm with other methodologies that can compensate for these deficiencies. In fact, the deficiencies can be lumped into a single problem: “How to perform foot placement or swing leg task so as to achieve stable walking?” That is, if we can find an approach to perform the swing leg task so as to achieve stable locomotion, the deficiencies can be removed. This motivates me to study the issue of the foot placement or the swing leg control task and to find a methodology that solves the problem. I wish to augment the *Turkey Walking* algorithm with the methodology for the foot placement task. The overall control architecture should be general enough to apply to a wide range of bipedal robots that have similar structure but different inertia parameters.

Foot placement is a key determinant for the stabilization of walking speed and lateral motion of a biped. However, there is no closed form expression for the foot placement parameters in term of the walking speed or other gait parameters. This is because of the unpowered d.o.f. and the fact that the stance leg’s foot is not bolted on the ground (compared with traditional manipulators).

Since there is no analytical solution for the foot placement task, I propose using learning methodology for it. Model-free reinforcement learning (RL) algorithms (Kaelbling, Littman and Moore, 1996) will be used. The robot is expected to learn the swing leg task for stable walking without a dynamic model of the system or the environment. Other subtasks like height and body posture control can be achieved by using the Turkey Walking algorithm. This results in a control architecture for bipedal walking which I call “TW-RL”.

There are two advantages of this architecture. Firstly, the learning time is expected to be shorter than those learning approaches that learn all the joints’ trajectories from scratch. This is because the scope of learning is small and less complex if the learning algorithms just focus on a subtask. Furthermore, I can utilize the previous research results of the Turkey Walking algorithm and thus not invent the wheel. In summary, the robot should only need to learn whatever tasks that really require learning.

Such an architecture should not require any existing data to prime the algorithm. Thus, it eradicates the process of data adaptation. The robot can learn the foot placement behaviors by trial-and-error without using the dynamic model. Thus, the architecture is general and the resulting algorithm should be applicable to another bipedal robot having the same d.o.f. but different mass distributions, length parameters, actuator power etc..

The architecture is also general in that it allows a programmer to freely choose the strategy for control, for example, the swing leg strategy. Then, the learning algorithms learn the appropriate parameters for the foot placement so as to achieve overall stable walking.

Since the algorithm should eventually be applied to a physical robot, it should be able to run in real-time. Furthermore, the number of trials needed for the robot to learn to walk successfully should be low. Therefore, computation requirement and rate of learning are two key considerations for the implementations.

The architecture is applied to simulated bipedal robots that are constrained to move in the sagittal plane. I will illustrate and discuss several ways of implementing the architecture for steady walking. Then, I will extend it to a 3D bipedal robot.

## 2 Related Work

In biomechanics research, Redfern and Schumann (1994) analyzed the swing trajectory of the foot with respect to the pelvis during the walking gait. They also proposed a model of foot placement control for stable base support of human locomotion. Experimental data were collected to test the model during walking trials of different speeds. Townsend (1985) found that lateral stability of human walking is maintained through foot placement based on conditions at each new step. The body center of mass trajectory can be controlled by foot placement alone. Beckett and Chang (1968) investigated the behavior of the leg in the swing phase during normal walking. They also studied the energy consumption of the walking and concluded that there is a natural gait at which a person can walk with minimum effort.

The researches listed above were carried out for human beings. It is not clear whether the model derived for human walking can be generally applied to other bipeds or used for the control of a bipedal robot. One reason is that a bipedal robot usually has different mass distribution and length parameters from humans. They also have different actuators system.

Raibert and Wimberly (1984) applied tabular control of balance to a planar hopping monoped. A tabular relationship between the lift-off and touchdown states was first obtained by simulating the monoped from different initial conditions. The foot placement location that minimized a performance index could be found by one-dimensional search through the tabulated data. One drawback of this approach is that the tabular relationship was done in simulation. It is unclear whether the information obtained could be applied to the physical system.

