Robotica (2004) volume 22, pp. 29–39. © 2004 Cambridge University Press
DOI: 10.1017/S0263574703005253 Printed in the United Kingdom

Frontal plane algorithms for dynamic bipedal walking

Chee-Meng Chew* and Gill A. Pratt†

(Received in Final Form: May 4, 2003)

SUMMARY
This paper presents two frontal plane algorithms for 3D dynamic bipedal walking. One of which is based on the notion of symmetry and the other uses reinforcement learning algorithm to learn the lateral foot placement. The algorithms are combined with a sagittal plane algorithm and successfully applied to a simulated 3D bipedal robot to achieve level ground walking. The simulation results showed that the choice of the local control law for the stance-ankle roll joint could significantly affect the performance of the frontal plane algorithms.

KEYWORDS: Bipedal walking; Frontal plane algorithms; Control law; Stance-ankle joint roll.

1. INTRODUCTION

In our previous paper,[1] a general control architecture was proposed. In the architecture, we assume that the motion in the three orthogonal planes (sagittal, frontal and transverse) can be independently considered. Three-dimensional walking algorithms can then be formulated by synthesizing the control algorithms for each of the orthogonal planes. The paper presented a few examples of control algorithms for the sagittal plane motion control.

In this paper, the control algorithms for the frontal plane are described. It is responsible for the lateral balance of the bipedal walking. In comparison with the sagittal plane motion control, much fewer strategies for the frontal plane motion control have been proposed. Goddard et al.[2] and, Hemami and Wyman[3] used a three-link planar model to study the frontal plane motion of a biped. Iqbal et al.[4] presented a model for studying the involvement of the central nervous system in the execution of voluntary movements in the frontal plane.

In this paper, two strategies for the frontal plane are explored. In the first implementation, a reinforcement learning algorithm is used to learn the lateral foot placement of the swing leg at the end of each step such that a stable lateral motion can be achieved. The other implementation uses the notion of symmetry to generate the control law for the lateral swing leg behavior. The reinforcement learning algorithm is used, if necessary, to learn the offset value for the control law.

The algorithms for the frontal and sagittal planes are then combined and applied to a simulated 3D bipedal robot "M2" (Figure 1) to achieve dynamic walking behaviors. In the simulation, we apply two different local control laws at the stance-ankle roll joint. Simulation results reveal that an appropriate choice of the local control law is crucial for the algorithms to be successful.

The organization of this paper is as follows. Section 2 presents the heuristics of the frontal plane motion for bipedal walking. Section 3 describes the proposed frontal plane motion control algorithms. The two strategies mentioned earlier are introduced. Section 4 studies the effects of the local control law at the stance-ankle roll joint on the learning rate for the first strategy. Two local control laws are compared by applying the overall control algorithm to the simulated M2. Section 5 presents the implementation of the frontal plane strategy that is based on the notion of symmetry. Simulation studies are used to demonstrate the effectiveness of such a strategy. An analysis is carried out to verify the approach.

2. HEURISTICS OF FRONTAL PLANE MOTION

In the frontal plane, the biped uses lateral foot placement to maintain its lateral balance. During steady forward walking, the body's lateral velocity varies periodically like a sinusoidal waveform.[5] The body needs to swing from side to side because each leg alternatively bears the weight of the body and the legs are separated in lateral direction.[6] The average lateral velocity has to be zero if the biped follows a straight line walking path. At each step, the lateral velocity of the body is expected to change sign exactly once.

If we consider the legs' inertia in the frontal plane to be negligible, the body height to be maintained at a constant value (in level ground walking), the stance ankle to be unactuated, and the body posture to be upright; the frontal plane motion can be coarsely modelled using a linear inverted pendulum model adopted by Latham[7] and Kajita and Tani[8] as shown in Figure 2. The dynamic equation for the linear inverted pendulum model is as follows:

$$\ddot{x}=\frac{g}{h}x \tag{1}$$

where x is the horizontal coordinate of the body's center of mass from a vertical plane that passes through the ankle joint, g is the gravitational constant and h is the height.

Equation 1 can be integrated to give the relationship between $\dot{x}$ and $\ddot{x}$:

$$\frac{\dot{x}^2}{2}=\frac{gx^2}{2h}+C \tag{2}$$

where C is the integration constant which Kajita, Tani and Kobayashi[9] called the *orbital energy*.

* Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260.
† F.W. Olin College of Engineering, 1735 Great Plain Ave., Needham, MA 02492–1245 (USA)

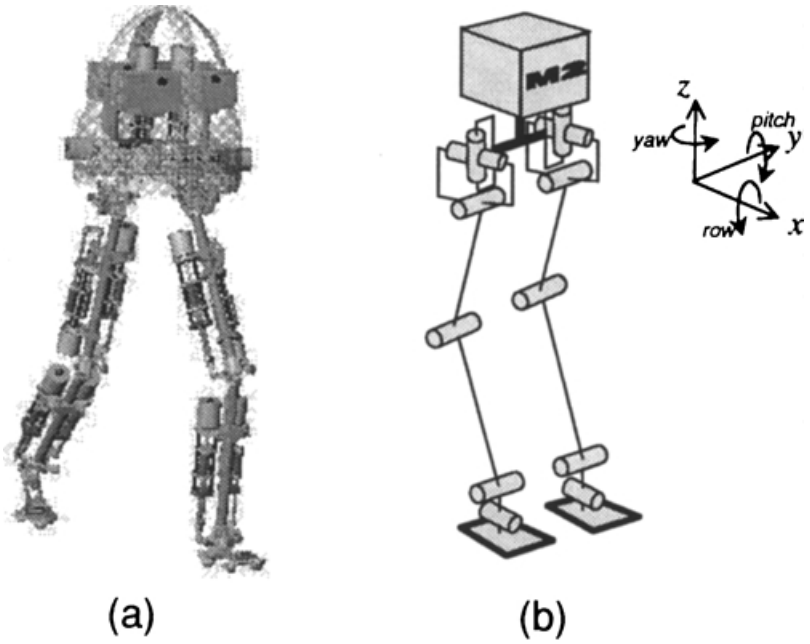

Fig. 1. Three dimensional biped: M2.[1]

Given the velocity $\dot{x}$ and the position x of the system at any instant, C can be computed, and equation 2 defines the relationship between $\dot{x}$ and x for all other time before the next support exchange event.

When C is greater than zero, the mass approaching the vertica plane that passes through the pivoting point will be able to travel across the plane. When C is less than zero, the mass will not be able to travel across the vertical plane. Instead, it reverts its direction of travel at some instance of time. In the frontal plane implementation, C is desired to be less than zero because it is desirable for the body's lateral velocity to change sign within the step.

