Real-Time Neural Network Control of a Biped Walking Robot

W. Thomas Miller, III

Cerebellar model arithmetic computer (CMAC) neural networks can be applied to the problem of biped walking with dynamic balance. The project goal here is to develop biped control strategies based on a hierarchy of simple gait oscillators, PID controllers, and neural network learning, that do not require detailed kinematic or dynamic models. While results of simulation studies using two-dimensional biped simulators have appeared previously, the focus in this article is on real-time control studies using a ten axis biped robot with foot force sensing. This ongoing study has thus far produced several preliminary results toward efficient walking. The experimental biped has learned the closed chain kinematics necessary to shift body weight from side-to-side while maintaining good foot contact, has learned the quasi-static balance required to avoid falling forward or backward while shifting body weight from side-to-side at different speeds, and has learned the dynamic balance required in order to lift a foot off of the floor for a desired length of time, during which the foot can be moved to a new location relative to the body. Using these skills, the biped is able to march in place and take short steps without falling (too often).

Challenges of Bipedal Locomotion

The challenges of bipedal locomotion have intrigued robotics researchers for a number of years. The first successful solutions emphasized walking with static balance, passing through a succession of states of static equilibrium. This resulted in undesirable constraints on both the biped structure and walking efficiency. Generally, static bipedal walkers had large feet and moved slowly. Summary discussions of the early history of biped walking machines have been presented by Todd [1] and Raibert [2].

When walking with dynamic balance, the projected center of mass is allowed outside of the area inscribed by the feet, and the walker may essentially be falling during parts of the gait cycle. A foot must be moved so as to catch the biped at the proper instant, breaking the fall and achieving a desired net translational acceleration or deceleration. The control problems for dynamic walking are more complicated than for walking with static balance, but dynamic walking promises to provide higher walking speeds and greater efficiency with more versatile walking structures. Numerous investigators have discussed specific theoretical results relevant to the dynamic modeling and control of biped robots. Extensive considerations of the major issues in walking with dynamic balance have been presented by Raibert [2] and Vukobratovic et al. [3], among others.

model of the multilink structure to perform foot placement planning, and developed decoupled control systems for body attitude, body height, and foot placement. Zheng [6]-[8] studied dynamic walking using an eight degree-of-freedom biped (the SD-2). He developed an explicit model for the multilink biped dynamics, and designed a feedback controller which optimized the trajectory of the center of mass using stability margin criteria. Furusho and Sano [9] used independent control models for sagittal and lateral plane motions of an experimental biped robot (the BLR-G2). Their work focused on the role of force/torque control of the sole and ankle during dynamic walking. Kajita et al. [10] developed simple linear differential equations for an ideal biped with massless legs, assuming that the body mass was restricted to horizontal motions. This simple model was used to control a four degree-of-freedom biped with low-mass legs. These systems all exhibited some success on horizontal surfaces.

In addition, the SD-2 robot was able to walk on modest slopes [8]. However, none of these systems represents a general purpose solution to the problems of walking with dynamic balance, in the sense of being able to achieve a variety of stepping rates and lengths, starting and stopping at arbitrary times, walking on various grades, and so forth.

The problems in achieving good performance under varying conditions have led several investigators to the study of on-line gait adaptation. Wagner et al. [11] reported a rule-based strategy for switching on-line between predefined gait controllers. The same group also investigated the use of a linear adaptive model for step length control as a function of body attitude and velocity, using on-line least squares adaptation [12]. Igarashi and Nogai [13] reported a technique for adaptively combining members of a set of predefined gaits in order to handle variations in walking command parameters (such as step length) and walking environments. The above studies involved at least partial control system evaluation using experimental bipeds.

