Virtual Model Control: An Intuitive Approach for Bipedal Locomotion

Jerry Pratt
Chee-Meng Chew
Ann Torres
Peter Dilworth
Gill Pratt

Leg Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA
http://www.ai.mit.edu/projects/leglab/

Abstract

Virtual model control is a motion control framework that uses virtual components to create virtual forces generated when the virtual components interact with a robot system. An algorithm derived based on the virtual model control framework is applied to a physical planar bipedal robot. It uses a simple set of virtual components that allows the robot to walk successfully over level terrain. This paper also describes how the algorithm can be augmented for rough terrain walking based on geometric consideration. The resulting algorithm is very simple and does not require the biped to have an extensive sensory system. The robot does not know the slope gradients and transition locations in advance. The ground is detected using foot contact switches. Using the algorithm, we have successfully compelled a simulated seven-link planar biped to walk blindly up and down slopes and over rolling terrain.

KEY WORDS—biped, legged locomotion, virtual model control, impedance control

1. Introduction

Dynamic bipedal robots are extremely difficult to control because they are nonlinear; interact with a semistructured, complex environment; are nominally unstable; are multi-input, multi-output (MIMO); and exhibit time variant and "nonsmooth" dynamics (during support exchange).

In addition, the performance measures of such robots are much different from typical notions of performance such as command following and disturbance rejection. Performance for these robots is usually defined in terms of efficiency, locomotion smoothness, maximum speed, and robustness to rough terrain.

The International Journal of Robotics Research
Vol. 20, No. 2, February 2001, pp. 129-143,
©2001 Sage Publications, Inc.

In the bipedal walking research, most researchers usually start by specifying the reference trajectories for the bipedal robots. The trajectories are usually prescribed in joint space. These trajectories are usually obtained by observing the walking patterns of humans (Vukobratovic et al. 1990; Furusho and Sano 1990) or by some forms of pattern generator formulation (Bay and Hemami 1987; Katoh and Mori 1984).

To realize the reference trajectories, a linearization approach is commonly adopted to reduce the nonlinear dynamic equations into a linear one (Gubina, Hemami, and McGhee 1974; Golliday and Hemami 1977; Miura and Shimoyama 1984; Mita et al. 1984). This facilitates the application of linear multivariable control methods. However, due to underactuation in the bipedal system, error in the parameters of the dynamic model, and significant deviation from the equilibrium states (about which the system is linearized), one usually needs to modify the reference trajectories and the model parameters iteratively before a successful implementation can be achieved. The problem is that most of these parameters do not possess intuitive behaviors. Thus, it is difficult to tune them.

In fact, most of the approaches in bipedal walking require some forms of iterative tuning processes in the control architecture. The question is how one could formulate a solution for the bipedal walking task so that the parameter tuning is minimally needed and/or can be easily carried out (e.g., one that possesses an intuitive tuning rule).

In this paper, we propose an intuitive control scheme called virtual model control for legged locomotion. In this framework, virtual components that have physical counterparts, for example, mechanical spring, damper, and so on, are placed at strategic locations within the robot or between the robot and the environment. Physical intuition is needed to place these virtual components. Once the placement is done, the interactions between these components and the robots automatically generate the desired torques or forces at the actuators. No