Since this model is a coarse model for the frontal plane, it is not utilized to plan the swing leg motion. Instead, it is only used for the failure detection in the learning algorithm (see Section 3). The model is also used in the frontal plane local control based on the stance-ankle roll joint (see Section 4).

3. FRONTAL PLANE ALGORITHM

This section describes the control algorithm for the frontal plane. They are designed for the biped walking along a straight path. The algorithm can be decomposed into the stance leg and swing leg sub-algorithms.

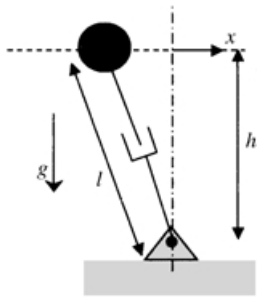

Fig. 2. Massless leg model: Linear inverted pendulum. x is the horizontal coordinate of the point mass from the vertical plane that passes through the ankle joint.

For the stance leg, we have the hip and ankle roll joints. Let's assume that the desired body roll angle with reference to the vertical is zero (upright posture). This can be regulated by installing a PD controller at the stance-hip roll joint. A DC term can be added to account for the moment due to the body and the swing leg weights so that high stiffness is not required for the joint. The stance-ankle roll joint implementation will be presented in Section 4.

For the swing-leg control in the frontal plane, the desired trajectory of the swing-hip roll angle and the swing-ankle roll angle are required. The desired swing-ankle roll angle is obtained by geometric consideration. For example, the swing foot is maintained to be parallel to the ground when the foot is in the air. Once these angles are given, PD controllers are used to generate the desired torques at the respective joints.

The behavior of the swing-hip roll angle is the key determinant for the lateral stability of bipedal walking. There are several ways to generate the trajectory of the desired swing-hip roll angle. Two strategies are presented here. In the first strategy, the desired end position ϕ^f (with respect to the vertical) of the swing leg in the frontal plane is generated before the beginning of the next step using the reinforcement learning method. The trajectory of the roll angle ϕ of the swing leg is then planned using a third degree polynomial. The swing time is assumed to be a constant.

The second strategy is based on the notion of symmetry. The desired swing leg roll angle is set to mirror the actual stance-leg roll angle. The reinforcement learning method is used to learn the offset Δ to the nominal desired swing-leg roll angle provided by the symmetry approach (see Section 5).

Q-learning using CMAC (Cerebellar Model Articulation Control) as the function approximator for the Q-factors[1,10] is adopted to be the reinforcement learning method for both

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

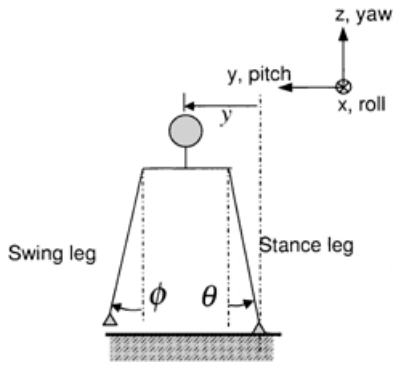

Fig. 3. State variables for the learning implementation in the frontal plane motion are the values of y, ϕ and θ at the end of the support exchange (just before a new step begins).

implementations. The learning aims to avoid the failure states in the frontal plane. The following subsections describe the state variables and reward function used in these strategies.

3.1. State variables

The frontal plane motion control is assumed to be weakly coupled to the sagittal plane motion. Thus, the state variables for the sagittal plane are not considered in the frontal plane learning implementation. The following state variables are chosen (see Figure 3) for the frontal plane motion control:

(1) The velocity of the body in the y-direction, $\dot{y}^+$;
(2) The roll angle of the new swing leg (with respect to the vertical), ϕ^+;
(3) The roll angle of the new stance leg (with respect to the vertical), θ^+.

The superscript + indicates that the variable is detected at the end of the support exchange (just before the beginning of a new step).

3.2. Reward function and return computation

The reinforcement learning algorithm adopts a reward function r that only issues a punishment value when a failure state is encountered:

$$r=\begin{cases} 0 & \text{for } \Theta_l \le \theta \le \Theta_u \text{ and } C_l \le C \le C_u \\ R_f & \text{otherwise (failure)} \end{cases} \tag{3}$$

where θ is the stance-leg roll angle (Figure 3) bounded by Θ_l, and Θ_u; C is the orbital energy bounded by C_l and C_u; and R_f is a negative constant (punishment). Note that the failure condition based on θ is checked at all times. The failure condition based on C is checked only at the end of the support exchange.

For such a reward function, a "farsighted" approach for the return computation has to be adopted. That is, the discount rate γ for the return computation is set to a value close to one (say 0.9).

4. LOCAL CONTROL IN FRONTAL PLANE

This section describes the use of the stance-ankle's roll joint as a local control mechanism to assist the lateral balance. Two local control law candidates will be compared. Considering the right support phase, the first control law candidate for the stance-ankle roll joint's torque τ_{ar} is given as follows:

$$\tau_{ar} = -B_{ar}\dot{y} \tag{4}$$

where B_{ar} is a constant gain and $\dot{y}$ is the lateral velocity of the body. This control law is chosen because the biped average lateral velocity is desired to be zero. It tries to reduce the lateral velocity to zero at all times.

The other local control law candidate is built upon a coarse model for the frontal plane. The linear inverted pendulum model described in Section 2 is selected to be the coarse model. The model has the following explicit solutions:

$$y(t) = C_1 e^{\omega t} + C_2 e^{-\omega t}$$
$$\dot{y}(t) = C_1 \omega e^{wt} - C_2 \omega e^{-wt}$$

where y is the horizontal position of the midpoint between the hips measured from the stance ankle; $\dot{y}$ is the time derivative of y; ω is equal to $\sqrt{g/h}$; C_1 and C_2 are constants that can be computed given the value of y and $\dot{y}$ at a given time.