Other investigators have studied the use of neural network learning for on-line gait modification. This approach makes possible the learning of new gaits which are not weighted combinations of predefined gait. Kitamura et al. [14] proposed a walking controller based on a Hopfield neural network in combination with an inverted pendulum dynamic model. The optimization function for the Hopfiled network was derived using a detailed model of the biped kinematics. Miller et al. [15] combined standard supervised learning and temporal difference learning in order to achieve gait adaptation for a simulated two-dimensional biped with massless legs. This work focused on learning appropriate gait adaptations for achieving sudden body translational accelerations and deaccelerations, and for recovering from unexpected disturbance forces, starting from a model of steady walking. Similar issues were studied by Latham [16], using a more realistic two-dimensional biped simulation which accounted for leg masses and foot/ground impact forces. His approach emphasized adapting gaits derived from an inverted pendulum model, in order to accommodate the unmodeled (in the controller) aspects of the biped dynamics. Salatian and Zheng [17], [18] studied both off-line and on-line reinforcement learning techniques for adapting a gait designed for horizontal surfaces, in order to walk on sloping surfaces. These algorithms were evaluated using a biped dynamic simulation.

This article presents preliminary real-time results of a study of the application of on-line neural network learning to the problem of bipedal walking with dynamic balance. The research involves learning control studies using an experimental biped structure inspired by the SD-2 developed by Zheng [6]-[8], but with two additional motors representing knee joints. The robot had five rotational joints per leg: two at each hip, one at each knee, and two at each ankle (a total of ten actuators). Feedback was provided by ten joint position sensors and four force sensors in the sole of each foot (a total of 18 sensors). The long-term goal of the project is to develop biped control strategies based on a hierarchy of simple gait oscillators, PID controllers, and neural network learning, but requiring no detailed kinematic or dynamic models. The goal of the research reported here was to develop low level on-line learning control strategies which would enable the biped to balance during changes in standing posture, and to link short steps without falling.

**Biped Structure**

The biped structure is shown in Fig. 1. The biped as used in these experiments was approximately 0.1 m tall from foot to hip, and 43 cm tall from hip to the top of the body. The separation between the legs was 20 cm. Each foot (a flat metal plate) was 7 cm wide and 12 cm long, with the ankle attached near the center-rear corner of the foot. Each leg (links plus actuators) weighed approximately 2.7 kg. The body and fixed hip structure weighed approximately 2.7 kg, and had an additional 1.4 kg lead weight attached at the top.

Each hip and ankle was actuated by two gearmotors, one for rotation of the leg towards the front of the biped and one for rotation towards the side. Each knee was actuated by a single gearmotor. For the purposes of this paper, the hip, knee, and ankle axes which moved the leg towards the front and back will be referred to by the names Hip_Y, Knee, and Ankle, where the boldface name refers to a vector of two values (one for each leg). Similarly, the hip and ankle axes which moved the leg towards the side of the biped will be referred to by the names Hip_X and Ankle_Y. The ten gearmotors were driven by linear voltage
amplifiers interfaced to the control computer via independent 8 bit D/A converters. The positions of the ten joints were sensed by potentiometers interfaced to the control computer via a single 12 bit A/D converter with an analog multiplexer. The effective repeatability of the position sensing was approximately ±0.15°, including both quantization and noise effects. Absolute resolution was not determined.

Polymer thick film force sensing resistors were mounted on the underside of each foot, near each corner (four 1-in diameter sensors per foot). Each sensor was covered with a thin disc of rubber for shock absorption and to focus the foot/floor contact forces on the sensors. The eight force sensors were interfaced to the control computer via a second 12 bit A/D converter with analog multiplexer. The approximate resolution of the discrete force sensor measurements was 0.2 N, with sensor circuit saturation at a total force of approximately 25 N. However, while the measured force signals increased monotonically with increasing force, the circuits were not strictly linear and no effort was made in software to linearize or calibrate the measurements.