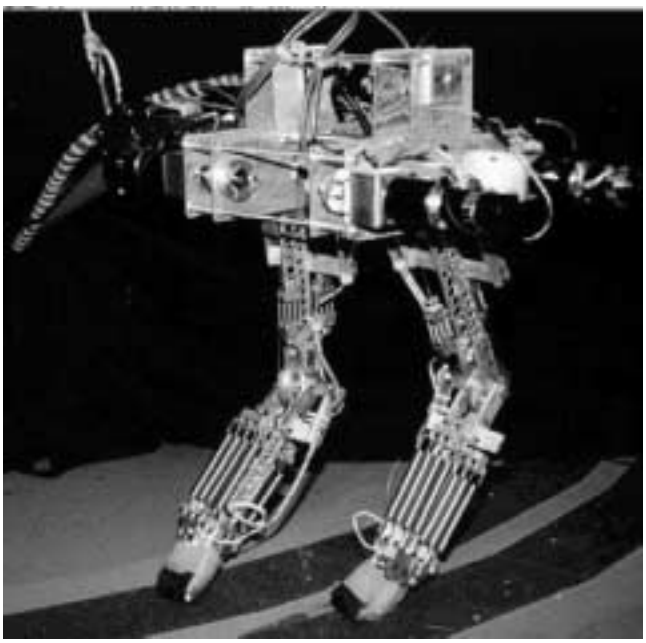

Fig. 1. Photo of Spring Turkey. There are four actuators attached to the body. Power is transmitted to the hips and knees via cables. The unactuated feet consist of a U-shaped strip of rubber. A boom is used to prevent motion in the lateral, roll, and yaw directions. Note the spring packs used to implement series elastic actuation.

dynamic model of the robot is necessary in the control algorithm. This approach is applied to two planar bipeds (constrained to move in the sagittal plane). By selecting a proper set of gait parameters, we demonstrate that a stable dynamic walking can be achieved for the bipeds.

Section 2 describes the two planar bipedal walking robots to which the control algorithm is applied. Section 3 provides an overview of virtual model control. Section 4 presents the mathematics required to transform the virtual forces (generated by the virtual components) into the desired joint torques for both single-support and double-support phases. Section 5 describes the control algorithm and strategy for level ground walking of the biped. Section 6 describes how the level ground walking algorithm can be extended for rough terrain walking.

2. Bipedal Walking Robots

The control strategies presented in this paper are designated for two bipedal walking robots. They are Spring Turkey (see Fig. 1) and Spring Flamingo (see Fig. 2). Spring Turkey is a five-link planar biped that has 4 actuated degrees of freedom. It was designed and built by Peter Dilworth and Jerry Pratt in 1994. Spring Flamingo is a seven-link planar biped that has 6 actuated degrees of freedom. It was designed and built by Jerry Pratt in 1996.

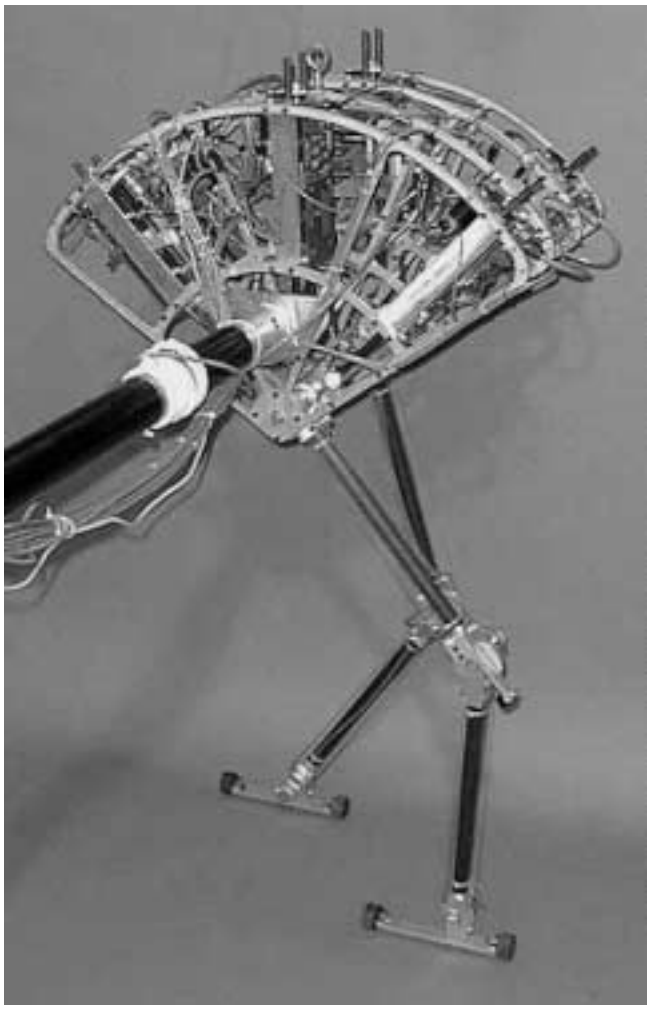

Fig. 2. Photo of Spring Flamingo. Its structure and mechanism are very similar to spring turkey except that the series elastic actuators are contained in the body, it has feet, and the legs are less massive compared to the body's mass.