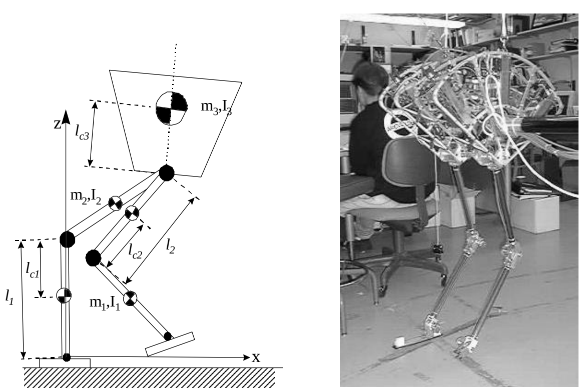
Kajita and Tani (1995) derived a massless leg model for a planar biped that followed linear motion. From the model, the touchdown condition of the swing foot could be determined based on an energy parameter.

Benbrahim and Franklin (1997) applied reinforcement learning algorithms for a biped to achieve dynamic walking. They adopted a "melting pot" and modular approach in which a central controller used the experience of other peripheral controllers to learn an average control policy. The central controller was pre-trained to provide nominal trajectories to the joints. One disadvantage of the approach was that nominal trajectories of the joints were required before any learning began.

Miller (1994) designed a learning system for a biped that was capable of learning the balance for side-to-side and front-to-back motion. Neural network learning provided accurate and smooth feedforward control to the joints.

## 3 Target Systems

Two bipeds are considered in this research. One is a headless and armless planar bipedal robot called *Spring Flamingo* (Figure 1). It is constrained to move in the sagittal plane. The legs are much lighter than the body. Each leg has three actuated rotary joints. The joints axes are perpendicular to the sagittal plane. It has a total weight of about *14 kg*.




(a) (b)

Figure 1. 7-Link 6 dof planarbiped – Spring Flamingo

The other robot is an unconstrained bipedal robot called *M2* (Figure 2). It is also headless and armless. Each leg has six active d.o.f. of which three d.o.f. is available at the hip (yaw, roll, pitch), one at the knee (pitch) and two at the ankle joint (pitch, roll). It has a total weight of about *23 kg*.

In both systems, the joint actuators are *Series Elastic Actuators* (Pratt and Williamson, 1995). The actuators are force or torque controlled. The inputs to the actuators are the desired force or torque, generated by the walking algorithm, for individual joints.

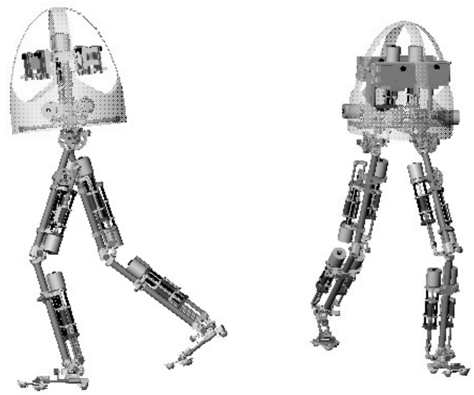


Figure 2. Three dimensional biped: M2 (Picture courtesy from Danial Perluska)

## 4 Approach

For normal walking, we assume that the dynamics in the sagittal and the frontal plane are decoupled. Based on this assumption, we consider the motion control in both planes separately.

For the sagittal plane, the bipedal walking task can be decomposed into three subtasks. They are the height control of the body, the pitch control of the body, and the horizontal velocity of the body. It is easy to achieve the first two subtasks using the Turkey Walking algorithm (Section 4.1) without the need of learning.