The measurements from the eight force sensors were used to derive six relative "center-of-force" measures which were used directly by the controller. For the right foot, the relative front/back and right/left locations of the center-of-force were computed on a scale of -1 to 1 as follows:

\[
CF_{fr, r} = \frac{F_{fr, r} + F_{fr, l} - F_{fb, r} - F_{fb, l}}{F_{fr, r} + F_{fr, l} + F_{fb, r} + F_{fb, l}}
\]

\[
CF_{rl, r} = \frac{F_{fr, r} + F_{fr, l} - F_{fr, r} - F_{fr, l}}{F_{fr, r} + F_{fr, l} + F_{fr, r} + F_{fr, l}}
\]

In the above equations, each term of the form \(F_{fr,l}\) represents a measurement from an individual force sensor (right-front sensor on the right foot, for example). Center-of-forces were also computed for the left foot. The biped overall center of force \(CF_{fr}\) was similarly computed based on the relative total forces on the right and left feet. In addition, the biped overall front/back center-of-force \(CF_{fb}\) was computed as the weighted average of the \(CF_{fr}\) and \(CF_{fl}\) values for the two feet (the weighting factors were the fractions of the total force sensed at each foot).

Control computations were carried out on a 33 MHz 80486 personal computer with a network of three 25 MHz T425 transputers. Sensor/actuator interface management, actuator level control, and the user interface were implemented on the 80486 processor. Gait level control was carried out on the transputer directly interfaced to the 80486 system, and neural network computations were carried out on the two additional transputers, which acted as slaves to the first transputer. All computations were performed using 16 or 32 bit fixed point operations. Joint position measurements and commands were manipulated internally in 0.01° units. Lengths were manipulated in 0.01 cm units. Forces were manipulated in 0.2 N units.

**Actuator Level Control**

The drive signal to each actuator was provided by a PD controller implemented in software, with a 500 Hz control update rate per actuator. Actuator gains were set experimentally to the maximum values consistent with joint stability. It was found that higher gains could be used when the corresponding leg was in compression (i.e., the foot was in contact with the floor) than when the leg was suspended during a step. For this reason, a simple gain switching strategy was used at each actuator, with the switching occurring whenever the corresponding foot made or broke contact with the floor. New reference positions for the ten actuator controllers were provided every 28 ms (35.7 Hz update rate) by the gait control software implemented on the transputer network. The biped structure was in general sensitive to vibrations caused by abrupt changes in actuator commands. Position command slew rate limits implemented in the actuator control algorithms served to smooth the position command updates. Dither signals with uniform pseudo-random distributions in the range ±1.4 V were added to the ten motor drive signals (±12 V range) to reduce gear box friction effects.

**CMAC Neural Network**

We have been involved in the study of the application of CMAC (cerebellar model arithmetic computer) neural networks [19]-[21] to a variety of control problems. In a series of real time experimental [22]-[24] studies, neural network learning has been shown to be effective for both kinematic and dynamic control problems in robotics. In accordance with our experimental success, a hardware CMAC neural network design was developed which provides submillisecond response and training times for typical CMAC neural networks with tens of 16 bit inputs, 1-8 16 bit outputs, and 512K 16 bit weights [25]. These specifications are suitable for many real-time control problems.

Detailed discussions of the properties of CMAC neural networks and of our implementations are beyond the scope of this report, and have been presented in detail elsewhere [19]-[21], [25]. The following discussion thus focuses on the relevant characteristics of CMAC, without justifying the stated characteristics mathematically. The CMAC neural network is basically an associative memory technique for storing vector responses to vector inputs. The input vector space is tiled with overlapping multidimensional receptive fields of fixed size, corresponding to the interiors of rectangular solids. The tiling pattern assures that each vector input falls into the same number of receptive fields (referred to as the receptive fields "excited" by the input). This constant number of excited receptive fields is called the generalization parameter. Each receptive field contains a vector response, and the neural network output for a given input vector is taken as the average of the responses stored in the receptive fields excited by the input. Since the output is linear relative to the responses stored in the excited receptive fields, standard LMS adaptation can be used to adjust the stored responses for specific input/output training targets.