Both bipeds have an actuated hip and knee on each leg. Spring Flamingo also has an actuated ankle, although it is assumed to be limp in this paper. Unactuated booms are used to constrain the biped's roll, yaw, and lateral motion, thereby reducing it to a planar robot. Potentiometers at the hips, knees, and boom measure joint angles and body pitch.

Series elastic actuation (Pratt and Williamson 1995) is employed at each actuated degree of freedom, allowing for accurate application of torques and a high degree of shock tolerance. In series elastic actuation, spring elements are inserted between the gear transmission and the external load. By controlling the deflection of the springs, net force acting on the load can be indirectly controlled.

Spring Turkey weighs approximately 10 kg and stands 60 cm tall from toe to hip. Spring Flamingo weighs approximately 12 kg and stands 84 cm tall from toe to hip.

3. Virtual Model Control

Virtual model control is a motion control framework that uses simulations of virtual components to generate desired joint torques. These joint torques create the same effect that the virtual components would have created, had they existed, thereby creating the illusion that the simulated components are connected to the real robot. Such components can include simple springs, dampers, dashpots, masses, latches, bearings, nonlinear potential and dissipative fields, or any other imaginable component. Virtual components can even contain adaptive and learning elements (Pratt 1994; Chew and Pratt 1999). Virtual model control borrows ideas from virtual reality, hybrid position-force control (Raibert and Craig 1981), stiffness control (Salisbury 1980), impedance control (Hogan 1985), and the operational space formulation (Khatib 1986).

Many complex tasks that are difficult to describe using traditional techniques can be readily characterized with a simple set of virtual components. For example, consider a robot wishing to impart an impact onto an unknown surface (e.g., knocking on a door). Ordinarily, this would be a very difficult task to specify. However, with virtual model control, we merely need to attach a virtual mass with a given kinetic energy to the robot's hand via a virtual spring and damper. The robot's hand will now move to strike out and, after imparting the desired impact to the environment, bounce back due to mass resonating with the virtual spring-damper.

Some benefits of virtual model control are that it is compact, requires relatively small amounts of computation, and can be implemented in a distributed manner (for information on how to implement virtual components, see Pratt 1995; Pratt et al. 1996). Furthermore, a high-level controller could be implemented as a state machine that simply changes virtual component connections or parameters at the state transitions. Even though a discrete high-level controller is used, the overall motion can be smooth if the virtual components have a low-pass filter effect.

Note that virtual model control does not use inverse dynamics to alter the behavior of the robot. We believe that the inverse dynamics approach should only be used when high-performance requirements or other extreme situations dictate. This is because plant inversion adds computational complexity, and fighting the natural dynamics of the robot can be inefficient.

Also note that with virtual model control, we usually talk in terms of spring set points, for example, and not *commanded* positions. Except for actuator and computation nonidealities, we can perfectly implement virtual components, whereas very few control algorithms can perfectly track a commanded trajectory. In this light, we believe that robots cannot be *commanded* to perform a task; they can only be given *hints* and *suggestions*.

Virtual model control has been used to control dynamic walking bipedal robots (described in the next section) and an agile 3D hexapod in simulation (Torres 1996).

4. Virtual Model Implementation for Bipeds

In this section, we present the mathematics to transform virtual forces into desired joint torques for the support leg in single support or both legs in double support. We follow the procedure described in Pratt et al. (1996).

4.1. Single-Leg Implementation

Figure 3 shows a simple 2D, four-link, three-joint serial robot model that we use to represent a single leg of our walking robots. We wish to connect a virtual component between frame $\{A\}$, which is attached to the foot, and frame $\{B\}$, which is attached to the body. The angles θ_a, θ_k, and θ_h are those of the ankle, knee, and hip, respectively. The lower link (tibia) is of length L_1, whereas the upper link (femur) is of length L_2. In this example, we assume that the foot is flat on the ground so that ${}^{O}_{A}R = I$ (${}^{O}_{A}R$ is the rotation matrix that describes $\{A\}$ relative to a reference frame $\{O\}$).

