Learning Hybrid Position/ Force Control of a Quadruped Walking Machine Using a CMAC Neural Network

Yi Lin and Shin-Min Song*
Department of Mechanical Engineering
University of Illinois at Chicago
P.O. Box 4348
Chicago, IL 60680
e-mail: Smsong@uic.edu

Accepted February 5, 1997

Learning control algorithms based on the cerebellar model articulation controller (CMAC) have been successfully applied to control non-linear robotic systems in the past. Most of these previous works are focused on the position controls of manipulators. In this article, a CMAC-based learning control method for the hybrid force/position control of a quadruped walking machine on soft terrains is presented. The relationship between the foot force and the control variables is derived for various force control methods. By using the CMAC to approximate the dynamics of one leg, we are able to demonstrate the improved control accuracy without the exact leg model. The same concept is extended to the control of a quadruped walking machine. *© 1997 John Wiley & Sons, Inc.*

既に、Cerebellar Model Articulation Controller（CMAC：小脳モデル間接コントローラ）に基にした学習制御アルゴリズムで、非線型ロボットシステムを制御することは可能になっている。過去に行われたほとんどの研究では、マニピュレータの位置制御に注目していた。この発表では、軟弱な地形上を移動する四足歩行ロボットのハイブリッド力／位置制御を行うために使われる、学習制御法を基づいたCMACについて説明する。脚の力と制御変数の関係は、各種の力制御法に対して導出される。CMACを使うと一本の脚の力学を近似することが出来るので、正確な脚モデルがなくても制御精度の改善を確かめられる。そして、同じ概念を、四足歩行ロボットの制御にも拡張して適用する。

*To whom all correspondence should be addressed.

Journal of Robotic Systems 14(6), 483–499 (1997)
© 1997 by John Wiley & Sons, Inc.
CCC 0741-2223/97/060483-17

1. INTRODUCTION

The objective of walking machine control is to move the machine along a desired trajectory according to a given gait in a stable manner. Normally, the desired foot positions are calculated according to the gait equations and to the foot trajectory planning in the body coordinate system. The foot positions are then converted into the appropriate joint angles through inverse kinematics. The control computer compares the measured joint angles with the desired joint angles and reduces the difference through some types of feedback control. This approach is straightforward and easy to implement; however, on soft terrains or in any unknown environment, the walking machine may lose its stability due to the position planning errors and imbalance of foot forces. Hence, position control alone is not sufficient for practical walking control and foot force control is necessary.[9,10] For example, in the normal direction to the ground, foot forces have to be controlled to ensure firm foot support and even foot force distribution among all supporting legs; in the tangential direction, foot forces need to be monitored to avoid any slippage. Furthermore, foot force control allows better force distribution among the legs and thus reduces the chance of overloading the leg actuators.

To control forces one can use passive compliance such as the remote center compliance (RCC)[3] or use active compliance such as impedance control[7,8] and hybrid position/force control.[15] In the hybrid position/force control, any of the following force control methods may be used: stiffness control, damping control, combined stiffness and damping control, and explicit force control.[19]

In Ref. 6 a feedback force controller based on a simplified dynamic model around the operating point was used to control a six-legged walking machine on the sandy terrain. The force control is achieved through the modification of the planned position via an admittance matrix: $r(x, y, z) = r^* + \Lambda(F - F^*)$ where r is the modified foot position, r^* is the desired foot position, Λ is the admittance matrix, and $(F - F^*)$ is the force control error. The elements of the admittance matrix can be varied to achieve different active compliance, similar to the effect of a spring. However, the dynamic effects were neglected and the control system was essentially a kinematic controller, which only computes the commanded motion of the legs. Another drawback of the stiffness/admittance matrix method is that the system may become unstable if the linear force feedback gains are not selected properly.

In Ref. 17 a neural network was trained to compensate for the uncertainty of the dynamic model of multiple robots with redundancy. It effectively controlled the force/position of multiple cooperating robots through learning. The main drawback is the long training time of the learning algorithm. Another way to implement the learning control is by using the learning and generalization features of the cerebellar model articulation controller (CMAC). CMAC is a distributed table look-up method capable of approximating multi-variable and non-linear functions through learning.[1] For any input to the CMAC network, only a few neighboring memory locations are selected according to the mapping algorithm. The contents in those selected memory locations contribute to forming the output. CMAC also has the generalization property, i.e., it generates similar outputs for similar inputs. During the training process only the contents of the selected memory locations are updated according to certain training rules to reduce the difference of the CMAC output and the desired output. Due to its simple design and rapid learning capabilities CMAC is very suitable for real-time control implementations.

In the past some studies have applied CMAC to control serial and parallel manipulators.[5,14] These studies dealt mostly with position control of the manipulators. In this work, a hybrid position/force learning control architecture using CMAC is developed for walking control. We applied CMAC to approximate the inverse dynamics of one walking machine leg and used it in the feedforward control loop. In parallel, a feedback controller reduces the remaining control errors caused by the change of geometry, soil property, and other disturbances. The feedback controller also provides the initial training data for the CMAC learning. This algorithm is then extended to control all four legs of a quadruped walking machine.

2. MODELING AND FEEDBACK CONTROL

For this work, we assume that the terrain is soft and flat, and the foot force can be modeled as an elastic restoring force according to Hooke's law:

$$F = K_n \cdot D_a \quad D_a \geq 0 \tag{1}$$

where K_n is the ground stiffness and D_a is the ground deformation. We also assume that the leg stiffness, K_L, is much higher than the ground stiffness, K_n, such that the system stiffness is roughly equal to the ground stiffness.

It is common to use a linearized dynamic model as the basis for the controller design. Here, the leg motion is described by the motion of a point mass which is governed in free space by

$$\ddot{X} = \frac{U_x}{M}$$

$$\ddot{Y} = \frac{U_y}{M} \tag{2}$$

where $\ddot{X}$ and $\ddot{Y}$ represent the accelerations in the X and Y direction, M is the mass, and U_x and U_y are the applied forces on the point mass.

In the Y direction, after the point mass comes in contact with the ground, the reaction force on the point mass has to be included in the compliant system model. The superscripts f and c denote the transfer functions in the free space and in the compliant space:

$$^{f}G_y = \frac{Y}{U_y} = K_y \frac{1}{M} \frac{1}{S^2} = \frac{K_a}{S^2} \tag{3}$$

$$^{c}G_y = \frac{D_a}{U_y} = \frac{K_a}{S^2 + K_n K_a} \tag{4}$$

In Figure 1, J^T is the transpose of the Jacobian. In the case of the point mass it is a unit matrix.

The design objectives for the closed loop transfer function are: (a) the resonant magnitude M_p is less than 110%; and (b) the bandwidth (the break frequency) of the closed loop remains the same as the open loop. These objectives can be achieved through the proper placement of the dominant roots of the closed loop system.[2] The following LEAD controller is utilized for both free and compliant motion:

$$G_c(s) = K_c \frac{1 + \dfrac{S}{\omega}}{1 + \dfrac{S}{\alpha\omega}} \tag{5}$$

where ω and $\alpha\omega$ are the lower and upper corner frequencies of the LEAD controller. For digital implementation the LEAD controller is transformed from the Laplace space into the Z space using bi-linear transformation:

$$G_c(Z) = \frac{U(Z)}{E(Z)} = K_c \frac{1 + \dfrac{2T_0}{T_s}\dfrac{Z-1}{Z+1}}{1 + \dfrac{2T_1}{T_s}\dfrac{Z-1}{Z+1}} \tag{6}$$

where $E(Z)$ and $U(Z)$ represent the input and output of the LEAD compensator.

The difference equation of the LEAD controller is

$$U_k = \frac{E_k K_c(T_s + 2T_0) + E_{k-1} K_c(T_s - 2T_0) - U_{k-1}(T_s - 2T_1)}{T_s + 2T_1} \tag{7}$$

The sampling time, T_s, is determined by the natural frequency of the system, $\omega_n = \sqrt{K_a/m}$. The higher the natural frequency, the smaller the sampling time T_s.

3. COMPARISON OF FORCE CONTROL METHODS

In practice, there are more than two axes to be controlled, and they are not always perfectly aligned with the fixed ground coordinate frame. However, they can always be brought to the basic form of orthogonal axes with the concept of the selection matrix and proper coordinate transformations.[13] Without losing generality we assume that the force control occurs in the Y direction while the position control is in the X direction. In the following, the stiffness control, the damping control, the explicit force control, and the combined stiffness and damping control are briefly compared.

3.1. Stiffness Control

The goal of stiffness control is to achieve a defined relationship between the contact force and the displacement in the compliant space. The system with the stiffness control behaves like a spring. By adjusting the elasticity constant, P_{fn}, one can achieve different stiffness (or spring constant) for the overall system: the smaller the P_{fn}, the stiffer the controlled system, and vice versa. Therefore, the selection of P_{fn} should always follow the general rule to complement the stiffness of the environment, in order to achieve satisfactory control results (Fig. 2).