Note that if the generalization parameter is one, the CMAC algorithm corresponds to table lookup. If the generalization parameter is much greater than one (as is usually the case), the CMAC algorithm corresponds to the nearest-neighbor smoothed table lookup. As with any table lookup approach, arbitrary vector transformations can be approximated, the approximation quality depending on the density and size of the receptive fields in the input vector space. These aspects are controlled by two parameters for each component of the input vector which are set a priori: the fine quantization and the coarse quantization. The fine quantization refers to the minimum change required in the input in order to guarantee that at least one receptive field is different in
Biped Adaptive Posture Control System

Fig. 2: A block diagram of the biped gait control system. The signals inside angle brackets indicate measured quantities, while the signals without angle brackets indicate quantities generated by the controller. For example, <Knee> represents a vector of two measured knee joint angles (one for each leg), while Knee represents a vector of two commanded knee joint angles. The Δ symbol represents a discrete estimate of the time derivative. The three signal paths labeled using upper case letters (STEP LENGTH, STEP INTERVAL, and LIFT MAGNITUDE) are supervisory constants set by the operator. All other signals are either measured or are generated by the controller.

the set excited by the new input, relative to the set excited by the old input. Changes in an input which are less than the fine quantization may produce no change in the output. The coarse quantization refers to the minimum change required in the input in order to guarantee that all receptive fields in the set excited by the new input are different than those in the set excited by the old input. Changes in an input which are greater than the coarse quantization will produce an independent output.

The literature on neural network learning for control is too extensive to be reviewed in this report [26], [27]. In the control structure reported here, CMAC neural networks were used essentially as context sensitive integral error terms in the controller, where the control context was defined by the CMAC input vector. The continuous on-line training integrated control errors over successive visits to the same region of the CMAC vector input space, rather than over contiguous control cycles as in a standard integral control term. Thus, after training, the neural networks produced different control adjustments for different regions of the input space. This technique is commonly referred to as "feedback error learning" in the neural networks literature.

Walking Gait Control

Due to the distribution of mass within the structure, the biped was not capable of walking with static balance. A foot could not be lifted slowly without falling. In order to move a foot, it was necessary to generate a lateral momentum toward the opposite side by knee extension and hip rotation, sufficient to lift the foot when knee extension terminated. The resulting gravitation force with the foot lifted would then break the momentum and allow the biped to fall back on to the lifted foot. The side-to-side and foot movement motions in this process were initiated by a simple gait oscillator which drove the knee extensions and hip angles. Three CMAC neural networks were used in combination with the simple gait oscillator and the actuator-level PD controllers to create the control structure shown in Fig. 2.

The Fixed Gait Generator module received constant values for the total step length (in 0.01 cm units), total step interval (in control cycle units), and maximum foot lift magnitude (in 0.01 cm units) from a supervisory control level not shown in the figure (a keyboard command input level). The module then repetitively generated hip and knee position reference commands for each control cycle during eight gait phases (four for each leg) as follows:

1. **Left Leg Extension**: Extend the left knee at maximum drive voltage, bringing the hip forward to a position over the left foot (via approximate kinematics). Terminate when the left knee position passes a target value supplied by the Right/Left Balance CMAC.

2. **Left Foot Movement**: Lift the left foot and translate it smoothly to a new position using open-loop control (via approximate kinematics). Terminate when the left foot contacts the floor.

3. **Left Leg Relaxation**: Shorten the left leg, bringing the hips forward to a position between the right and left feet (via approximate kinematics). Terminate after a fixed time interval.

4. **Pause**: Wait for the integrated AC energy from the CF}_{ FB} signal to decay. During all gait phases, the CF}_{ FB} signal was high-pass filtered, rectified, and filtered again using a "leaky" integrator to produce a "quality of balance" signal. The pause phase of the gait did not terminate until the integrated signal fell below some threshold. As a result, the robot automatically paused for a longer time between steps if the balance during the prior gait phases had been poor, but proceeded to the next phase after a shorter pause if the balance had been good. Adjustment of the termination threshold for this phase affected the degree of step-to-step coupling at a given level of training.