The forward kinematic map from frame $\{A\}$ to frame $\{B\}$ of this example is as follows:

$$ {}^{A}_{B}\vec{X} = \begin{bmatrix} x \\ z \\ \theta \end{bmatrix} = \begin{bmatrix} -L_1 s_a - L_2 s_{a+k} \\ L_1 c_a + L_2 c_{a+k} \\ -\theta_h - \theta_k - \theta_a \end{bmatrix}, \qquad (1) $$

where s_a, s_{a+k}, c_a, and c_{a+k} denote $\sin(\theta_a)$, $\sin(\theta_{a+k})$, $\cos(\theta_a)$, and $\cos(\theta_a + \theta_k)$, respectively.

Partial differentiation produces the Jacobian,

$$ {}^{A}_{B}J = \begin{bmatrix} -L_1 c_a - L_2 c_{a+k} & -L_2 c_{a+k} & 0 \\ -L_1 s_a - L_2 s_{a+k} & -L_2 s_{a+k} & 0 \\ -1 & -1 & -1 \end{bmatrix}. \qquad (2) $$

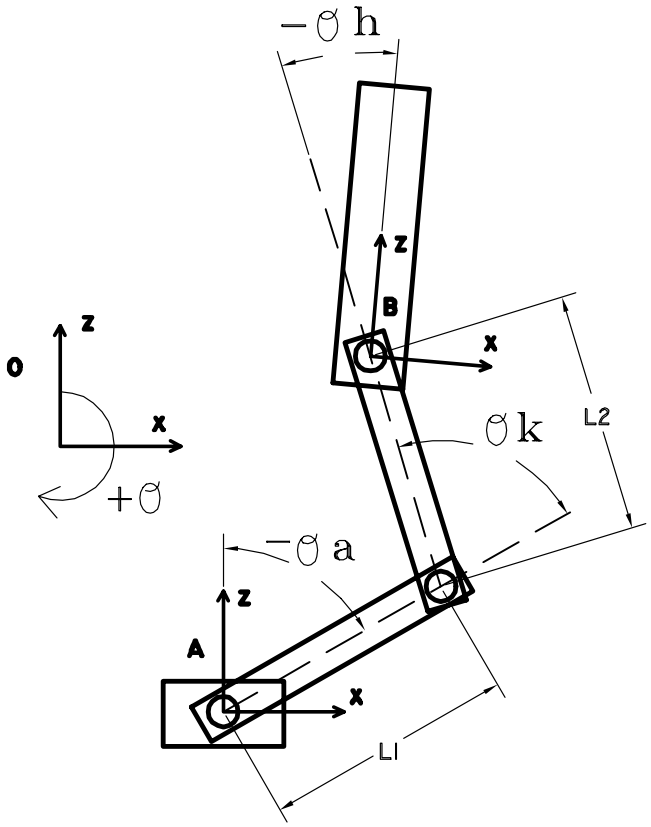

Fig. 3. Single-leg implementation. Reaction frame $\{A\}$ is assumed to be in the same orientation as reference frame $\{O\}$ so that ${}^{O}_{A}R = I$.

The Jacobian relates the virtual velocity ${}^{A}_{B}\dot{\vec{X}}$ between frames A and B with the joint velocities $\dot{\vec{\Theta}} = [\theta_a\ \theta_k\ \theta_h]^T$

$$ {}^{A}_{B}\dot{\vec{X}} = {}^{A}_{B}J\ \dot{\vec{\Theta}} \tag{3} $$

and the virtual force $\vec{F} = [f_x\ f_z\ f_\theta]^T$ to joint torque $\vec{\tau} = [\tau_a\ \tau_k\ \tau_h]^T$

$$ \vec{\tau} = ({}^{A}_{B}J)^T\ ({}^{A}_{B}\vec{F}). \tag{4} $$

The Jacobian is of full rank, indicating that all virtual force directions are admissible. Because the stance ankle is assumed to be limp, we add the constraint of an unactuated ankle, $\tau_a = 0$. This will constrain the direction in which virtual forces can be applied. With a limp ankle, eq. (4) is constrained,

$$ \begin{bmatrix} 0 \\ \tau_k \\ \tau_h \end{bmatrix} = \begin{bmatrix} -L_1 c_a - L_2 c_{a+k} & -L_1 s_a - L_2 s_{a+k} & -1 \\ -L_2 c_{a+k} & -L_2 s_{a+k} & -1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_x \\ f_z \\ f_\theta \end{bmatrix}. \tag{5} $$