The actual foot force in the compliant space at steady state (for details see Ref. 12):

$$F_n = D_a K_n = \frac{K_a}{K_n + K_a + K_a K_n P_{fn}} F_r \tag{8}$$

where $F_r = K_n D_r$ is the reference foot force.

The reference displacement can be calculated such that the steady state foot force, F_n, equals the desired foot force, F_d

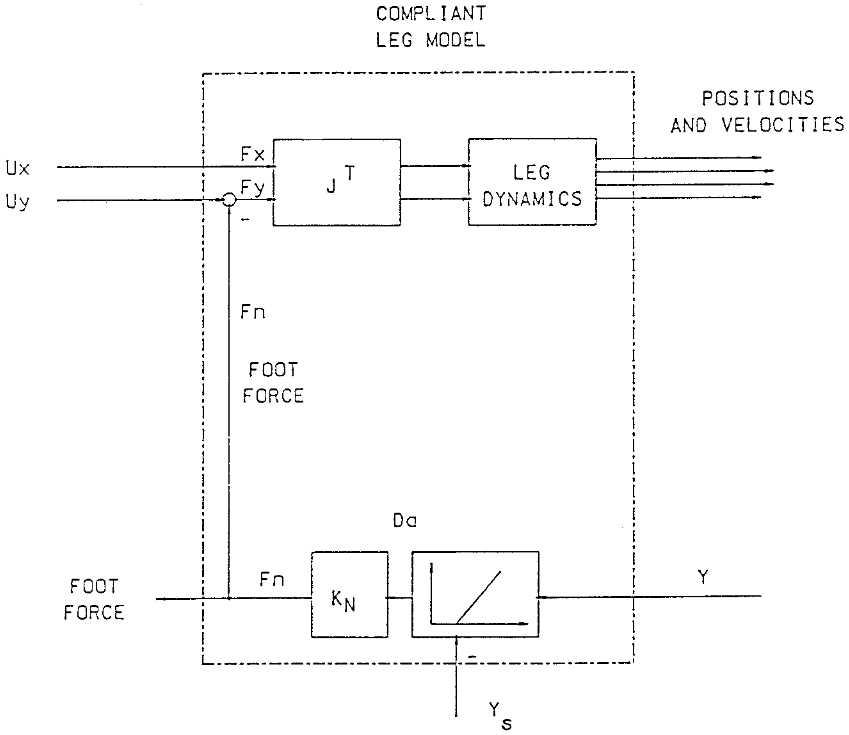

Figure 1. Compliant dynamic model of a walking machine leg.

$$D_r = \frac{F_r}{K_n} = \left(\frac{1}{K_n} + \frac{1}{K_a} + P_{fn}\right) F_d \qquad (9)$$

3.2. Damping Control

The objective of damping control is to achieve a defined relationship between the contact force and the velocity. The system with damping control behaves like a mass-dash pot system. The reaction force is proportional to the velocity with which the body moves in the compliant space in the Y direction (Fig. 3). The contact force is

$$F_n = \frac{K_a}{1 + K_a V_{fn}} \dot{Y}_r \qquad (10)$$

The proper reference velocity, $\dot{Y}_r$, to achieve desired foot force is

$$\begin{aligned} \dot{Y}_r &= \frac{1 + K_a V_{fn}}{K_a} F_d \\ &= \left(V_{fn} + \frac{1}{K_a}\right) F_d \end{aligned} \qquad (11)$$

3.3. Combination of Stiffness and Damping Control

In this case, the steady state contact force depends on both the reference position and velocity. The controlled system exhibits a behavior that is similar to the mechanical mass–spring–damper system. The reaction force, F_n, is transformed into position and velocity via an elasticity constant, P_{fn}, and an admittance constant, V_{fn}, respectively. The elasticity and admittance parameters can be adjusted to achieve desired stiffness and damping. From the control structure in Figure 4 the foot force at the steady state can be derived (for details see Ref. 12) and given as

$$F_n = \frac{(K_{ap} D_r + K_{av} \dot{Y}_r) K_n}{K_n + K_{av} V_{fn} K_n + K_{ap} + K_{ap} P_{fn} K_n} \qquad (12)$$

We set $\dot{Y}_r = 0$ and use eq. (12) to determine the reference displacement, D_r, such that the contact force, F_n, is equal to the desired foot force, F_d

$$D_r = \left(\frac{1}{K_n} + \frac{1}{K_{ap}} + P_{fn} + \frac{K_{av}}{K_{ap}} V_{fn}\right) F_d \qquad (13)$$

3.4. Explicit Force Control

The explicit force control is the direct form of the force control. The actual contact force is compared

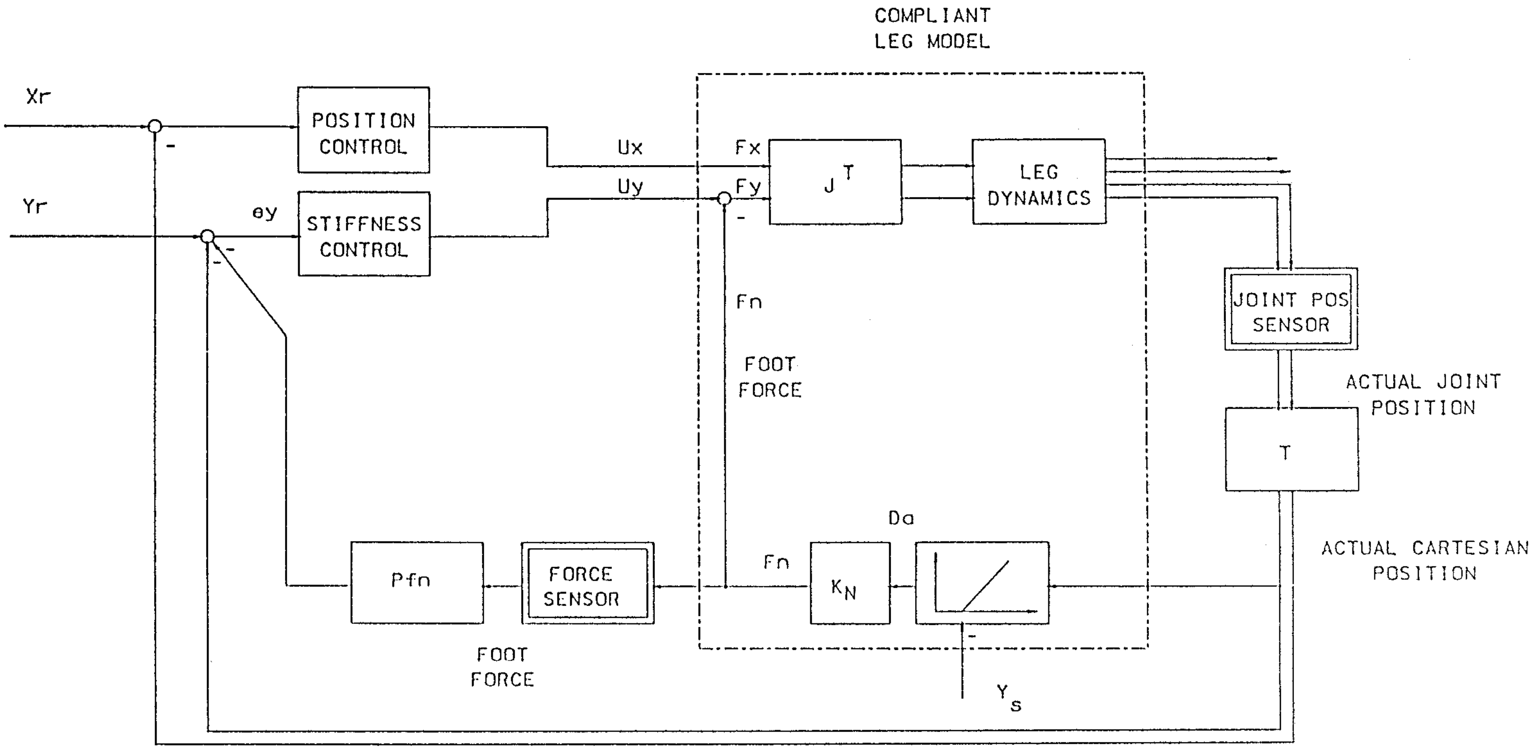

Figure 2. Structure of stiffness control.