5-8. **Repeat for the right leg**: Phases 1-8 were automatically repeated in a continuous cycle until stopped by the operator or until a fall was detected.

The **Gait Phase** signal in Fig. 2 was a constructed two-dimensional cyclic signal of the form {sin(θ), cos(θ)}, where θ varied smoothly from 0 to 2π as the eight gait phases progressed. The two-dimensional form of this signal was intended to prevent discontinuity in the neural network inputs as θ switched from 2π to 0 during the transition from phase 2 to phase 1.

Since it was impossible to sense foot position relative to the floor or body, the gait control algorithms used a simple planar model of the leg kinematics (the same model was used independently for both legs), as shown in Fig. 3 in a lateral view. The inputs to the model were two fixed leg segment lengths (thigh and shin) and the desired horizontal and vertical displacements of the foot relative to the body. The model then produced the corresponding approximate hip and knee joint angles. Note that the model totally ignored the coupling between the leg segment orientations in the frontal and lateral planes, and the coupling between the two legs via their common connection to the body.

The Right/Left Balance CMAC (CMAC)_R neural network was used to predict the correct knee extension (in 0.01° units) required to achieve sufficient lateral momentum in order to lift the corresponding foot for the desired length of time (Knee Extension Target in the figure). The ten inputs to this CMAC included the foot lift magnitude command (0.05/1.60 cm), the
The approximate leg kinematics used by the gait generator is shown in Fig. 3.

---

Step length commands (0.05/1.60 cm), the two gait phase signals (0.004/0.128), the two measured \(<\text{Hip}_Y>\) joint positions (0.25/8.00°), the two measured \(<\text{Knee}>\) joint positions (0.25/8.00°), and the two estimated \(<\text{Ankle}_Y>\) joint velocities (0.02/0.64°/cycle). For each input parameter, the first number refers to the CMAC fine quantization, and the second number refers to the CMAC coarse quantization. The CMAC generalization parameter was set to 32, and the total number of available receptive fields was 20,000.

An insufficient knee extension would cause the foot to lift for too short a duration (or not at all), while an overextension would cause the foot to lift for too long a duration (or for the robot to fall over laterally). The proper amount of extension was dependent on the configuration of the robot at the beginning of the knee extension phase, which varied somewhat from step to step and was trained using temporal difference learning (i.e., the CMACrl output at time \(t\) was used as the desired training target for the CMACrl response at time \(t-1\)).

The training signal \(\Delta KET(t)\) was computed as:

\[
\Delta KET(t) = 0.25 \cdot (KET(t) - KET(t-1)) + 15 \cdot DE(t)
\]

where \(KET(t)\) represents the CMACrl response at time \(t\), \(KET(t-1)\) represents the change in the CMACrl response (given the input vector used at time \(t-1\)) as the result of training, and \(DE(t)\) is a signal which had the value 1 if the raised foot contacted the floor on that control cycle and was sooner than expected, had the value \(-1\) if the raised foot contacted the floor on that control cycle and was later than expected, and had the value 0 for all other control cycles. The values 0.25 and 15 are somewhat arbitrarily selected learning system gains, which scale the individual weight adjustments terms (for example, when \(DE(t)\) was 1, the value of the knee extension threshold predicted by the neural network was adjusted by 15 units, corresponding to 0.15°).

The Closed-Chain Kinematics CMAC (CMACk) neural network was used to learn kinematically consistent robot postures. The six inputs to this CMAC included the two measured \(<\text{Hip}_Y>\) joint positions (0.25/8.00°), the two measured \(<\text{Hip}_Y>\) joint positions (0.25/8.00°), and the two measured \(<\text{Knee}>\) joint positions (0.25/8.00°). The four outputs of this CMAC included adjustments to the two \(\text{Ankle}_Y\) joint position commands (0.01° units) and two \(\text{Ankle}_Y\) joint position commands (0.01° units). The CMAC generalization parameter was set to 32, and the total number of available receptive fields was 20,000.