For our walking robot, we are more concerned about applying forces in the vertical direction and torques about the body then we are concerned about applying horizontal forces. Therefore, we specify f_z and f_θ and solve for f_x:

$$ f_x = \frac{-1}{L_1 c_a + L_2 c_{a+k}} \begin{bmatrix} L_1 s_a + L_2 s_{a+k} & 1 \end{bmatrix} \begin{bmatrix} f_z \\ f_\theta \end{bmatrix}. \tag{6} $$

Substituting this equation into eq. (5), we get

$$ \begin{bmatrix} \tau_k \\ \tau_h \end{bmatrix} = \begin{bmatrix} \frac{-L_1 L_2 s_k}{L_1 c_a + L_2 c_{a+k}} & \frac{-L_1 c_a}{L_1 c_a + L_2 c_{a+k}} \\ 0 & -1 \end{bmatrix} \begin{bmatrix} f_z \\ f_\theta \end{bmatrix}. \tag{7} $$

We now have a simple set of equations for determining joint torques given virtual forces. These equations will be used in the subsequent sections in the control of the bipedal walking robots during the single-support phase. Note that the matrix in eq. (7) is of full rank for all values of $\vec{\Theta}$ except for $\theta_k = 0$. This corresponds to a fully extended knee, for which the virtual forces in the z-direction can be arbitrary. If the knee is not fully extended, the virtual forces are a unique function of the joint torques at the hip and knee.

4.2. Dual-Leg Implementation

The previous example discussed a serial robot model corresponding to the robot body being supported by only one leg. Here, we examine a parallel mechanism representing the robot body being supported by both legs (see Fig. 4). That is, this model consists of the previous single-leg example plus another leg. We wish to connect a multiframe virtual component between the reaction frames $\{A_l\}$ and $\{A_r\}$, which are connected to the feet, and the action frame $\{B\}$, which is connected to the body. The individual leg parameters and joint angles are identical to those of the single-leg example, with the l subscript denoting the left leg and the r subscript denoting the right leg. Again, we assume that the feet are flat on the ground so that ${}^{O}_{A_l}R = {}^{O}_{A_r}R = I$.

We computed the Jacobian for each serial chain of this parallel mechanism in the previous example. We now combine them in the following manner:

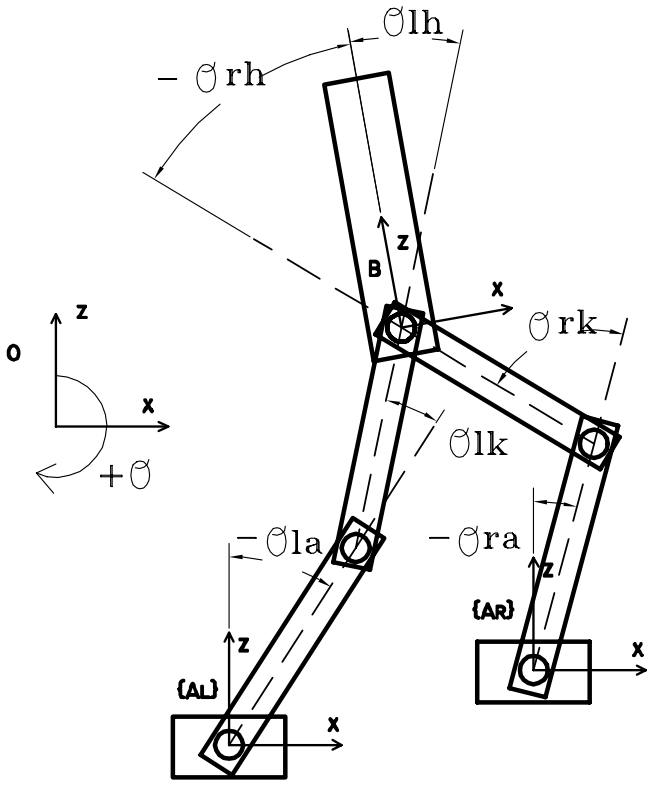

Fig. 4. Dual-leg example. Reaction frames $\{A_l\}$ and $\{A_r\}$ are assumed to be in the same orientation as reference frame $\{O\}$ so that ${}^{O}_{A_l}R = {}^{O}_{A_r}R = I$.