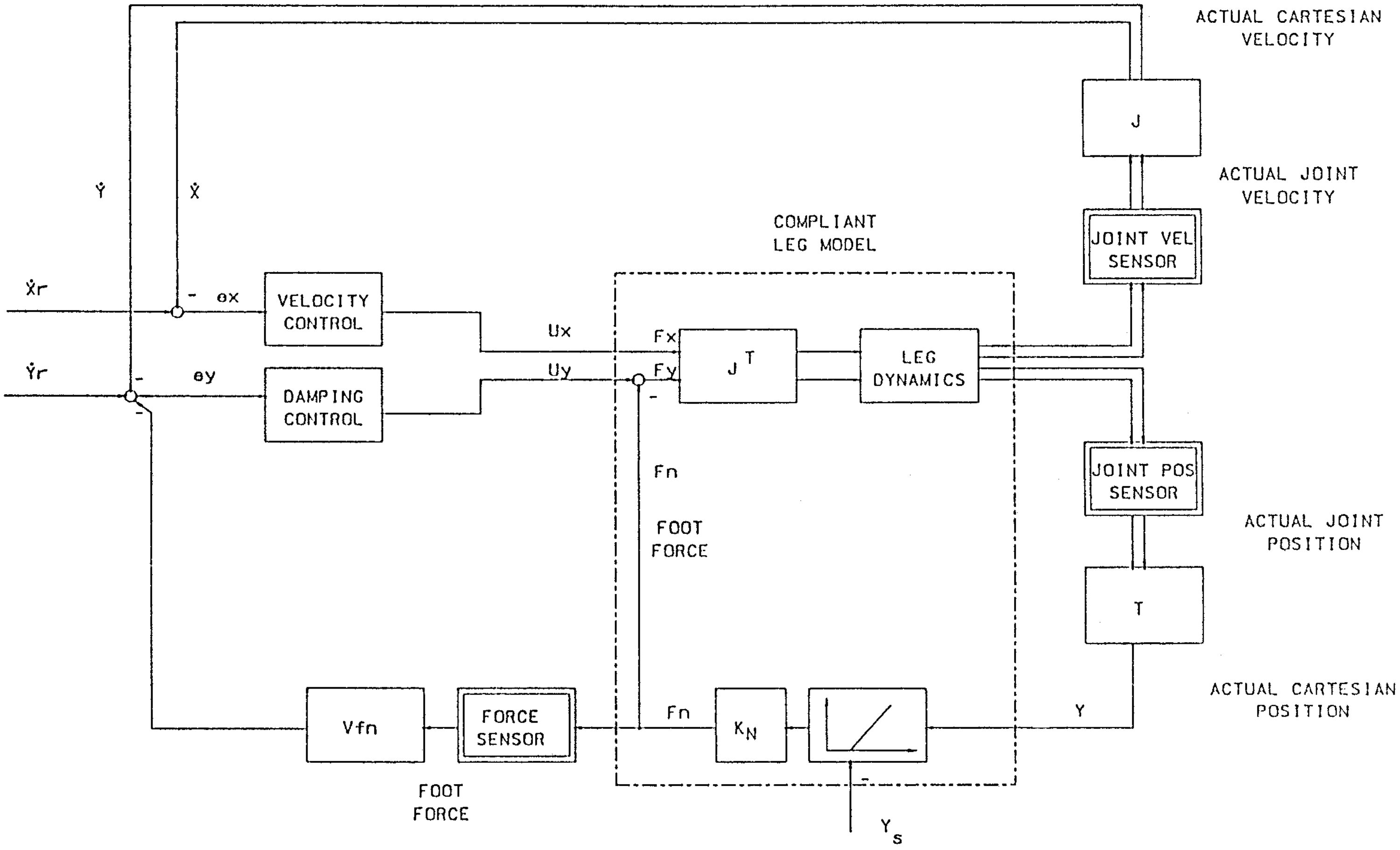

Figure 3. Structure of damping control.

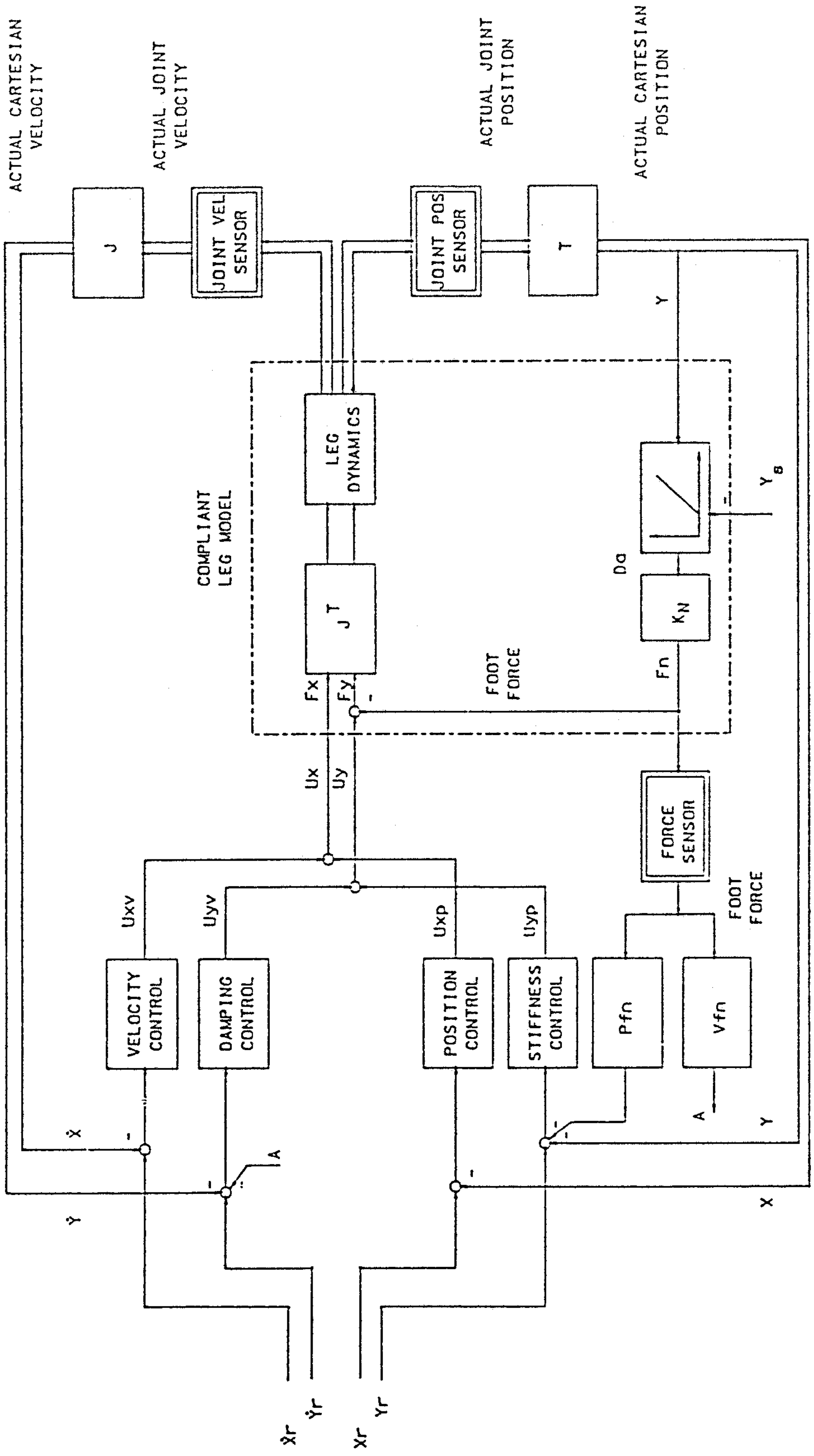

Figure 4. Structure of combined stiffness/damping control.

with the desired foot force directly and a feedback controller minimizes the difference between these two values. The control structure is shown in Figure 5.

The foot contact force at the steady state is

$$F_n = \frac{K_a}{1 + K_a} F_r \tag{14}$$

To achieve the desired foot force, F_d, in the steady state, the reference foot force, F_r, is set to the following value

$$F_r = \frac{1 + K_a}{K_a} F_d = \left(1 + \frac{1}{K_a}\right) F_d \tag{15}$$

According to the computer simulation[12] the combined stiffness and damping control appears to be most suitable for the leg control and will be used as the force controller in the following sections.

4. CMAC-BASED LEARNING CONTROL OF POSITION AND FORCE

The dynamic model of a walking machine leg in the free space can be derived using Lagrange's method and is expressed as

$$M_i(\theta_i)\ddot{\theta}_i + C_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + G_i(\theta_i) = \tau_i \tag{16}$$

where

$M_i \in R^{n\times n}$ is the positive definite, symmetrical inertia matrix of the ith leg.

$C_i\dot{\theta}_i \in R^{n\times 1}$ is the Coriolis and centrifugal term.

$G_i \in R^{n\times 1}$ is the gravitational term.

$\tau_i \in R^{n\times 1}$ is the vector of applied torques from motors.