Based on the linear inverted pendulum model, the desired frontal plane motion during the right support phase can be represented as in Figure 4. Let's assume that the height h is constant. The black circle represents the point mass of the linear inverted pendulum model. The figure illustrates that the body changes the direction of lateral motion once during the single support phase. This is possible only if the orbital energy C is less than zero. For a given swing time T_s, it is desirable by symmetry that the lateral velocity $\dot{y}$ of the body be zero at half time $T_s/2$. Furthermore, for regular walking, it is desirable that $y(T_s/2)$ be the same for every step. Once a value for $y(T_s/2)$ is chosen, the values of C_1 and C_2 can be obtained as follows:

$$C_2 = \frac{y(T_s/2)}{2} e^{\frac{\omega T_s}{2}} \tag{5}$$

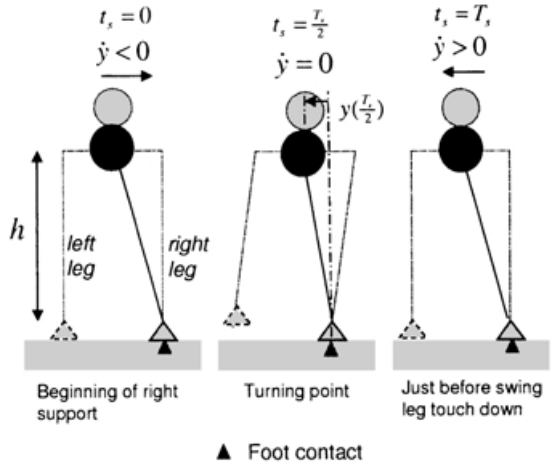

Fig. 4. The frontal plane motion during the right support phase based on the linear inverted pendulum model. The black circle represents the point mass of the linear inverted pendulum model.

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

$$C_1 = C_2 e^{-\omega T_s} \quad (6)$$

For this case, the control law at the stance-ankle roll joint (for the right support phase) is given as follows:

$$\tau_{ar} = K_{ar}(y^d - y) + B_{ar}(\dot{y}^d - \dot{y}) \quad (7)$$

where y^d and $\dot{y}^d$ are the desired lateral position and velocity of the midpoint between the hips (measured from the stance ankle) given by the linear inverted pendulum model; K_{ar} and B_{ar} are constant gains.

The desired torque τ_{ar} from both control laws are bounded to prevent the stance foot from tipping over its side edges. Let's denote τ_{aru} and τ_{arl} to be the upper and lower bounds for τ_{ar}, respectively. Simple static analysis is used to generate the bounds:

$$\tau_{aru} = K_{aru} M\, g l_{fw}/2 \quad (8)$$

$$t_{arl} = -K_{arl} M g l_{fw}/2 \quad (9)$$

where M is the total mass of the biped; g is the gravitational constant; l_{fw} is the foot width; K_{aru} and K_{arl} are the positive discount factors.

4.1. Implementations

In the frontal plane, the reinforcement learning algorithm is used to learn the end position of the swing leg roll angle ϕ_f, whereas the swing time T_s is set to be constant. The algorithm is summarized in Table I. The effectiveness of both local control law candidates will be compared.

In the sagittal plane, we adopt the implementation in which a myopic reinforcement learning agent is used to learn the horizontal end position of the swing foot with reference to the hip.[1] The algorithms for the frontal and sagittal planes are combined to form a three-dimensional (3D) walking algorithm.

There are many ways to implement this algorithm. One approach is to train the sagittal plane's motion control first followed by the frontal plane's, or vice versa. Another approach is to train them simultaneously. The latter approach is more attractive for physical implementation since the former approaches require special training setup to constrain the biped to the sagittal or frontal plane motion. However, the latter approach requires proper partitioning of the failure conditions so that the cause of failure can be correctly attributed, that is, whether it is due to the sagittal or frontal plane motion.

Table I. Reinforcement learning implementation for the frontal plane: F_1.

Description	Remark
Implementation code	F_1
Swing time	Constant
Learning output (action), u	End position of the swing leg roll angle, ϕ_f
Key parameters:	
Reward function	Equation 3
Θ_u	$0.26 rad$
Θ_l	$-0.15 rad$
C_u	0.1
C_u	-0.1
Discount factor γ	0.9 (farsighted)
Action set U	$\{0.001n \mid \text{for } -0.06 \le 0.001n \le 0.1 \text{ and } n \in \mathbf{Z}\}$
Policy	greedy

The 3D algorithm is applied to the simulated biped M2. In every iteration, the biped starts to walk from a standing posture with both ankles directly below the respective hip joints. Its body is given an initial velocity of $[\dot{x}\ \dot{y}\ \dot{z}]^T = [0.6 \text{ m/s}, -0.42 \text{ m/s}, 0 \text{ m/s}]^T$.

The learning processes in both the sagittal and frontal planes are carried out simultaneously. The learning target is to achieve 100 seconds of walking without violating the failure conditions in both the sagittal and frontal planes. The desired walking speed and swing time TB are set at 0.4 m/s and 0.5 second, respectively. In the frontal plane, $y(T_s/2)$ is set to be equal to 1/8 of the hip spacing (0.184 m) for the local control based on equation 7.

4.2. Simulation results

Figure 5 shows the resulting learning curves for the implementations. The solid and dotted graphs are the learning curves for the implementations in which the frontal plane local controls were based on equations 4 and 7, respectively. The latter implementation had a much better performance since it was able to reach the target at around 200 iterations.

For the implementation whose local control is based on equation 7, the simulation data for the walking after the learning target had been achieved are shown in Figure 6. The top graph shows the state of the system. State 5 and 6 correspond to the right-support and the left-support, respectively. State 7 corresponds to the transition from the right-support to the left-support and State 8 corresponds to the transition from the left-support to the right-support. *q.y*, *qd.y* and *q.roll* are the graphs for the horizontal position, horizontal velocity and roll angle of the body with respect to the global frame in the simulation, respectively. *q.lh_roll* and *q.rh_roll* are the roll angles of the left and right legs, respectively. Those variables with "i" preceding them are the state variables for the reinforcement learning algorithm and that with "u" preceding it is the action.

Although the biped was able to walk without reaching the failure state, it is observed that the behavior of the biped in the frontal plane was "chaotic" (from the graphs of *q.lh_roll* and *q.rh_roll*). Furthermore, the action sequence ($u(n)$) from the frontal plane agent did not seem to converge to any regular pattern. It could be due to the fact that there was no mechanism installed to make the agent behave in a regular manner. That is, the agent was just "told" to ensure the biped did not violate the failure conditions, but not to behave in a regular manner. The next section presents an implementation in which regularity for the frontal plane motion is built into the algorithm.