$$\begin{bmatrix} \vec{\tau}_l \\ \vec{\tau}_r \end{bmatrix} = \begin{bmatrix} {}^{A_l}_{B}J^T & 0 \\ 0 & {}^{A_r}_{B}J^T \end{bmatrix} \begin{bmatrix} \vec{F}_l \\ \vec{F}_r \end{bmatrix}. \tag{8}$$

This expands to

$$\begin{bmatrix} \tau_{la} \\ \tau_{lk} \\ \tau_{lh} \\ \tau_{ra} \\ \tau_{rk} \\ \tau_{rh} \end{bmatrix} = \begin{bmatrix} A & B & -1 & 0 & 0 & 0 \\ Q & R & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C & D & -1 \\ 0 & 0 & 0 & S & T & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_{xl} \\ f_{zl} \\ f_{\theta l} \\ f_{xr} \\ f_{zr} \\ f_{\theta r} \end{bmatrix}, \tag{9}$$

where

$$A = -L_1\cos(\theta_{la}) - L_2\cos(\theta_{la}+\theta_{lk})$$

$$B = -L_1\sin(\theta_{la}) - L_2\sin(\theta_{la}+\theta_{lk})$$

$$C = -L_1\cos(\theta_{ra}) - L_2\cos(\theta_{ra}+\theta_{rk})$$

$$D = -L_1\sin(\theta_{ra}) - L_2\sin(\theta_{ra}+\theta_{rk})$$

$$Q = -L_2\cos(\theta_{la}+\theta_{lk}), \quad R = -L_2\sin(\theta_{la}+\theta_{lk})$$

$$S = -L_2\cos(\theta_{ra}+\theta_{rk}), \quad T = -L_2\sin(\theta_{ra}+\theta_{rk}).$$

Equation (9) maps the virtual forces for each leg to the required joint torques. Because the action frame $\{B\}$ is coincidental, we have the compatibility relation that the force vector must equal the vector sum of the forces produced by each serial chain,

$$\begin{bmatrix} f_x \\ f_z \\ f_\theta \end{bmatrix} = \begin{bmatrix} f_{xl} \\ f_{zl} \\ f_{\theta l} \end{bmatrix} + \begin{bmatrix} f_{xr} \\ f_{zr} \\ f_{\theta r} \end{bmatrix}. \tag{10}$$

In fact, we are interested in generating the three virtual forces, f_x, f_z, and f_θ, which are effective forces acting on the robot body due to the joint torques. Because we have six joints and wish to control these three forces, we require three constraints. Unactuated ankles provide two constraints,

$$\tau_{la} = 0, \quad \tau_{ra} = 0. \tag{11}$$

The third constraint provides us with a design degree of freedom. We could choose it to maximize a performance criterion, for example. Here, we simply choose to match the hip torques,

$$\tau_{lh} = \tau_{rh} \Longrightarrow f_{\theta l} = f_{\theta r}. \tag{12}$$

Putting the above constraints in vector form, we have

$$\begin{bmatrix} f_x \\ f_z \\ f_\theta \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ A & B & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C & D & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_{xl} \\ f_{zl} \\ f_{\theta l} \\ f_{xr} \\ f_{zr} \\ f_{\theta r} \end{bmatrix}. \tag{13}$$