$\theta_i, \dot{\theta}_i, \ddot{\theta}_i \in R^{n\times 1}$ are vectors of joint position, velocity and acceleration, respectively.

n is the number of joints in each leg.

The dynamic model of a walking machine leg in the compliant space can be expressed as

$$M_i(\theta_i)\ddot{\theta}_i + C_i(\theta_i, \dot{\theta}_i)\dot{\theta}_i + G_i(\theta_i) = \tau_i - {}^{f}_{\theta}J_i^T\, {}^{f}F_{fi} \tag{17}$$

where

${}^{f}_{\theta}J_i^T \in R^{n\times 3}$ is the transpose of the Jacobian matrix for the ith leg.

${}^{f}F_{fi} \in R^{6\times 1}$ is the foot contact force of the ith leg.

The CMAC-based learning control law consists of the learning law and the control law:

$$\tau(t) = \hat{D}_d(t) + U_{force}(t) \tag{18}$$

where

$\hat{D}_d(t) = \sum_{i=1}^{C} W_i$ is the CMAC approximation of the inverse dynamics in eqs. (16) and (17).

W_i: contents in the ith CMAC memory location, also called weights.

C: quantity of the selected memory locations.

$U_{force}(t)$: output of the combined stiffness and damping control algorithm.

We propose the following learning law to update the weights in the C CMAC memory locations using the current system states and the output of the feedback controller. The feedback output provides the initial training signal to the CMAC. As CMAC progresses in learning, the share of feedback decreases in the overall control output. Ideally, CMAC approximates the inverse dynamics with 100% accuracy, then $U_{force}(t)$ will become zero. Thus, the rate of change of the weights is the feedback control output, U_{force}, multiplied by the learning rate, β, and divided by the generalization size, C, and by the time period, Δt, between two successive updates:

$$\frac{d}{dt} W_i(t) = \beta \frac{U_{force}(t)}{C \cdot \Delta t} \tag{19}$$

Hence, the new, updated CMAC output after each learning step is

$$\begin{aligned} \hat{D}_d(t) &= \sum_{i=1}^{C} \left[W_i(t - \Delta t) + \Delta t \cdot \frac{d}{dt} W_i(t - \Delta t) \right] \\ &= \sum_{i=1}^{C} \left[W_i(t - \Delta t) + \Delta t \cdot \beta \frac{U_{force}(t - \Delta t)}{C \cdot \Delta t} \right] \\ &= \sum_{i=1}^{C} \left[W_i(t - \Delta t) + \beta \frac{U_{force}(t - \Delta t)}{C} \right] \end{aligned} \tag{20}$$

where $W_i(t - \Delta t)$ is the weight at the ith location before the update.

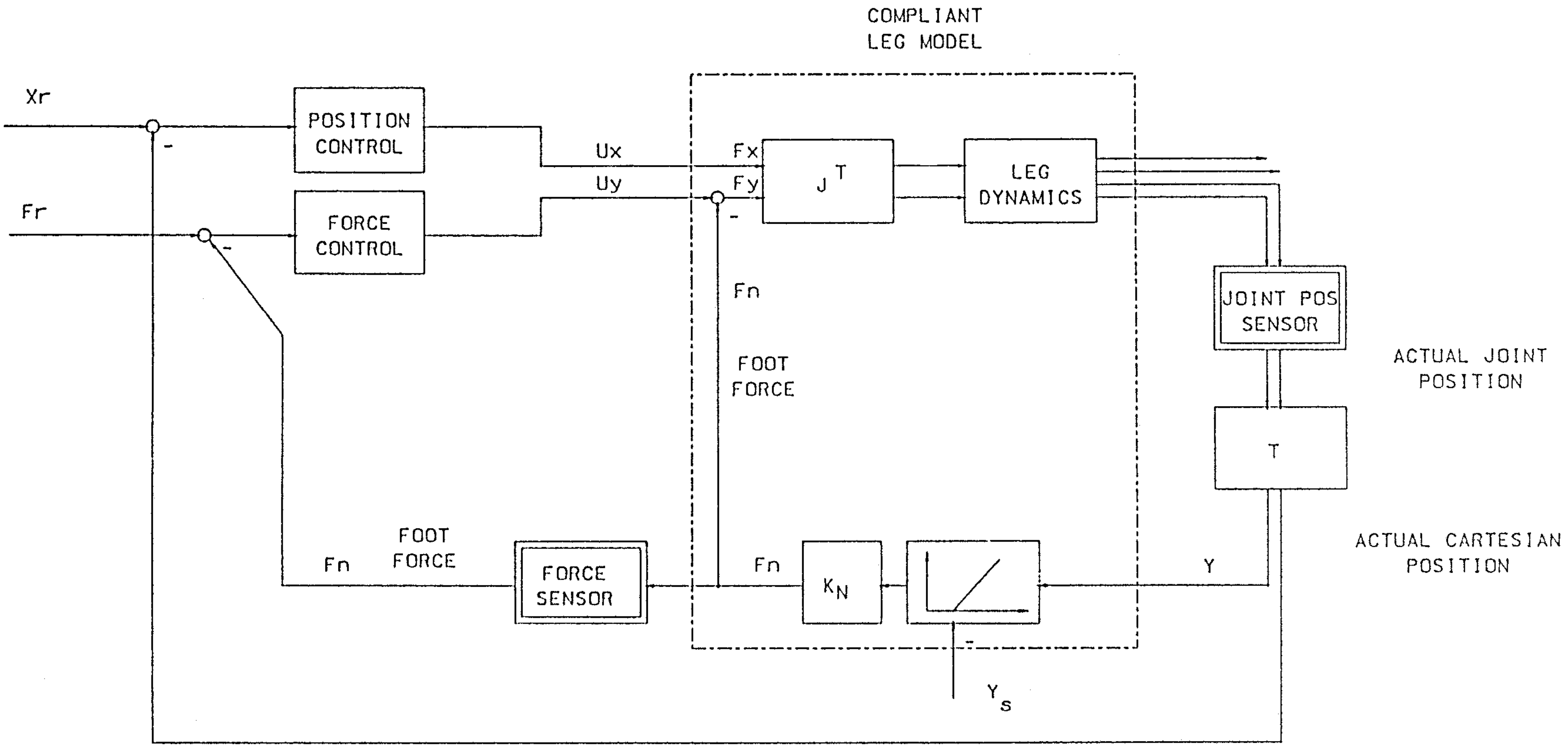

Figure 5. Structure of explicit force control.

The complete CMAC-based learning control law is the combination of eq. (18) and eq. (20):

$$\tau_k = \sum_{i=1}^{C} \left[W_{i,(k-1)} + \beta \frac{U_{force,(k-1)}}{C} \right] + U_{force,k} \tag{21}$$

$$\begin{aligned} U_{force,k} &= \frac{E_k K_c(T + 2T_0) + E_{k-1} K_c(T - 2T_0) - U_{k-1}(T - 2T_1)}{T + 2T_1} \\ &\quad + \frac{E'_k K_c(T + 2T_0) + E'_{k-1} K_c(T - 2T_0) - U_{k-1}(T - 2T_1)}{T + 2T_1} \end{aligned} \tag{22}$$

where the stiffness control errors are $E_k = Y_{r,k} - Y_k - F_{n,k}P_{fn}$, ·and the damping control errors are $E'_k = Y_{r,k} - Y_k - F_{n,k}V_{fn}$.

It was proven analytically that the above learning law is convergent; i.e., that the CMAC approximates the inverse dynamic model of the leg with certain accuracy. Furthermore, the learning control system has complete stability.[12]

In this work, we assumed that the CMAC has an unlimited number of memory locations. Thus, the consequence of memory reduction algorithms (hashing) and the resultant data collision will not be dealt with in the scope of this work. In practice, however, the hash coding is often used to keep the physical memory small and manageable. By choosing proper generalization size C and proper size of physical memory one can keep the rate of collision below a noticeable level. So far, there are few rigorous analyses on the impact of hashing on CMAC learning convergence,[4] although our simulation and the simulation of others[18] have shown that CMAC learning is robust to collisions.