5. NOTION OF SYMMETRY

One key problem of the approach presented in the previous section was that the frontal plane learning agent behaved in a chaotic manner. That is, the desired end position of the swing-hip roll angle did not converge to a limit point or

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

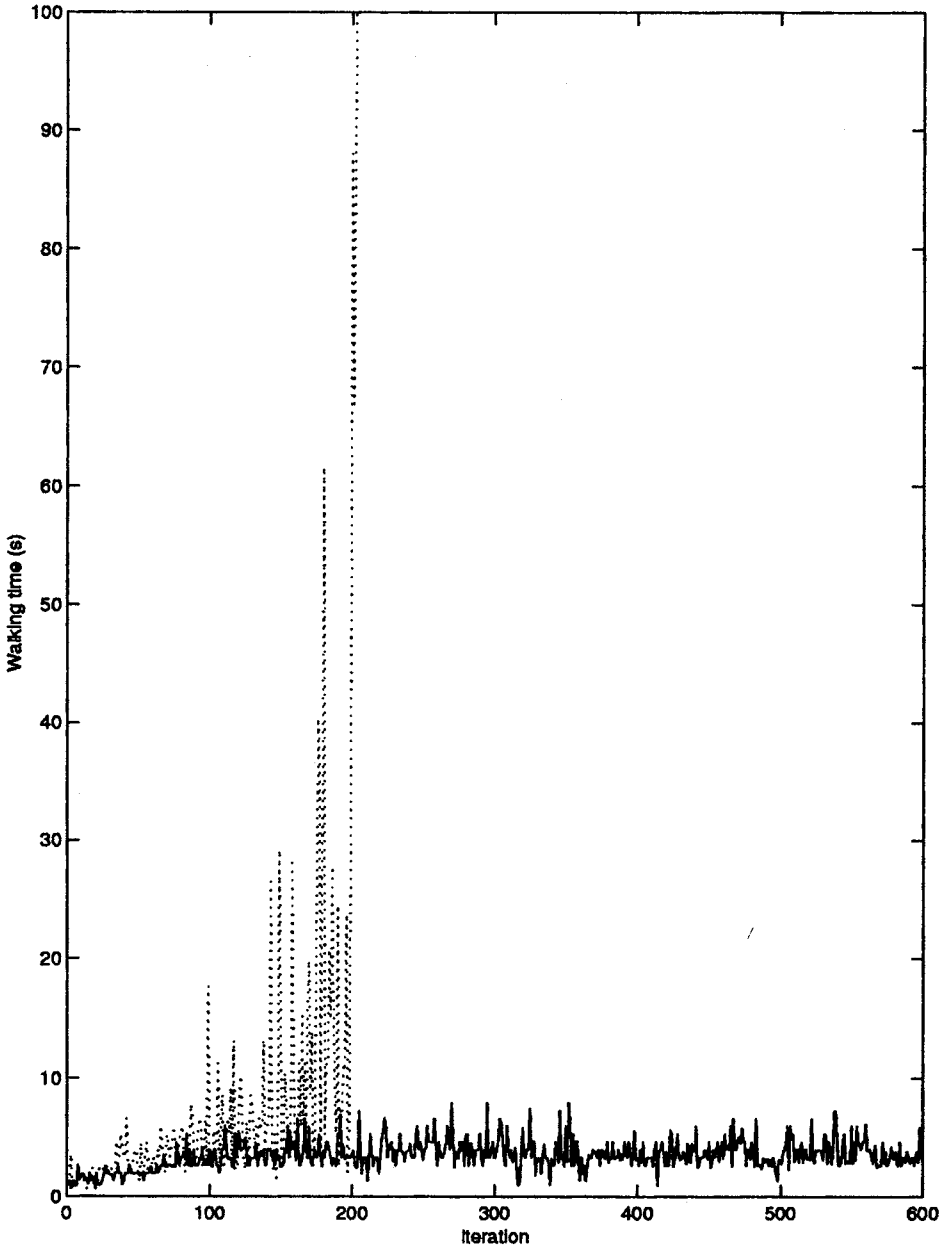

Fig. 5. The learning curves for the simulated biped when implementations introduced in Section 4 were used. The solid-line and dotted-line graphs are the learning curves for the implementations in which the frontal plane's local controls were based on equations 4 and 7 respectively.

regular pattern. One solution is to use a reward function that penalizes such a chaotic behavior. Another approach is to build regularity into the algorithm. The latter approach is adopted in this section and it is based on the notion of symmetry. Ralbert[11] used it to design an algorithm for foot placement to control the forward speed of a hopping machine. His work resulted in a simple implementation which did not require any dynamic model of the legged machine.

5.1. Implementation

The notion of symmetry is a simple but general concept. In the context of the frontal plane motion, this notion is applied to set the desired swing-leg roll angle ϕ of the biped. The desired swing-leg roll angle is set to mirror the behavior of the stance-leg roll angle θ. Assuming that the body roll angle is equal to zero at all times, the control law to generate the desired torque τ_{hrs} *for the swing-hip roll joint* (*for the right support phase*) *can be formulated as follows*:

$$\tau_{hrs} = K_{hrs}(\theta - \phi) + B_{hrs}(\dot{\theta} - \dot{\phi}) \tag{10}$$

where K_{hrs} and B_{hrs} are the proportional and derivative gains, respectively.

The local control law for the stance-ankle roll joint is the same as equation 4. The desired torque τ_{ar} of the stance-ankle roll joint is also bounded by the same bounds (equations 8 and 9). A reinforcement learning algorithm almost similar to the previous implementation is used to learn the offset Δ for the desired swing-hip roll angle before the beginning of each step. That is, equation 10 becomes:

$$\tau_{hrs} = K_{hrs}(\theta + \Delta - \phi) + B_{hrs}(\dot{\theta} - \dot{\phi}). \tag{11}$$