Whenever it was desired that both feet be in solid contact with the floor (the double-support phases of the biped gait), the closed-chain kinematics of the structure had to be addressed. Target positions for all ten motors could not be specified independently. In our implementation, knee extension and hip frontal and lateral rotation for each leg were specified independently (the reference positions for the two lateral hip rotations, \(\text{Hip}_Y\) in the figure, were held constant at 0° in the reported trials). The CMACk neural network then predicted appropriate positions for the frontal and lateral ankle rotations in order to keep the biped's feet parallel to the floor with spatially balanced forces on the sole of each foot. The training rules for this network were as follows.

If the center-of-forces of the two feet were nearer to opposite right/left edges of the feet, CMACk was trained to counter rotate the two \(\text{Ankle}_Y\) axes, making the feet more parallel. If the center-of-forces of the two feet were nearer to the same right or left edges of the feet, CMACk was trained to commonly rotate the two \(\text{Ankle}_Y\) axes, with the effect of tilting the robot. If the center-of-forces of the two feet were nearer to opposite front/back edges of the feet, CMACk was trained to counter rotate the two \(\text{Ankle}_Y\) axes, making the feet more parallel.

The CMACk training adjustment for the \(\text{Ankle}_Y\) command was thus computed as:

\[
\Delta \text{Ankle}_Y(t-1) = 0.125 \cdot \left(\text{Ankle}_Y(t) - \text{Ankle}_Y(t-1)\right)
\]

\[
\pm 0.0005 \cdot \left(\text{CF}_{\text{fr}_t\text{r}_t}(t) - \text{CF}_{\text{fr}_t\text{l}_t}(t)\right) \cdot \text{F}_{\text{fr}_t}(t) \cdot \text{F}_{\text{fr}_l}(t)
\]

where \(\text{Ankle}_Y(t)\) represents an individual CMACk response, \(\text{CF}_{\text{fr}_t\text{r}_t}(t)\) and \(\text{CF}_{\text{fr}_t\text{l}_t}(t)\) are the relative front/back center of forces for the right and left feet, and \(\text{F}_{\text{fr}_t}(t)\) and \(\text{F}_{\text{fr}_l}(t)\) are the total forces on the right and left feet (0.2 N units). The latter term was added to the training signal for the right ankle and was subtracted from the training signal for the left ankle. The CMACk training adjustment for the \(\text{Ankle}_Y\) command was computed as:

\[
\Delta \text{Ankle}_Y(t-1) = 0.125 \cdot \left(\text{Ankle}_Y(t) - \text{Ankle}_Y(t-1)\right)
\]

\[
+ 0.0005 \cdot \left(\text{CF}_{\text{lf}_t\text{l}_t}(t) + \text{CF}_{\text{lf}_t\text{r}_t}(t)\right) \cdot \text{F}_{\text{lf}_t}(t) \cdot \text{F}_{\text{lf}_l}(t)
\]

\[
\pm 0.0005 \cdot \left(\text{CF}_{\text{lf}_t\text{l}_t}(t) - \text{CF}_{\text{lf}_t\text{r}_t}(t)\right) \cdot \text{F}_{\text{lf}_t}(t) \cdot \text{F}_{\text{lf}_l}(t)
\]

where \(\text{Ankle}_Y(t)\) represents an individual CMACk response, \(\text{CF}_{\text{lf}_t\text{r}_t}(t)\) and \(\text{CF}_{\text{lf}_t\text{l}_t}(t)\) are the relative right/left center of forces for the right and left feet, and \(\text{F}_{\text{lf}_t}(t)\) and \(\text{F}_{\text{lf}_l}(t)\) are the total forces on the right and left feet (0.2 N units). The latter term was added to the training signal for the right ankle and was subtracted from the training signal for the left ankle. Note that during single-sup-
port phases of the gait (only one foot in contact with the floor), either $F_x(t)$ or $F_y(t)$ was zero and training occurred solely from the temporal difference term. The values 0.0125 and 0.0005 are somewhat arbitrarily selected learning system gains, which scale the individual weight adjustment terms.