5. CMAC-BASED LEARNING CONTROL OF ONE LEG

In this section, we simulated the CMAC-based learning control of force and position of a walking machine leg. The control structure is shown in Figure 6. A walking machine leg usually has three degrees of freedom in space. Without losing the nonlinear nature of the system we consider a planar foot motion generated by two parallel revolute joints, and the motion is frictionless. The non-linear inverse dynamics of the two-dimensional (2D) leg is

$$\tau(t) = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{23}$$

where

$$\tau(t) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \end{bmatrix}, \ddot{\theta} = \begin{bmatrix} \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix}$$

$$M(\theta) = \begin{bmatrix} l_2^2 m_2 + 2l_1 l_2 m_2 \cos\theta_2 + l_1^2(m_1 + m_2) & l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2 \\ l_2^2 m_2 + l_1 l_2 m_2 \cos\theta_2 & l_2^2 m_2 \end{bmatrix}$$

$$C(\theta, \dot{\theta})\dot{\theta} = \begin{bmatrix} -m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1 \end{bmatrix}$$

$$G(\theta) = \begin{bmatrix} -(m_2 l_2 \cos(\theta_1 + \theta_2) + m_1 l_1 \cos\theta_1 + m_2 l_1 \cos\theta_1)g \\ -m_2 l_2 \cos(\theta_1 + \theta_2)g \end{bmatrix}$$

For the numerical example, the masses are $m_1 = m_2 = 0.5$ kg and the link lengths are $l_1 = l_2 = 0.35$ m. The integration of the dynamic equation of motion is performed by the fourth order Runge–Kutta algorithm with a fixed step size of 0.005 s. The walking cycle time is $T = 3$ s. Thus, each cycle has 600 data points. The CMAC learning takes place on a dedicated CMAC computer board with 1 megabyte of onboard memory. The number of quantization functions is $C = 32$. Each quantization function has 33 resolution elements. The learning rate is $\beta = 50/256$. The feedback control is implemented through a LEAD compensator in all simulations. For all simulation tests the starting conditions are: sampling time = 0.005 s, the loop gains are $K_{ap} = 100$, and $K_{av} = 5$. The elasticity and admittance constants, P_{fn} and V_{fn}, are set to 0.1. The overall system stiffness is 10 N/m. The desired foot force is 1 N.

The leg is controlled to track the trajectory of a fifth order and a third order polynomial, which describe the foot movement in the horizontal and vertical directions during the transfer phase. The coefficients of the polynomials are determined through the boundary conditions at the take-off and touch-down points.[11] The foot trajectory on the ground is a straight line with a constant velocity. The ground stiffness is assumed to be known and the leg mechanism is treated as rigid.

In Figure 7 the desired foot trajectory for the transfer phase and the reference foot positions for the support phase are displayed. The reference foot position, D_r, is not the same as the desired foot position, because D_r is calculated according to the eq. (13) such that the desired foot force can be achieved. In

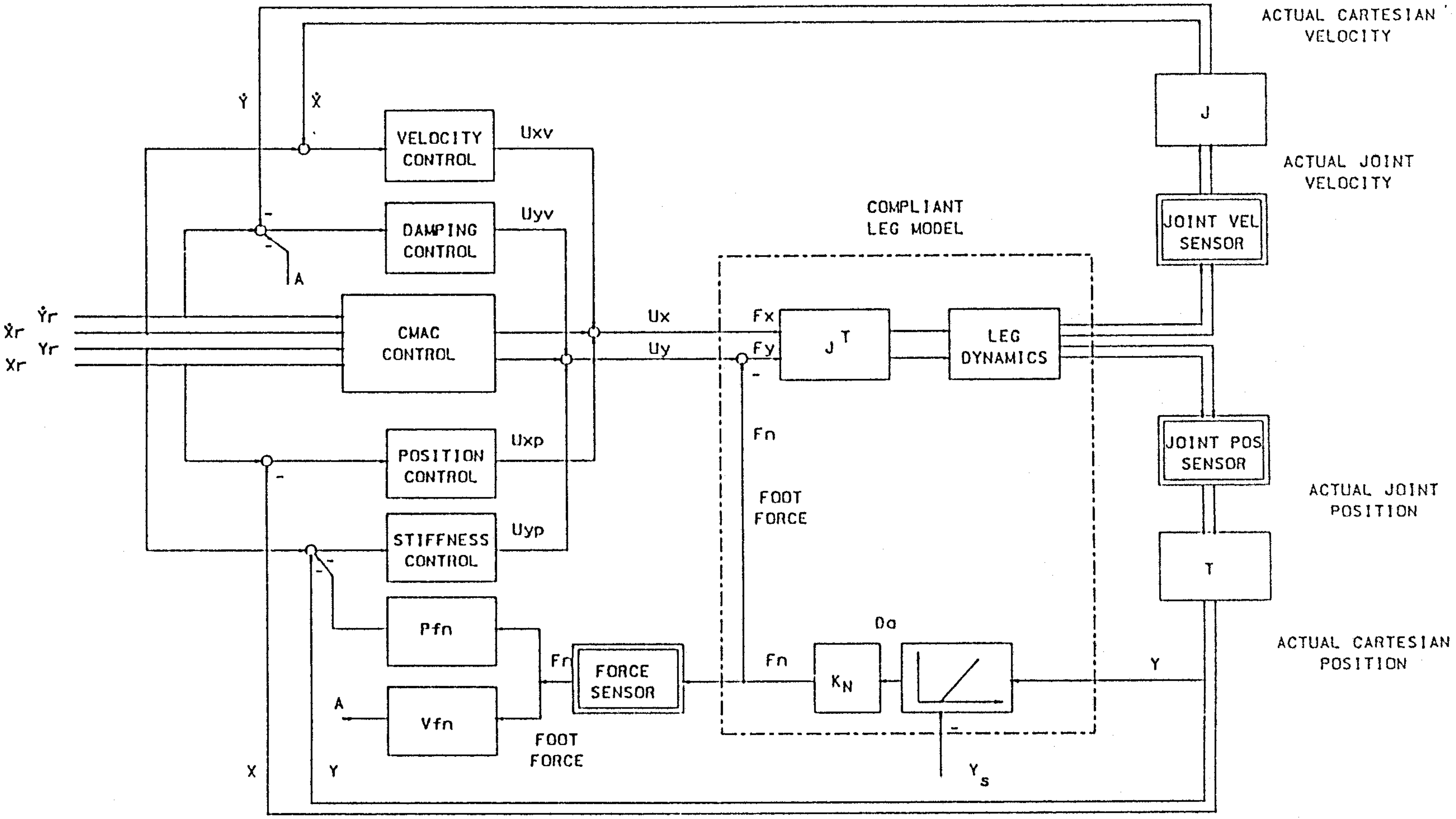

Figure 6. Structure of CMAC-based learning control of position and force.

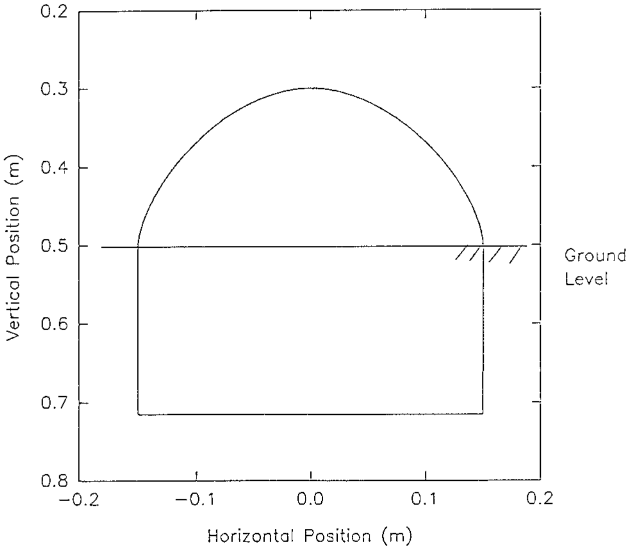

Figure 7. Desired foot trajectory during the transfer phase and reference position during the support phase.

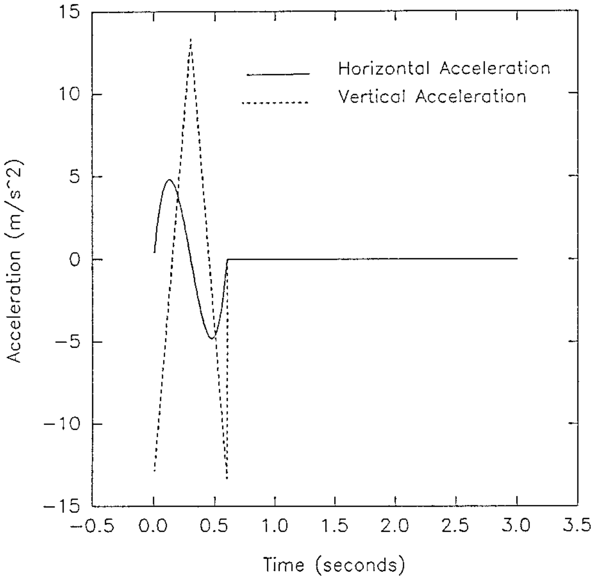

Figure 9. Desired accelerations of the foot.

Figures 8 and 9 the desired velocities and accelerations of the foot in the horizontal and vertical directions are shown. The ground is at the 0.5-m level.