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

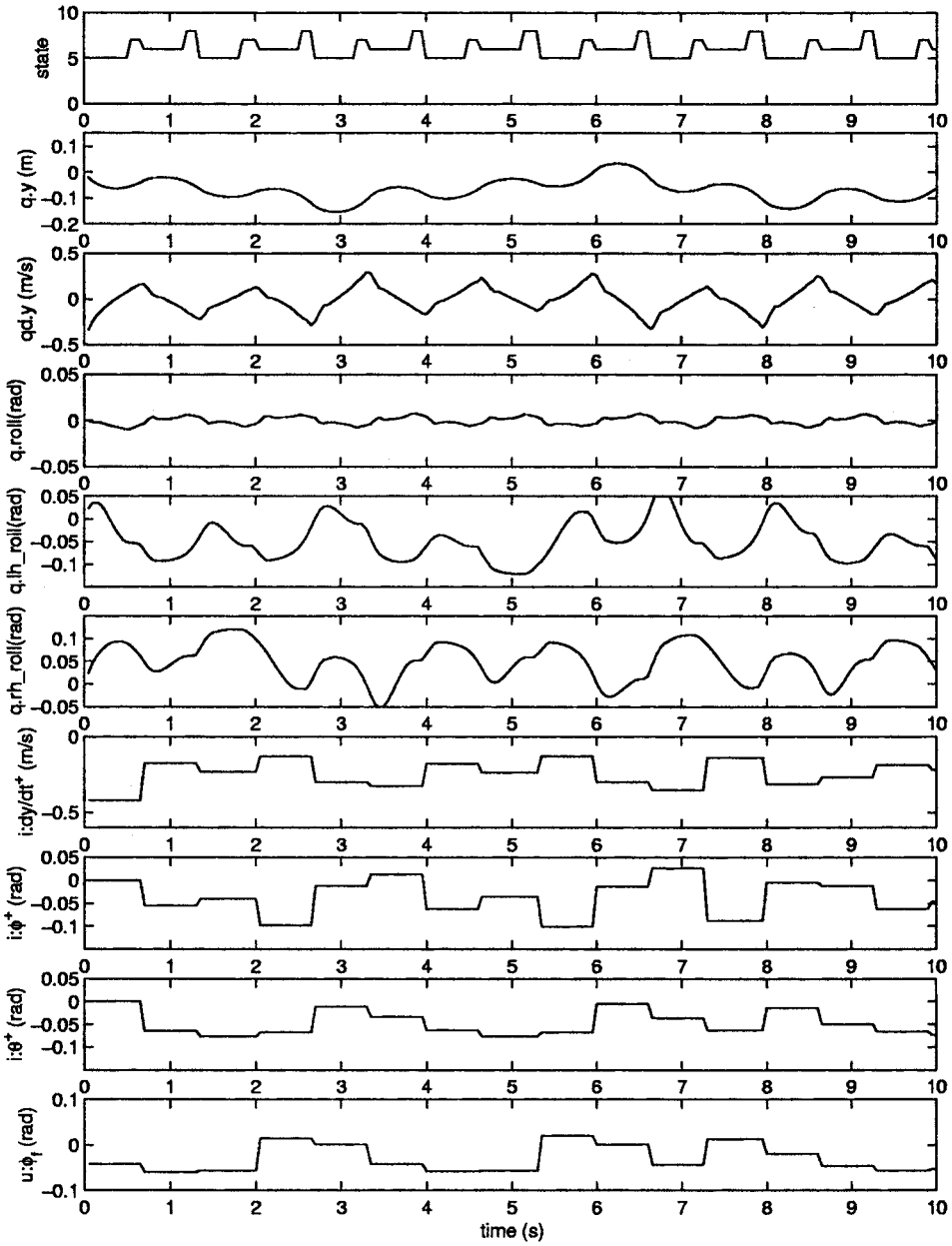

Fig. 6. The simulation data (after the simulated M2 has achieved the learning target) for the implementation introduced in Section 4. The local control was based on equation 7.

The details of this reinforcement learning implementation are not listed here because it turns out that learning is not required for such an implementation. In the sagittal plane, we adopt the same implementation as before. The combined algorithms is then applied to the simulated M2. In every iteration, the biped starts to walk from a standing posture with an initial velocity. The learning processes in both the sagittal and frontal planes are carried out simultaneously. The learning target is to achieve 100 seconds of walking without violating the failure conditions in both the sagittal and frontal planes. The desired walking speed and height are set at 0.4 m/s and 0.84 m, respectively. The swing time is fixed at 0.5 second.

5.2. Simulation results

It was discovered that the frontal plane learning is not required for the 3D walking. That is, the swing-hip roll joint control based on equation 10 and the local control law based on equation 4 for the stance-ankle roll joint could yield a stable frontal plane motion.

Three of the learning curves (corresponding to different start conditions) for the overall implementation are shown in Figure 7. The results show that the biped was able to achieve the learning target much faster than that in the previous implementations (Section 4). The reason is that the biped is now only required to learn in the sagittal plane. Thus, the learning curve is comparable to that of the planar (sagittal plane) implementation in our previous paper.[1]

Figure 8 shows the stick diagram of the dynamic walking of the simulated M2 after the learning target was achieved. The simulation data concerning the frontal plane motion (after the learning target was achieved) are shown in Figure 9. The lateral velocity and the roll angle of the body were well-behaved. The hip-roll motions were also much more

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

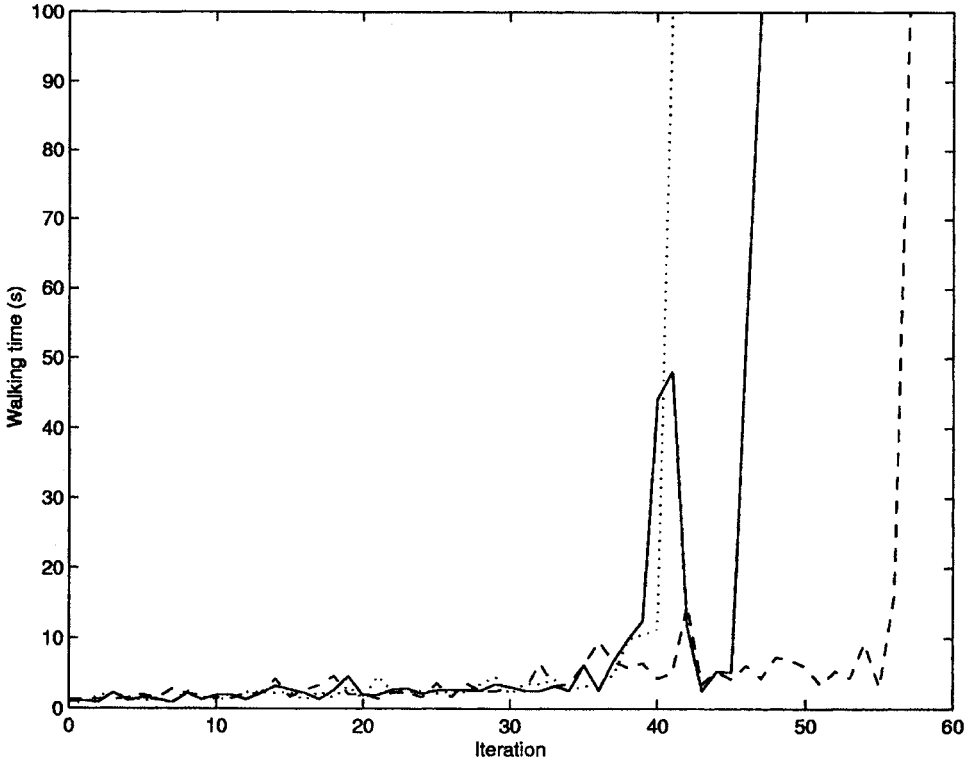

Fig. 7. Three learning curves for the dynamic walking of the simulated biped. The learning was only required in the saggital plane. The frontal plane implementation was based on the notion of symmetry.

regular than the previous implementation (see the graphs of *q.lh_roll* and *q.rh_roll*). However, there was a slight drifting of the average horizontal position y of the body. It was mainly because there was a slight asymmetry in the frontal plane motion of the biped.