As for the horizontal velocity regulation, I propose using the foot placement approach instead of using the double support phase as in Pratt, et al. (1997). Since there is no analytical solution for the foot placement in terms of walking speed or the gait parameters, I apply a reinforcement learning algorithm (Section 4.2) to learn the task. The behavior of the swing leg can be determined binseveral ways. For example, we could specify the swing foot trajectory in the spatial coordinates before each swing task. However, the resulting solution space is too huge to apply general learning. To obtain a well-posed learning problem, we could narrow the scope of learning by fixing the shape of the swing foot trajectory and learning only the scalar parameters of the swing leg. For example, the biped can learn the end-position of the swing foot relative to the hip and/or the swing time for successful walking.

For the frontal plane, I use rotational a virtual spring-damper component to control the body posture, and a reinforcement learning algorithm learns the lateral behavior of the swing leg. The aim is to prevent the biped from falling in the frontal plane.

The resulting control architecture (called "TW-RL") is summarized in Figure 3. The biped has no prior idea or nominal model for foot placement behavior. The learning agents select appropriate swing leg parameters, for example, step length, swing time etc. and evaluate them accordingly.

### 4.1 Turkey Walking algorithm

The Turkey Walking algorithm is constructed using a control language called Virtual Model Control (VMC) (Pratt, 1995; Pratt et al., 1997). In VMC, we use virtual components to generate joints' torque (or force) commands. The torque generated at the joints creates the same effect that the virtual components would have created, had they existed, thereby creating the illusion that the virtual components are connected to the real robot. Examples of virtual components include simple springs, dampers, masses, etc..

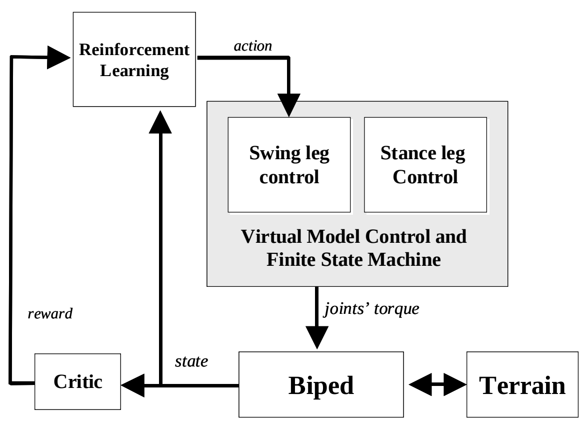



Figure 3: TW-RL: The proposed control architecture.

VMC provides a low computational framework for the control of legged robots since it does not require the computation of inverse dynamics.

Pratt (1995) constructed the Turkey Walking algorithm for a planar biped called "Spring Turkey" that had a similar structure as the planar biped as shown in Figure 1. In this algorithm, the height control is achieved by a virtual parallel spring-damper attached vertically between the hip and the ground. It generates a virtual vertical force $F_z$ at the hip. For the body pitch control, a virtual rotational spring-damper is applied at the hip. This generates a virtual torque $M_\alpha$ about the hip. The virtual forces are then be transformed into the joints' torque using a transformation matrix (Pratt, 1995) for postural control.

### 4.2 Reinforcement learning (RL)

A characteristic of reinforcement learning (RL) algorithms is that the learning is done with the help of a critic rather than a teacher (Sutton and Barto, 1998). This is distinct from supervised learning where a set of input-output data are used to train a learner. Most of the time, the RL algorithms learn an intermediate function (called the value function) from which decisions or control actions are deduced. One feature of the RL approach is that non-linearities due to bandwidth limitation, torque limitation etc. are encapsulated from the learning agents.

There are many algorithms proposed for RL. One popular algorithm is the Q-learning algorithm by Watkins (1992). It is an off-policy and model-free RL algorithm. A brief outline of the algorithm is given as follows.

Let $s$ denote the state, $a$ denote the present action; and $Q(s,a)$ be the state-action value function corresponding to $s$ and $a$. The key equation of the *Q-learning* algorithm is as follow (Sutton and Barto 1998):

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)]$$

where the subscript indicates the stage number, $r_{t+1}$ denotes reward received after an action $a_t$ is taken at $t$-th stage when the system is in state $s_t$; $\alpha$ denotes the step-size parameter for the update, $\gamma$ denotes the discount rate in the computation of cumulative rewards (return). The equation provides an update of value estimate $Q(s_t,a_t)$ based on the observed $r_{t+1}$ and the value estimate $\max_a Q(s_{t+1},a)$.