In the following diagrams, the first 600 data points are taken from $t = 0$ s to $t = 3.0$ s (first cycle); the data points from 600 to 1200 are taken from $t = 4$ s to $t = 6$ s (second cycle). The data points that range from 1200 to 1800 correspond to the time period $t = 12$ s to $t = 15$ s (fifth cycle). The desired force or position trajectory are drawn with solid lines while the actual trajectories are drawn with dashed lines. In the first cycle, only the PD controller is used. From cycle 2 to cycle 5 (data points 600 to 1800) CMAC is applied in addition to the PD control.

For feedback control without learning during the first cycle, the foot tries to follow the desired foot position during the foot transfer phase in free space (Fig. 10). It is clear that there are fairly large position control errors in both the X and Y directions. With

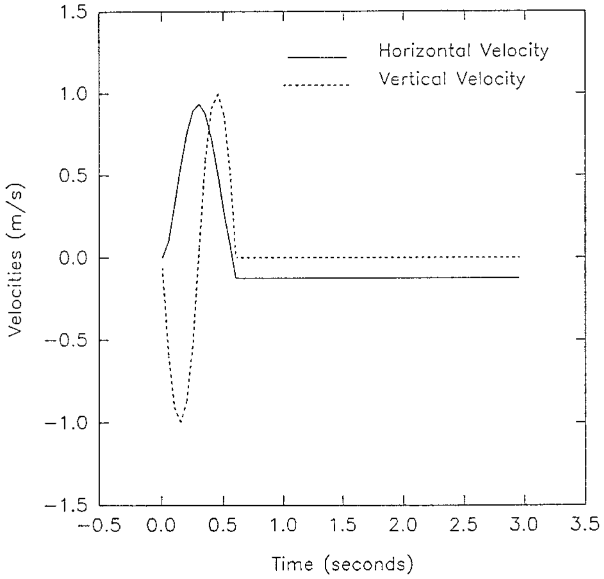

Figure 8. Desired velocities of the foot.

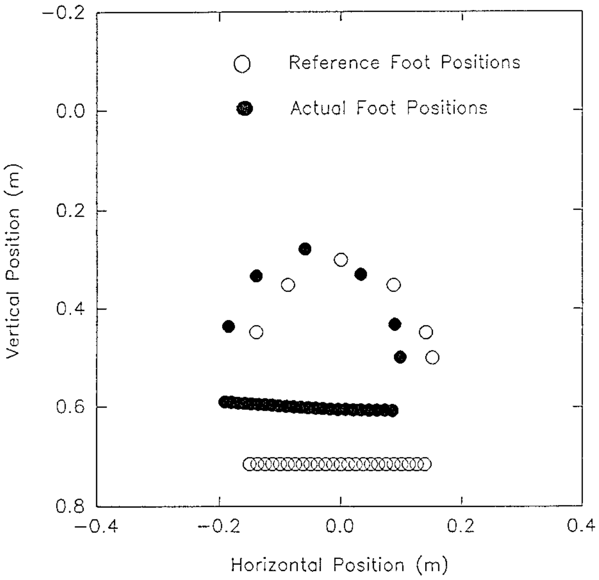

Figure 10. Foot position tracking, PD control only without CMAC.

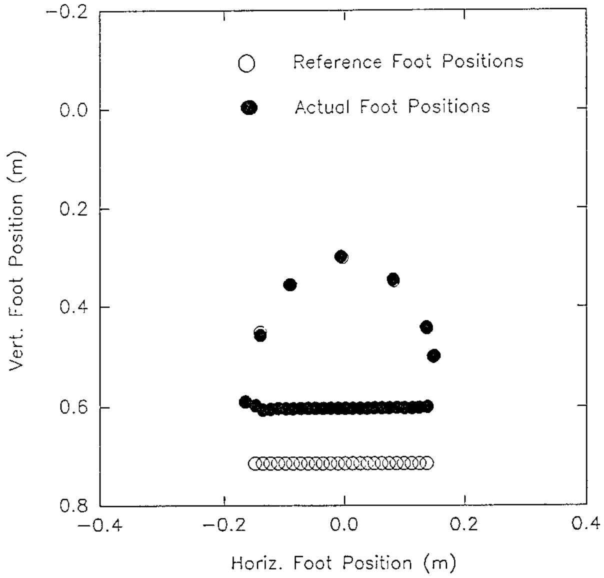

Figure 11. Foot position tracking with CMAC.

CMAC control (after 12 s of learning), the position errors during the transfer phase are virtually eliminated (see Fig. 11), without increasing the PD control gains.

In Figure 12, during the first 3 s (0 to 600 data points), only PD control is applied. The foot force goes to 1.1 N, then decreases to approximately 0.9 N. Between $t = 3$ s and $t = 6$ s CMAC is applied and some improvement can be observed. Steady state error is reduced to less than 5% and the foot force remains relatively steady during the support phase. Data points 1200 to 1800 (from $t = 12$ s to $t = 15$ s) show an interesting situation, in which more learning does not bring better performance. In fact, the foot forces show more oscillations and steady state error has increased. The reason for this behavior can be found in Figure 13.

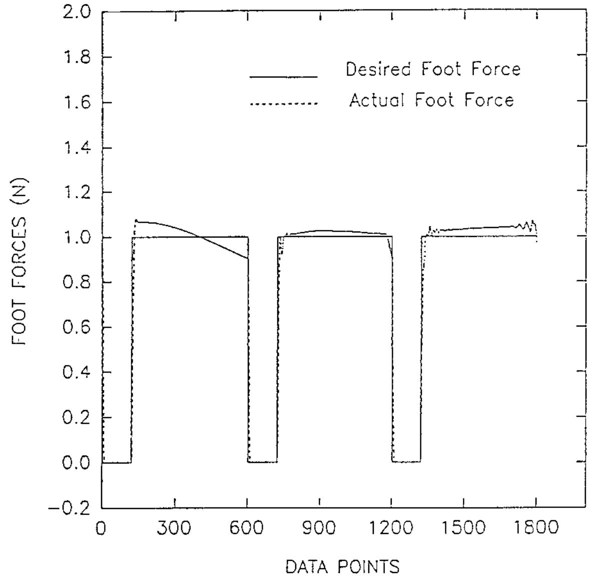

Figure 12. Foot forces control with PD controller only (from 0 to 600 data points), and with CMAC (from 600 to 1800 data points).

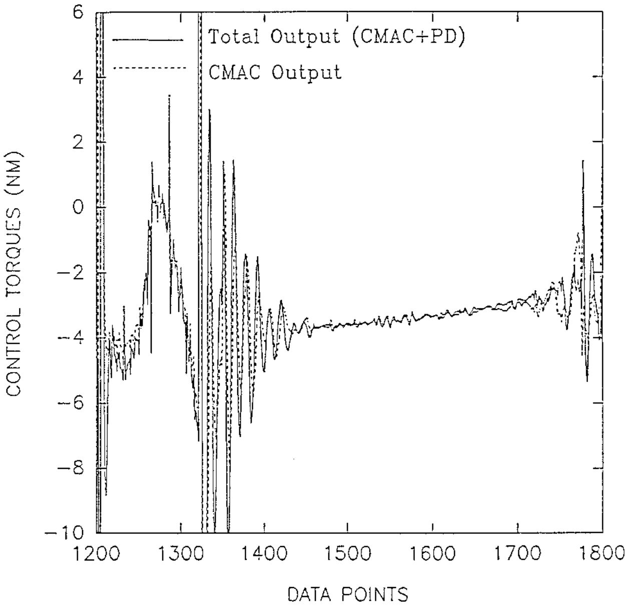

Figure 13. The control torques for joint 1 in detail.

In Figure 13, it can be seen that the learning control torques oscillate around those places where the torque function has abrupt changes; for example, around data point 1320, where the foot force jumps from 0 to 1 N. When the control torque function is smooth, such as during the support phase, then the CMAC output is also relatively smooth. This shows that a pre-defined CMAC is more suitable to approximate smooth functions. Of course, the unit step function for the foot force in this simulation is an extreme case. In practice, the foot force is a ramped function rather than a step function. Hence, the change of torques is expected to be less drastic as shown in this example. CMAC probably provides better approximation in a real application. Figure 13 also showed that CMAC provides the major part of the overall control torques.