There are several ways to correct the drifting in the lateral direction. One of which is to create asymmetry in the local control at the stance-ankle roll joint when the drifting is exceeded by a preset amount. Another approach is to use the transverse motion (yaw). For example, if the biped has drifted to one side, the stance-hip yaw joint can be used to change the walking direction to nullify the drift.

The feasibility of achieving the frontal plane control based on the notion of symmetry and the local control mechanism without the need for learning was a very interesting finding. In fact, the frontal plane control strategy was quite "robust" because the biped did not encounter a single failure in the frontal plane even while it was learning the sagittal plane motion.

We also observed that the local control law (equation 4) that did not work well in the previous implementation actually did a good job here. This illustrates that the choice of the control law for the stance-ankle roll joint depends on the swing leg strategy.

The following subsection analyzes this frontal plane motion control strategy using a simple model. The model is meant to provide some insight for the strategy, especially the stabilization issue.

5.3. Analysis

This subsection describes the analysis of the symmetry approach for the lateral balance. A simple linear double pendulum model as shown in Figure 10 is used in the

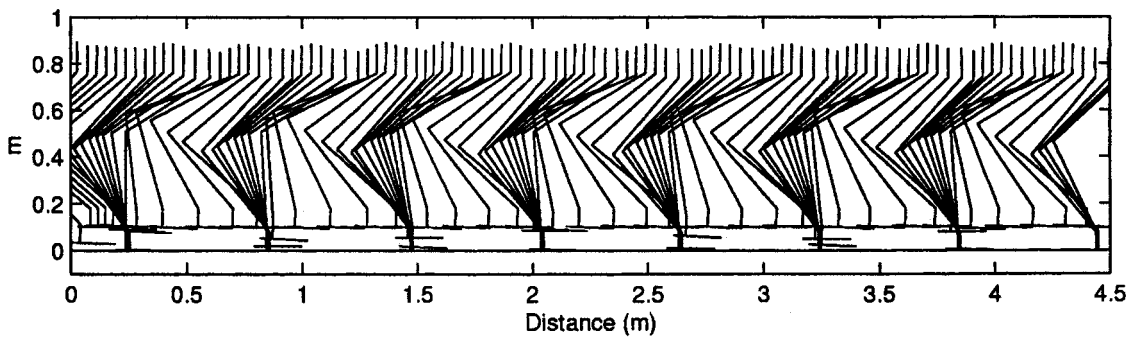

Fig. 8. Stick diagram of the dynamic walking of the simulated M2 after the learning target was achieved. Only the left leg is shown in the diagram and the images are 0.1 second apart.

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

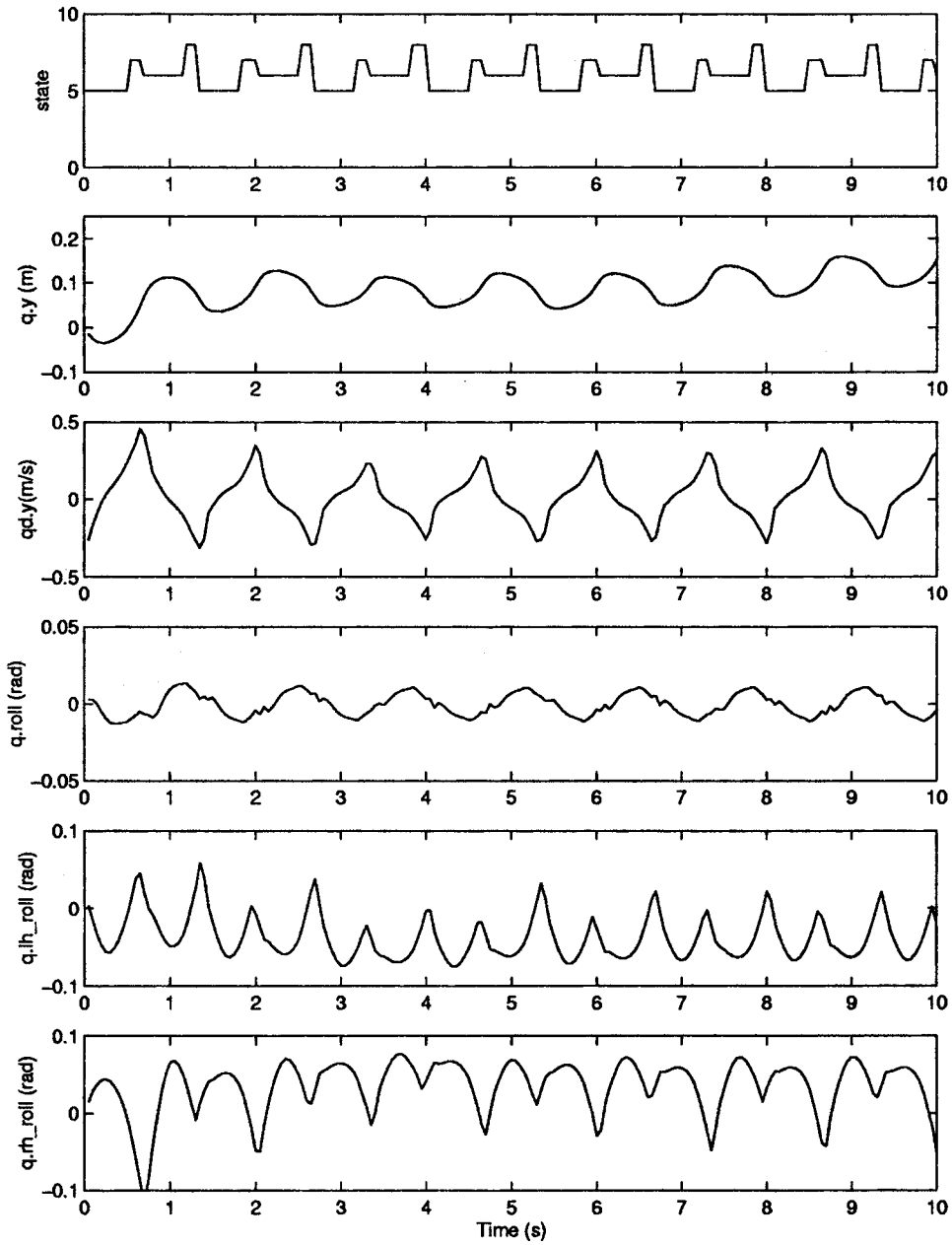

Fig. 9. The simulation data (after the simulated biped has achieved the learning target) for the implementation in which the frontal plane algorithm is based on the notion of symmetry.

analysis. It is used to validate that the symmetry approach can result in a stable motion. In this model, the body is a point mass M that is constrained to move along the horizontal constraint line 1. The swing leg's mass m is assumed to be concentrated at the foot and it is constrained to move along the horizontal constraint line 2. x_1 and x_2 are chosen to be the generalized coordinates for this model. x_1 is the horizontal distance of the body mass M measured from the stance ankle. x_2 is the horizontal distance of the leg mass m measured from the body mass. τ_1 is the torque exerted on the stance leg by the ground. τ_2 is the torque exerted on the swing leg by the stance leg. Let's assume that the biped's single support period or swing time T_s is fixed, and the support exchange occurs instantaneously.