CMACk was trained initially during slow trial movements based on individual foot center-of-force feedback. After training, the biped could stand still or move slowly through a range of statically balanced postures (swaying side-to-side) without falling. During normal walking, CMACk was trained using supervised learning based on error signals derived from the foot center-of-force measurements (as during the slow movements) whenever both feet were in contact with the floor. Whenever only a single foot was in contact with the floor, CMACk was trained using temporal difference learning. This enabled CMACk to predict the correct ankle orientations such that a lifted foot would land squarely on the floor.

The Front/Back Balance CMAC (CMACfb) neural network was used to provide for front/back balance during standing, swaying and walking. The eight inputs to this CMAC included the foot lift magnitude command (0.05/1.60 cm), the step length command (0.05/1.60 cm), the two gait phase signals (0.004/0.128), the two measured <Hip Y> joint positions (0.25/8.00°), and the two measured <Ankle> joint positions (0.25/8.00°). The CMAC generalization parameter was set to 32, and the total number of available receptive fields was 20000.

The single output of CMACfb (0.01° units) was added directly to each of the commanded hip frontal rotations. The CMACfb neural network was trained using the biped overall front/back center-of-force from the foot force sensors as the training error signal, providing adjustments to the hip frontal rotations in order to provide net toe/heel foot force balance during the stepping motions. This had the effect of preventing the biped from falling forward or backward. It was recognized that being in a state of imbalance at a given time during walking resulted from incorrect postures at earlier times, rather than from an incorrect current posture. Thus, input/output pairs for CMACfb were buffered and the immediate balance error was used to modify the CMACfb response from 16 control cycles (0.45 s) previous. The training signal for CMACfb also contained a smoothing term which attenuated second time derivative energy in the CMACfb output. In addition to smoothing the CMACfb output, this had the effect of helping to distribute the information from the delayed supervised learning over sequential time steps.

The CMACfb training adjustment was thus computed as:

$$\Delta \text{Hip}(t-16) = 0.0625 \cdot (\text{Hip}(t-17) - \text{Hip}(t-15) - 2 \cdot \text{Hip}(t-16) + 20 \cdot \text{CF}_{fb}(t))$$

where Hip(t) represents the CMAC response, and CF_{fb}(t) represents the overall front/back center-of-force. The values 0.0625 and 20 are somewhat arbitrarily selected learning system gains, which scale the individual weight adjustment terms.

**Results**

Training of the biped typically proceeded as follows. First, CMACk and CMACfb were trained in a series of slow side-to-side swaying motions without attempting to lift the feet. This provided initialization of the closed-chain kinematics required to give solid foot/floor contact and provided reasonable initial balance. The slow training normally was carried out for about ten minutes. Generally, the CMAC learning was sufficiently fast that the biped only needed human support during the first minute. Next, all three CMAC neural networks were trained during repetitive foot lift motions similar to marching in place (i.e., no attempt was made to translate the lifted foot). This typically was carried out for another ten minutes, with different settings for desired foot lift height (in the range 0.5 to 2.5 cm). Frequent human support was required to keep the biped from falling during the first half of this training, and occasional support was required during the second half. Finally, training of all three CMAC neural networks was carried out during attempts at walking (translating the lifted foot forward) for step lengths in the range of 1 to 6 cm (the distance that one foot was placed in front of the other). Again, frequent human support was required during early training for each new step length setting, while less frequent support was required after 5 to 10 min of training at a given step length.