In summary, CMAC is able to improve the control performance in both the force control and position control, in the free space and in the compliant space. Tracking errors are minimized due to the CMAC learning capabilities. It also shows the limitations of

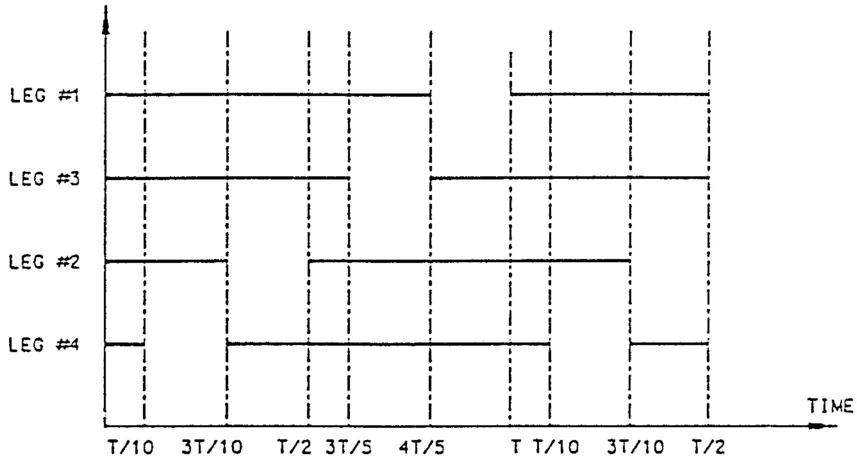

Figure 14. Gait diagram of a wave gait.

the CMAC. A pre-defined CMAC may not be suitable to learn rapidly changing functions with peaks and valleys.

6. QUADRUPED WALKING CONTROL USING CMAC

In this section, CMAC-based hybrid force and position control is applied to the control of a quadruped walking machine in statically stable walking, meaning at least three feet are on the ground at any given time. A posture control algorithm is added to obtain the desired posture of the walking machine. The ground depression is also considered in the algorithm. Finally, an animation program demonstrates the effectiveness of the control algorithm.

6.1. Basis for Walking Control

To illustrate the essentials of learning control of the walking machine the following factors are considered.

The quadruped consists of a main body structure and four identical cylindrical legs. The animation program allows change of geometry of each of the individual legs. Each leg has 2 degrees of freedom (DOF) with two rotational joints. Both links are 0.35 m in length and weigh 0.5 kg. The main body structure connects the four legs at the hip joints. The structure is 0.45 m long, 0.45 m wide, and weighs 1 kg. For walking control we chose a simple and effective gait for the level terrain, the wave gait.[16] With the wave gait the leg movements on the right side are identical to those on the left side, except they are a half-cycle out of phase. Figure 14 shows the gait diagram of a wave gait with a duty factor of 0.8. The duty factor is the time fraction of a cycle which the leg is on the ground. The black lines indicate the support phase of the leg on the ground, whereas the empty space indicates the transfer phase of the leg in the air. One can see that the foot lifting sequence propagates from the hind legs to the front legs like a wave. A diagram of a walking machine is given in Figure 15.

Due to the higher speed with which the legs are moved compared with body movement, the dynamics effect of the legs are significant.[11] In fact, the larger the duty factor, the higher is the effect of the leg dynamics on walking performance. Because this work is focused on the statically stable walking gaits with large duty factors, the dynamics of the leg links are the primary object of the control. Body mass is taken into consideration only for static force calculations.

The animation of the walking consists of display of many discrete instances. For each instance, the joint angles of each leg are calculated using the equation of motion. The final position and orientation of the body is then displayed using the forward kinematics and the adjustment due to the ground depression. The details of the animation program and posture control can be found in Ref. 12.

6.2. Force Control in the Walking Machine

The force control for the quadruped is related to the force and position control of one leg. However, there is a significant difference between them: a quadruped has a moving coordinate system with its origin at the body mass center, whereas the position / force control in the one-leg case has a fixed coordinate system. In

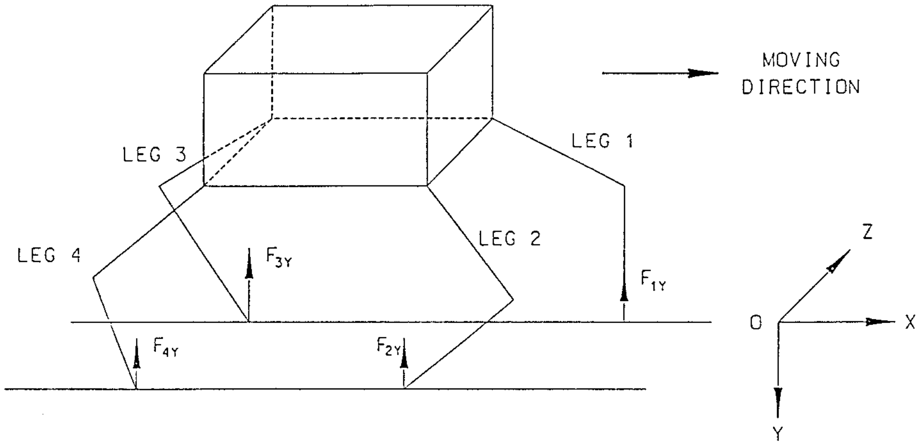

Figure 15. Diagram of a qudruped.

the one-leg case, the control variable is the foot contact force. One could set any amount within the capacity limit of the actuators as the desired contact force and achieve that goal. In contrast, the foot contact forces in a walking vehicle cannot always be set and changed by certain actuators as part of a force control scheme. Rather, the foot contact forces are dependent on the posture of the machine. The foot contact forces can be manipulated through change of body posture.

Based on the planned body trajectory the desired foot contact force on the ith leg, F_{di}, can be calculated. This desired force is translated into a reference position, D_{i_ref}, through the stiffness/damping relationship:

$$D_{i\_ref} = \left(\frac{1}{K_n} + \frac{1}{K_{ap}} + P_{fn} + \frac{K_{av}}{K_{ap}} V_{fn}\right) F_{di} \qquad (24)$$

Four CMACs are initialized for four legs and each CMAC has 256 kilobytes of physical memory. For simulation, the number of the quantization functions, C, is 32. Each quantization function has 33 resolution elements. The learning constant is 50/256. Other control constants are: $K_{av} = 5$, $K_{ap} = 50$, $P_{fn} = 0.0001\ m/N$, $V_{fn} = 0.0001$ m/N · s, and $K_n = 10{,}000\ N$/m. The posture control gain is set to 8.

6.3. Simulation Results

Throughout the simulation the sample time is 0.005 s and the body weight is 40 N. Figure 16 shows the initial posture of the walking machine. The left front, right front, left rear, and right rear legs are leg numbers 1, 2, 3, and 4, respectively.

In Figure 17, foot 3 is lifted. The center of gravity (COG) remains in the foot support pattern; however, the distance between the projection of the COG and the rear support boundary is very short. The distance is called the rear stability margin.[16] Another indicator for stability in this case could be the foot force, F_{n2}. When $F_{n2} = 0$, the system is likely to fail.

Figure 18 shows the instance immediately after foot 3 touches down and before foot 1 is lifted. According to the wave gait diagram, the lift-off of leg 1 and the touch-down of foot 3 should occur at the same time. Even though $F_{d1} = 0$, $F_{n1} \neq 0$ due to the delay in the response of leg 1. However, his delay has no adverse effect on the walking stability.

Figure 19 shows another critical moment for stability. Foot 2 is still in the transfer phase while the COG is close to the front support boundary. Even if the COG crosses over the front support boundary, the system will be likely to fall forward. However,

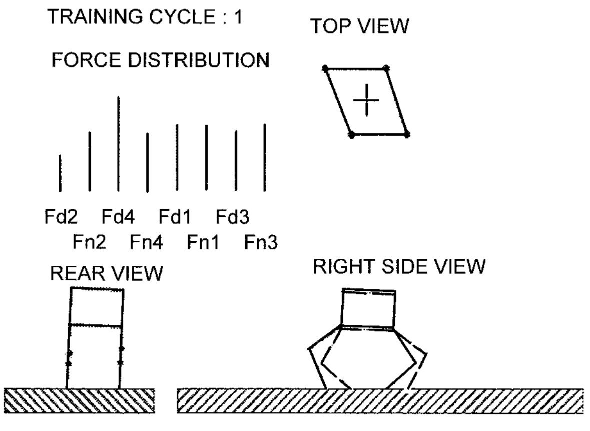

Figure 16. Front, side, and top views of the quadruped.

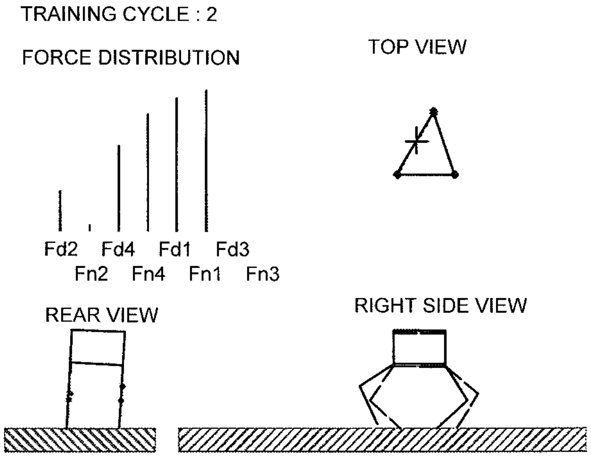

Figure 17. Walking with force/position control, posture control, and CMAC: lift-off of foot 3.