If the support exchange event is temporarily neglected, the dynamic equations of this model can be obtained by Newtonian mechanics as follows:

$$\begin{bmatrix} (m+M)h & mh \\ mh & mh \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} -(m+M)g & 0 \\ 0 & mg \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{12}$$

Based on the symmetry approach adopted earlier, the control laws for τ_1 and τ_2 are formulated as follows:

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

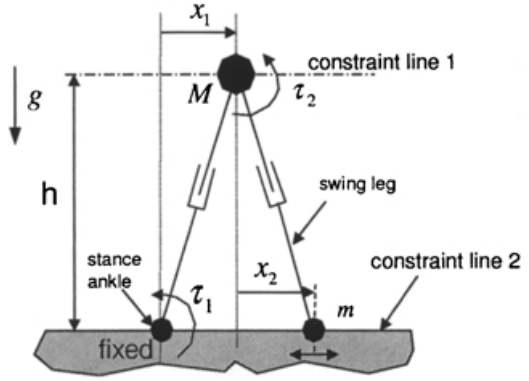

Fig. 10. A linear double pendulum model for the analysis of the symmetry approach used in the frontal plane algorithm.

$$\tau_1 = K_1 \dot{x}_1 \tag{13}$$

$$\tau_2 = K_P(x_1 - x_2) + K_D(\dot{x}_1 - \dot{x}_2) \tag{14}$$

Now, let's substitute the control laws for τ_1 and τ_2 into equation 12 to yield equation 15:

$$\mathbf{A}\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{bmatrix} + \mathbf{B}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} + \mathbf{C}\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix} \tag{15}$$

where

$$\mathbf{A} = \begin{bmatrix}(m+M)h & mh\\ mh & mh\end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix}-K_D + K_1 & K_D\\ -K_D & K_D\end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix}-(m+M)g - K_P & K_P\\ -K_P & mg + K_P\end{bmatrix}$$

Equation 15 can be rearranged as follows:

$$\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{bmatrix} = \mathbf{D}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} + \mathbf{E}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{16}$$

where

$$\mathbf{D} = -\mathbf{A}^{-1}\mathbf{B}$$

$$\mathbf{E} = -\mathbf{A}^{-1}\mathbf{C}$$

Finally, the above equations can be expressed in the following state space form:

$$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\\ \dot{y}_3\\ \dot{y}_4\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ E(1,1) & D(1,1) & E(1,2) & D(1,2)\\ 0 & 0 & 0 & 1\\ E(2,1) & D(2,1) & E(2,2) & D(2,2)\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\\ y_4\end{bmatrix} \tag{17}$$

where $\vec{y} = [y_1\ y_2\ y_3\ y_4]^T = [x_1\ \dot{x}_1\ x_2\ \dot{x}_2]^T$; $D(i,j)$ and $E(i,j)$ denote the elements at row i and column j of matrix **D** and **E**, respectively.

Let $\vec{y}_i^{(n)}$ denote the initial state vector (subscript i) at the beginning of the nth step. If the swing time T_s is fixed, the final state vector (subscript f) just before the support exchange is given by the explicit solution of equation 17 as follows:

$$\vec{y}_f^{(n)} = \exp(\mathbf{F}T_s)\vec{y}_i^{(n)} \tag{18}$$

where

$$\mathbf{F} = \begin{bmatrix}0 & 1 & 0 & 0\\ E(1,1) & D(1,1) & E(1,2) & D(1,2)\\ 0 & 0 & 0 & 1\\ E(2,1) & D(2,1) & E(2,2) & D(2,2)\end{bmatrix}$$

Now, let's consider the support exchange event. Let's assume that the support exchange is instantaneous and the previous stance foot is not acted on by any impulsive force. Also, by the conservation of angular momentum about the ankle of the new supporting leg and assuming that both masses remain in their respective constraint lines, the following boundary conditions can be obtained:

$$\vec{y}_i^{(n+1)} = \mathbf{G}\vec{y}_f^{(n)} \tag{19}$$

where

$$\mathbf{G} = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & -1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}$$

By substituting equation 19 into equation 18, the following discrete linear time-invariant system is obtained:

$$\vec{y}_i^{(n+1)} = \mathbf{H}\vec{y}_i^{(n)} \tag{20}$$

where $\mathbf{H} = \mathbf{G}\exp(\mathbf{F}T_s)$ is the state transition matrix.

From linear control theory, the modulus of all the eigenvalues λ_i of **H** must be less than unity for equation 20 to be asymptotically stable. Now, let's assign values to the parameters of this model so that quantitative analysis can be carried out. The key parameters are set as follows: $M = 20$ kg, $m = 2$ kg, $h = 0.84$ m, $T_s = 0.5$ s, and $g = 9.81$ m/s^2. The maximum modulus of the eigenvalues is plotted against K_P and K_D for $K_1 = 1$,10, 20 and 50 using *Matlab*™ as shown in Figure 11. The figure shows that the system indeed has a zone in the K_P-K_D plane where the maximum modulus of the eigenvalues is less than one. The zone is especially obvious and large for $K_1 = 10$. From the figure, it is also observed that K_P and K_D should not be too large or too small. Having large values for these parameters means better tracking of the desired trajectory. That is, perfect symmetry is not desirable for the algorithm. On the other hand, if K_P and K_D are too small, this will result in a poor tracking of the desired trajectory. That is, the stance-leg roll angle has little influence on the swing-leg roll angle.

This analysis validates the frontal plane algorithm presented in this section. We conclude that a stable motion in the frontal plane can be achieved if a proper set of K_P, K_D and K_1 is chosen.