To apply the Q-learning to continuous states and actions, a function approximator needs to be used to approximate $Q(s,a)$ (Sutton and Barto, 1998). In this research, I compare two function approximators: 1) the Gaussian Radial Basis Function (GRBF) network (Broomhead and Lowe, 1988; Poggio and Girosi, 1990) and 2) the Cerebellar Model Articulation Controller (CMAC) (Albus, 1981). Both have nice localized generalization property. Although the GRBF Network has localized generalization and is a universal approximator (Park and Sandberg, 1991), its implementation is computationally expensive. On the other hand, CMAC network has the advantage of having not only localized generalization, but also low computation.

### 4.3 Research Schedule

My research will be first carried out for the planar biped. Two parameters of the swing leg task will be studied. They are the swing time and the end position of the swing foot. We may specify one of them and let the learning agent decide the other. Or, we may let the learning agent decide the proper combination. That is, the biped should be able to execute the swing leg task for steady walking by trial-and-error starting from zero knowledge or skill.

If the result is successful, I will implement the frontal part of the architecture. Together with the sagittal plane implementation, the frontal plane implementation will be applied to the three-dimensional (3D) biped to achieve steady dynamic walking.

When the level ground steady dynamic walking for the 3D biped is successfully implemented, difficult terrain locomotion, e.g. random sloped terrain, stairs etc. will then be considered. The research schedule is summarized as in Table 1.

Table 1: Research Schedule

| *Quarter* | *Task* |
|---|---|
| *Q4 1998* | *Feasibility study of the proposed control architecture for 2D biped.* |
| *Q1 1999 - Q2 1999* | *Comparison between tabular and function approximation approaches in reinforcement learning algorithms.* |
| *Q3 1999* | *Comparison between Gaussian Radial Basis Function and CMAC as function approximator in reinforcement learning algorithms.* |
| | *Finalisation on the choice of function approximator and begin simulation study of the proposed control architecture for 2D biped. For example, learning different parameters of the swing leg task.* |
| *Q4 1999* | *Study of the generality of the control architecture by applying it to two different bipeds (planar).* |
| | *Extension to 3D biped* |
| | *Extension to slope terrain locomotion* |
| *Q1 2000* | *Revise Thesis* |
| *Q2 2000* | *Defend Thesis* |

.

## 5 Preliminary Implementation

The TW-RL control architecture has been implemented for the planar biped. In the implementation, the height of the biped is set and assumed to be constant during walking. The following set of variables is identified to be the state variables:

1. Horizontal velocity of the hip, $v^{+}{}_{hip\_x}$;
2. Horizontal component of the coordinates of the swing leg's ankle from the hip after the swing leg has landed, $d^{+}{}_{ha\_x}$;
3. Actual step length, $d^{+}{}_{step}$.

Superscript + indicates that a state variable is measured or computed momentarily after the landing of the swing foot.

The shape of the swing leg trajectory and the swing time are fixed. The RL algorithm has to learn the desired end position of the swing foot. In this particular implementation, the desired position of the swing foot is measured from the hip.

A simple reward function (critic) is formulated as follows:

$$r = \begin{cases} 0 & \text{for } -0.1 < v_x < 1.3\,{}^{m}\!/\!{}_{s} \\ -1, & \text{otherwise(failure)} \end{cases}$$

I also assign $r = -1$, if the robot falls down.

Q-learning algorithm with CMAC network as the function approximator is used in this implementation. The biped tries to walk until it has failed. The learning is stopped when it has completed 100 seconds of successful walking. Each time it fails, it restarts from beginning.