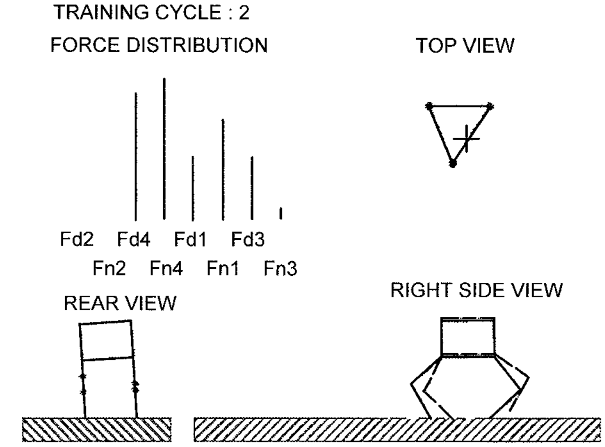

Figure 19. Walking with force/position control, posture control, and CMAC: transfer of foot 2.

the stability will be immediately restored by leg 2 which is in the process of gaining a firm foot support (Fig. 20). For the above example, the combination of force/position feedback, posture control, and CMAC provided stable and steady walking control. This is achieved in the absence of a detailed model of the system and with a diminishing contribution from the feedback control. The advantages are shorter response time in the feedforward control, and improved robustness toward the sensor noises because the input of CMAC comes directly from trajectory planning, not from measurement. However, the operation of the CMAC is still influenced in part by the sensor noises in the feedback loop, in fact, the unfiltered noises will get into the CMAC memory table through the learning algorithm. The exact effect of the sensor noises or sensor failure in the CMAC operation needs to be investigated further. Nevertheless, this work presents a potentially simple and yet effective way to handle a complex system (such as a walking machine) with little hardware and software, thus freeing up the operator from the low level motion control tasks.

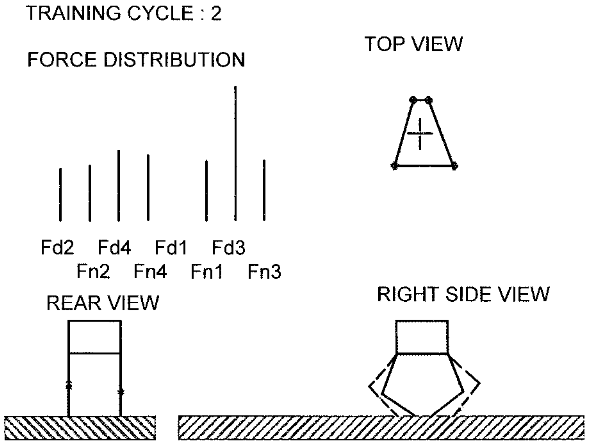

Figure 18. Walking with force/position control, posture control, and CMAC: touch-down of foot 3.

7. SUMMARY

The dynamic model of a walking machine leg changes its form when it goes from the transfer phase to the support phase. In addition, the complete model of a walking machine leg is also characterized by nonlinearities such as the Coriolis and centrifugal terms. In this work, a feedback controller was first designed based on a linearized model. Then, various force con-

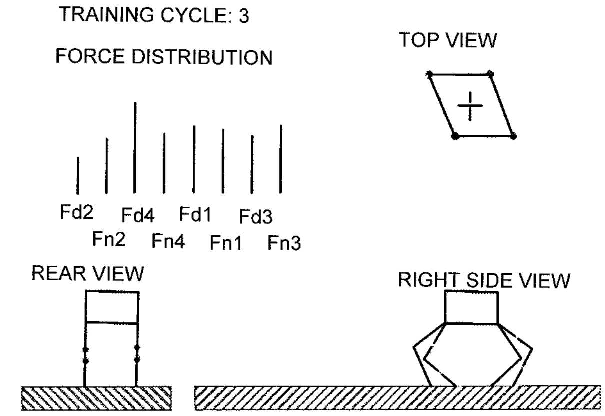

Figure 20. Walking with force/position control, posture control, and CMAC: touch-down of foot 2.

trol methods were evaluated. The relationships between the contact force, depression, control parameters, reference values, and ground properties were derived for the steady state. Among the methods evaluated, combined stiffness and damping control was shown to provide the best dynamic behavior for leg control.

Due to the lack of an exact dynamics model, the control performance of the feedback control was limited. To improve performance, a CMAC-based learning control algorithm was introduced. The output of the feedback controller was used to provide the initial training data for the CMAC. The trained controller could reduce the control errors in the position and force control. The limitations of the CMAC were also discussed.

The developed force and position learning control was also extended to the control of a quadruped. The walking performance was visualized by a software package, which proved to be helpful tool for monitoring control performance during walking. The vertical movement of the body and the rolling motion of the machine were shown from the front view. The information on the leg movement and the ground depression could be obtained from the side view. The pitching motion of the body and the overall posture of the quadruped was also shown. Another useful feature is the foot force display, which provided an overview of the actual foot force distribution.

The learning control methods developed in this work could be applied to manufacturing processes such as cutting, grinding, deburring, and part mating where force and position controls are required. Robots equipped with force control capabilities can perform peg-in-hole, door opening, screw driving, glass wiping, and other compliant tasks with more flexibility than with the elements of passive compliance.

The first author thanks UIC, Inc. for the continuous encouragement and support during the completion of this work.

REFERENCES

1. J. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Trans. ASME J. Dynam. Syst. Meas. Control.,* **97,** 220–227, 1975.
2. R. C. Dorf, *Modern Control Systems,* Addison-Wesley, Reading, MA, 1986.
3. S. H. Drake, *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly,* Ph.D Dissertation, Department of Mechanical Engineering, MIT, Cambridge, MA, 1977.
4. D. Ellison, "On the convergence of the multidimensional Albus perceptron," *Int. J. Robot. Res.,* **10,** 338–357, 1991.
5. Z. Geng and L. Haynes, "Neural network solution for the forward kinematics problem of a steward platform." *Proc. IEEE Int. Conf. Robot. Automat.,* 1991, pp. 2650–2655.
6. D. M. Gorinevsky and A. Y. Shneider, "Force control in locomotion of legged vehicles on rigid and soft surfaces," *Int. J. Robot. Res.,* **9,** 1990.
7. N. Hogan, "Impedance control—an approach to manipulation," Parts I, II, III. *ASME J. Dynam. Sys. Meas. Control,* **107,** 1–24, 1985.
8. H. Kazerooni, T. Sheridan, and P. Haoupt, "Robust compliant motion for manipulators. Part I: The fundamental concepts of compliant motion," *IEEE J. Robot. Automat.,* **2,** 83–91, 1986.
9. C. A. Klein, K. Olson, and D. Pugh, "Use of force and attitude sensors for locomotion of a legged vehicle over irregular terrain," *Int. J. Robot. Res.,* **2,** 3–17, 1983.
10. C. A. Klein and R. L. Briggs, "Use of active compliance in the control of legged vehicles," *IEEE Trans. Syst. Man Cybernet.* **10,** 393–400, 1980.
11. B. S. Lin, *Dynamics of a Quadruped in Walking, Running and Gait Transition,* Ph.D. thesis, Department of Mechanical Engineering, University of Illinois at Chicago, Chicago, IL, 1993.
12. Y. Lin, *Learning Control of a Quadruped Walking Machine Using the Cerebellar Model Articulation Controller Neural Network,* Ph.D. Thesis, Department of Mechanical Engineering, University of Illinois at Chicago, Chicago, IL, 1995.
13. M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. Syst. Man Cybernet.,* **11,** 418–432, 1981.
14. W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft, "Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller," *IEEE Trans. Robot. Automat.,* **6,** 1–9, 1990.
15. M. Raibert and J. Graig, "Hybrid position / force control of manipulators," *ASME J. of Dynam. Syst. Meas. Control,* **102,** 126–133, 1981.
16. S. M. Song and K. J. Waldron, *Machines that Walk: The Adaptive Suspension Vehicle,* MIT Press, Cambridge, MA, 1989.
17. J. M. Tao and J. Y. S. Luh, "Application of neural network with real-time training to robust position / force control of multiple robots," *IEEE Int. Conf. Robot. Automat.,* Vol. 1, 1993, pp. 142–148.
18. Y. Wong and A. Sideris, "Learning convergence in the cerebellar articulation controller," *IEEE Trans. Neural Networks,* **3,** 115–121, 1992.
19. D. E. Whitney, "Historical perspective and state of the art in robot force control," *Proc. IEEE Int. Conf. Robot. Automat.,* 1985, pp. 262–268.