6. DISCUSSION

In this paper, the 3D walking algorithms were constructed based on the assumption that the motions in the sagittal and

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

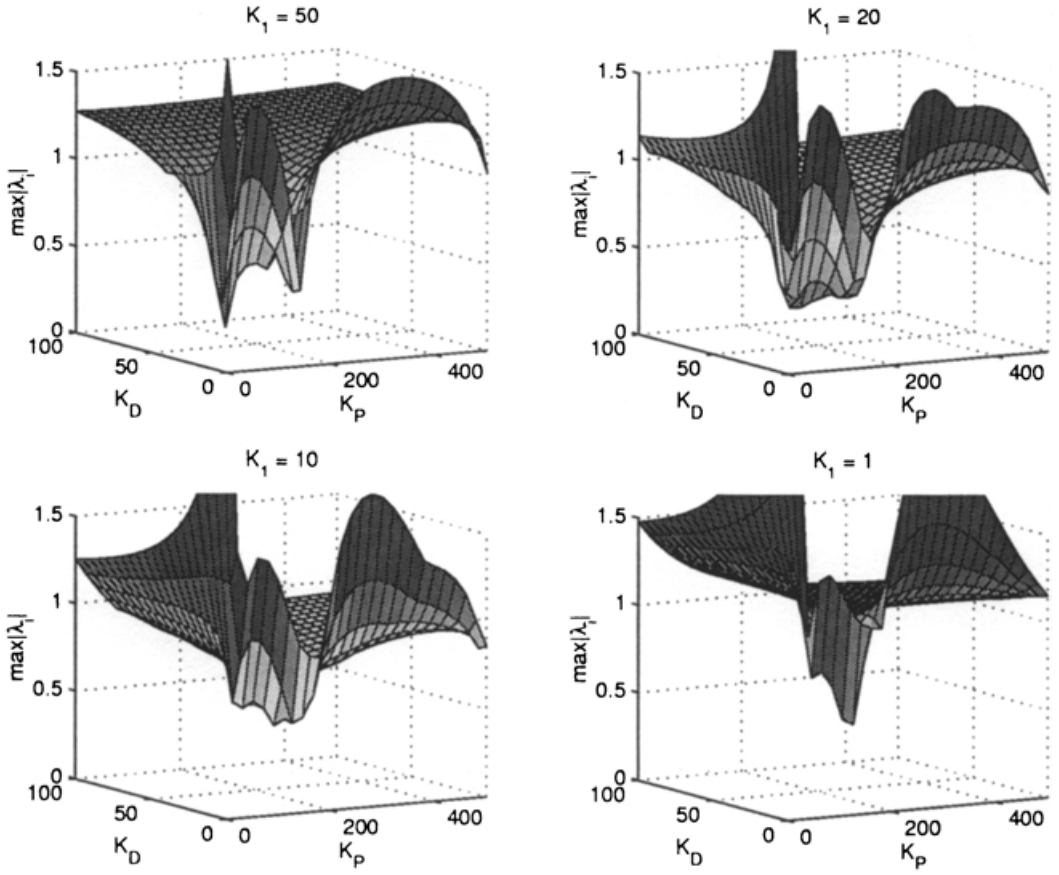

Fig. 11. Four surface plots of the maximum modulus of the eigenvalues of **H** against K_P and K_D corresponding to $K_1 = 1$, 10, 20 and 50.

frontal planes could be independently considered. The successful implementations of the algorithms indirectly validated this assumption and the proposed control architecture presented in our previous paper.[1]

In Section 4, the reinforcement learning algorithm was used to learn the end position of the swing-leg roll angle ϕ_f before the beginning of each single support phase. Two local control law candidates for the stance-ankle roll joint were compared. The simulation results demonstrated that the choice of the local cqntrol law for the stance-ankle roll joint could affect the learning rate.

Section 5 illustrated how one could tap the hidden potential of the notion of symmetry for the bipedal walking task. The successful application of the notion for the frontal plane motion control without complex control laws endorses Raibert's claim[11] that symmetry could be utilized to simplify the control of legged robots.

The simulation results in both Sections 4 and 5 also revealed that the choice of the control law for the stance ankle depends on the selected swing leg strategy. The control law that did not work well for the frontal plane implementation in Section 4 actually works well for the implementation in Section 5.

7. CONCLUSIONS

This paper presented two different frontal-plane motion control approaches for the simulated biped. In the first approach, the reinforcement learning algorithm was used to learn the desired swing-leg roll angle given a set of state variables. With an appropriate local control at the stance-ankle roll joint, the algorithm was able to perform reasonably well.

This paper also presented another approach for the frontal plane motion control. It was based on the notion of symmetry. The swing leg was commanded to mirror the behaviour of the stance leg in the frontal plane. With an appropriate local control at the stance-ankle roll joint, the algorithm was successfully implemented to achieve lateral balance without the need of learning. This approach was validated by using a simplified dynamic model.

The simulation results also validated that the 3D walking task could indeed be handled by breaking it down into three 2D motion control sub-tasks corresponding to each of the three orthogonal planes. This approach is very attractive because it significantly reduces the complexity of the overall algorithm.

References

1. Chee-Meng Chew and Gill A. Pratt, "Dynamic bipedal walking assisted by learning", *Robotica* **20**, Part 5, 477–491 (2002).
2. R.E. Goddard Jr., H. Hemami and F.C. Weimer, "Biped side step in the frontal plane", *IEEE Transactions on Automatic Control* **AC-28**, No. 2, 179–186 (Feb., 1983).
3. H. Hemami and B.F. Wyman, "Modeling and control of constrained dynamic systems with application to biped locomotion in the frontal plane", *IEEE Transactions on Automatic Control* **AC-24**, No. 4, 526–535 (Aug, 1979).

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press

4. K. Iqbal, H. Hemami and S. Simon, "Stability and control of a frontal four-link biped system", *IEEE Transactions on Biomedical Engineering* **BME-40**, No. 10, 1007–1017 (Oct, 1993).
5. V.T. Inman, H.J. Ralson and F. Todd, *Human Walking* (Williams and Wilkins, Baltimore, London, 1981).
6. Thomas A. McMahon, *Muscles, Reflexes, and Locomotion* (Princeton University Press, Princeton, New Jersey, 1984).
7. Paul W. Latham, "A Simulation Study of Bipedal Walking Robots: Modeling, Walking Algorithms, and Neural Network Control", *PhD dissertation* (University of New Hampshire, Sept. 1992).
8. S. Kajita and K.Tani, "Experimental study of biped dynamic walking in the linear inverted pendulum mode", *IEEE International Conference on Robotics and Automation* (1995) pp. 2885–2891.
9. Shuuji Kajita, Kazuo Tani and Akira Kobayashi, "Dynamic walk control of a biped robot along the potential energy conserving orbit", *IEEE International Conference on Intelligent Robots and Systems* (1990) pp. 789–794.
10. R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, l998).
11. Marc. H. Raibert, *Legged Robots That Balance* (MIT Press, Cambridge, MA., 1986).

https://doi.org/10.1017/S0263574703005253 Published online by Cambridge University Press