In this implementation, the learning rate of the system can be improved tremendously when a local control is applied to the stance leg's ankle to assist in regulating the walking speed. The equation to generate the required torque $\tau_a$ at the stance leg's ankle joint is as follow:

$$\tau_a = B_a (v_{hip\_x} - v^{d}{}_{hip\_x}) \qquad (2)$$

where $B_a$ is a constant gain and superscript $d$ indicates desired value. Since the stance leg is not bolted on the ground, the torque needs to be bounded within an upper and a lower limits to prevent the stance leg's foot from tipping at the toe or heel.

To show the effectiveness of the local control, a simulation is run in which the starting posture (legs are parallel to each other) and initial walking speed (0.4m/s) are the same for every trial. The result is shown in Figure 4. The dotted graph shows the simulation result corresponding to the case where the local control was implemented. It reaches 100 seconds walking in less than 30 iterations. The solid-line graph corresponds to the result of the simulation in which no local control is applied. Comparing both graphs, we can deduce that proper application of the local control (in this case, the ankle torque control of the stance leg) can speed up the learning rate for the walking task.

To verify that the local control at the stance leg's ankle joint can provide a consistently good learning rate, we have also randomized the starting posture and walking speed in the simulation. Six consecutive learning results are shown in Figure 5. The biped can reach 100 seconds of walking (without failure) within 120 trials. The best result shows that it can reach the target at 12th trials.

## 6 Potential Contributions

The main contribution of this thesis is the synthesis of the *Turkey Walking* algorithm with the *Reinforcement Learning* (RL) algorithms to provide a simple but general and systematic way of implementing dynamic bipedal walking without the need for dynamics formulation and nominal data. Other contributions include the usage of the local control to assist and speed up the learning for walking task. The proposed control architecture may provide another framework to analyze and understand how bipedal creatures learn to walk.

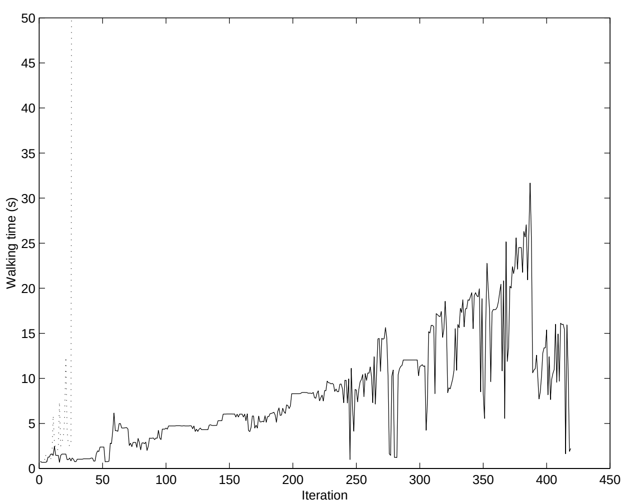



Figure 4. Local control at the stance leg's ankle joint increases the learning rate. The starting posture and initial speed is the same for both case. Dotted curve corresponds to the learning curve with the local control at the ankle. Solid curve is without the local control.

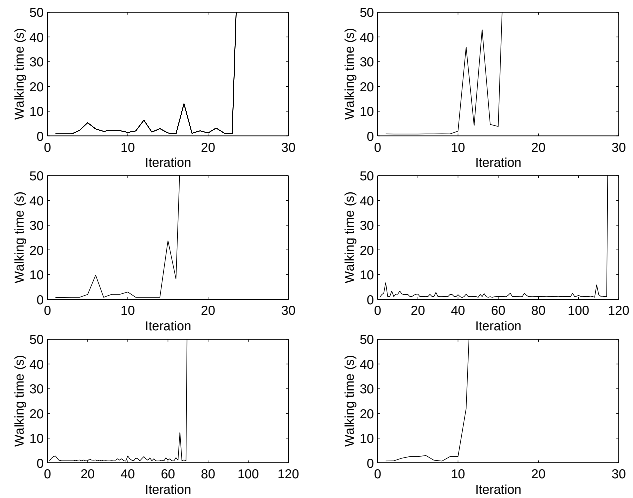



Figure 5. Several learning curves of the biped in which the local ankle torque control is used. In these implementations, the biped randomly selected the starting posture from a set of six postures. The starting velocity is also random.