After about 1 h of total training time, the biped was able to reliably walk at a rate of approximately 30 steps/min for a distance of 3.5 m (the working range of the tether) for step lengths from 1 to 4 cm. For step lengths of 5 to 6 cm, the biped was able to walk the distance reasonably reliably, but human intervention was required more often. For step lengths beyond 6 cm, left/right stability became a problem and the biped required support more frequently (the CMACfb provided the correct side-to-side motions for the lifting of each foot during stepping, but there was nothing in the control structure to adaptively augment left/right balance during the remainder of the step phases).

Given the multisensor, multi-actuator nature of the experimental biped it is difficult to concisely present quantitative

---

**Fig. 4. Biped balance data during attempts to walk without training.**

The front/back force signal (CF_{fb}) is 1 when all of the force is on the front force sensors, and is -1 when all of the force is on the back force sensors. The right/left force signal (CF_{rb}) is 1 when all of the force is on the right foot, and is -1 when all of the force is on the left foot. The front/back balance control signal (Hip) is the output of the CMACfb neural network. The Phase # signal progresses from 1 to 8, corresponding to the eight gait phases described in the text.
While general purpose walking has not yet been achieved, the experimental biped was able to learn the closed chain kinematics necessary to shift body weight from side-to-side while maintaining good foot contact, and was able to learn the quasistatic balance required to avoid falling forward or backward while shifting body weight from side-to-side at different speeds. It was able to learn the dynamic balance required in order to lift a foot off of the floor (via rapid knee extensions) for a desired length of time and for different initial conditions, during which time the foot could be moved to a new location relative to the body. Following human guided practice, the biped was able to march in place and to link short steps without falling (too often). Step length was found to be limited by the lack of any adaptive control of right/left balance (which is inherently stable for short steps but becomes less stable as step length increases). Experiments are currently proceeding in an attempt to overcome this limitation by using an additional CMAC to control the lateral hip axes (Hip_y) which were held stationary during the experiments described above.

The biped controller relied on simple fixed strategies augmented by neural network learning. No detailed kinematic or dynamic models were used. The real-time hierarchical controller was implemented entirely in fixed point operations on standard processors. One processor was used for hardware interface management and actuator control, one processor was used for gait control, and two processors were used for CMAC neural network computations. The four processor configuration had ample reserve computing capacity to allow the logical expansion of the control architecture.

As a mechanical structure, the biped was fundamentally unstable, particularly with regard to front/back motions. In addition, gear box play accumulated through the series arrangement...
of ten gearmotors resulted in significant structural wobble, even though the individual links were rigid. As a result, the biped was sensitive to incorrect postures and to vibrations induced by sudden changes in control signals. This made it unsuitable for control using purely reactive, error driven control. Good performance depended on accurate and smooth feedforward control, which was provided by the neural network learning in lieu of detailed system modeling.

In the above process, step length was preset by the operator and did not adapt. The controller learned the correct time sequences of body postures in order to complete the commanded steps without falling. This is only possible if desired net body accelerations are small. Previously reported results by our laboratory have included simulation studies of learned foot placement using two-dimensional bipedal simulators [15], [16]. In the simulation studies, standard supervised learning and temporal difference learning were combined to train the neural network. Random training using frequent sudden changes in desired average body velocity produced a robust controller able to track sudden changes in the desired body velocity command, and able to rapidly adjust to unexpected disturbances. Our next goal in the real-time experiments is to couple this learned foot placement strategy with our current learned control of static and dynamic balance during stepping, in order to achieve effective locomotion.

References


W. Thomas Miller, III, received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in bioengineering from the Pennsylvania State University in 1972, 1974, and 1977, respectively. He served as a Postdoctoral Fellow in biomedical engineering at Duke University, North Carolina, from 1977 to 1979. He has been a member of the Faculty in the Department of Electrical and Computer Engineering at the University of New Hampshire since 1979. He currently holds the rank of Professor. His current research interests include the design of neural architectures for problems in control and pattern recognition.