## References

Albus, J. S. (1981). *Brain, Behavior and Robotics*. Byte Books, Peterborough, NH.

Beckett, R. and Chang, K. (1968). An Evaluation of the Kinematics of Gait by Minimum Energy. *J. Biomechanics*, 1:147-159.

Benbrahim, H. and Franklin, J. A. (1997). Biped Dynamic Walking Using Reinforcement Learning. *Robotics and Autonomous Systems* 22: 283-302.

Broomhead, D.S. and Lowe, D. (1988), Multivariable Functional Interpolation and Adaptive Networks. *Complex Syst.*, 2:321-355.

Chew, C.-M. (1998). *Blind Walking of a Planar Biped over Sloped Terrain*. Master Thesis, Department of Mechanical Engineering, MIT.

Kaelbling, L. P., Littman M. L. and Moore A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* 4: 237-285.

Kajita, S. and Tani, K. (1991). Study of Dynamic Biped Locomotion on Rugged Terrain: Derivation and Application of the Linear Inverted Pendulum Mode. *Proc. 1991 IEEE Int. Conf. on Robotics and Automation*, California, pp 1405-1411.

Kajita, S. and Tani, K. (1995). Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode. *Proc. 1995 IEEE Int. Conf. on Robotics and Automation*, pp.2885-2891.

Kun, A. and Miller, W. T. Adaptive Dynamic Balance of a Biped Robot Using Neural Networks. *Proc. 1996 IEEE Int. Conf. On Robotics and Automation* 240-245.

Miller, W. T. (1994). Real-Time Neural Network Control of a Biped Walking Robot. *IEEE Control Systems*, 14(1):41-48.

Parks, P.C. and Sandberg, I.W. (1991). Universal Approximation Using Radial-Basis-Function Networks. *Neural Computation*, 246-257.

Poggio, T. and Girosi, F. (1990). Networks for Approximation and Learning. *Proc. IEEE*, 78:1481-1497.

Pratt, G. A. and Williamson, M. M. (1995). Series Elastic Actuators. *Proc. 1995 IEEE Int. Conf. On Intelligent Robots and Systems*, 399-406.

Pratt, J. (1995). *Virtual Model Control of a Biped Walking Robot*. Master Thesis, Department of Engineering in Electrical Engineering and Computer Science, MIT.

Pratt, J., Dilworth, P. and Pratt, G. (1997). Virtual Model Control of a Bipedal Walking Robot. *Proc. 1997 IEEE Int. Conf. on Robotics and Automation*, Albuquerque, New Mexico.

Raibert, M. H. (1986). *Legged Robots That Balance.* Cambridge, MA: MIT Press, 1986.

Raibert, M. H. and Wimberly, F. C. (1984). Tabular Control of Balance in a Dynamic Legged System. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(2):324:339.

Redfern, M. S. and Schumann, T. (1994). A Model of Foot Placement During Gait. *J. Biomechanics*, 27(11):1339-1346.

Sutton, R. S. (1996). Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. *Advances in Neural Information Systems*, 8: 1038-1044.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Townsend, M. (1985). Biped Gait Stabilization via Foot Placement. *J. Biomechanics*, 18(1):21-38.

Vukobratovich, M. Borovac, B., Surla, and Stokich, D. (1990). *Biped Locomotion: Dynamics, Stability, Control.* Springer-Verlag, 1990.

Watkins C.J.C.H. (1992). Q-Learning. *Machine Learning*, 8:279-292.