We must now perform a 6×6 matrix inversion to solve for the individual leg forces. We drop the terms that are multiplied by zero. This results in a 6×3 matrix relating the single vector of virtual forces to the individual leg virtual forces. This matrix is then substituted into eq. (9) and simplified, resulting in the virtual force to joint torque relation

$$\begin{bmatrix} \tau_{lk} \\ \tau_{lh} \\ \tau_{rk} \\ \tau_{rh} \end{bmatrix} = \begin{bmatrix} \frac{C\,V}{E} & \frac{D\,V}{E} & \frac{-V-Q\,D+R\,C}{2\,E} - \frac{1}{2} \\ 0 & 0 & -1/2 \\ \frac{-A\,W}{E} & \frac{-B\,W}{E} & \frac{W+S\,B-T\,A}{2\,E} - \frac{1}{2} \\ 0 & 0 & -1/2 \end{bmatrix} \begin{bmatrix} f_x \\ f_z \\ f_\theta \end{bmatrix}, \tag{14}$$

where

$$E = CB - AD$$

$$V = QB - RA = -L_1 L_2 \sin(\theta_{lk})$$

$$W = SD - TC = -L_1 L_2 \sin(\theta_{rk}).$$

Once again, we have a simple set of equations for relating virtual forces to joint torques. These equations will be used in the subsequent sections in the control of the bipedal walking robots during the double-support phase. Intuitively, the matrix in eq. (14) should be of full rank for all joint angles except when the two feet are colinear. In all other configurations, all virtual forces are admissible.

5. Level-Ground Walking

Virtual model control is applied to Spring Turkey to allow it to perform simple walking on level ground. The algorithm (called turkey walking) is summarized as follows:

- Attempt to maintain a constant height and pitch (in both the double-support and single-support phases).
- Attempt to correct for velocity disturbances (in the double-support phase).
- Attempt to swing the nonstance leg so that the foot is placed a nominal stride length away from the support foot when transitioning to double support (in the single-support phase).

The following intuitive rules are used for the state transitions between the single-support phase and the double-support phase:

- Transition from double support to single support if the body's x position becomes close to the point where a foot contacts the ground.
- Transition from single support to double support if the body's x position becomes far away from the support foot's ground contact point.

To implement turkey walking, we use simple sets of virtual components and a state machine. The state machine (see Fig. 5) is used to determine the state of the system. Table 1 lists the trigger and branch events and the virtual components that are used in each state. During both the double-support phase and the single-support phase, a virtual granny walker with spring-damper mechanisms (see Fig. 6) maintains a constant height and regulates the pitch angle to zero.

During the double-support phase, a virtual dogtrack bunny with a damper mechanism (see Fig. 7) applies a virtual force in the forward horizontal (x) direction to help maintain a desired velocity. Unlike many speed control algorithms, which operate by modulating foot placement, we chose to leave foot placement a free variable (so that the robot could choose to avoid stepping in certain areas). Instead, we maintain the walking speed by use of the virtual dogtrack bunny.

The swing leg is controlled via a virtual linkage with springs and dampers that compel the swing leg to mirror the stance leg while clearing the ground and to set down at the nominal stride length before transitioning back to the double-support phase. States Left Support 2 and Right Support 2 are used as buffer states between the single- and double-support phases. Because Spring Turkey has no foot switches to detect ground contact, in these states the swing leg is simply made limp (zero torque applied to the joints) for a set delay time, allowing for the swing leg to fall to the ground before the large forces, which the double-support phase requires, are applied.

The various virtual spring, damper, and force variables and walking parameters were chosen using physical insight and a manual search. The virtual granny walker spring-damper constants were experimentally varied while physically examining their effects (resistance to being pushed on, decay rate, etc.) until the desired effects were achieved; the walking parameters and virtual dogtrack bunny damper were changed through trial and error until the robot successfully walked. These walking parameters consisted of nominal stride length and percentage of stride length spent in the single support.

Walking was initiated in the single-support phase. A slight push was applied to the robot to propel it forward. After the push, no external intervention was required.

Figure 9 shows experimental data from Spring Turkey while performing turkey walking. The upper left graphs show the body's horizontal position (x), vertical position (z), and pitch (θ), and the corresponding spring set points (dotted). The upper right graphs show the virtual forces applied to the body due to the virtual components. The horizontal velocity, along with the virtual dogtrack bunny velocity (dotted), is plotted in the lower left-hand graph. The state of the state machine is plotted in the lower right-hand graph.

The data in Figure 9 are plotted in graphical form in Figure 8. The snapshots in Figure 8 are approximately 0.5 s apart. Lines are drawn to show the path of the tips of the feet and the center of the body.

Spring Turkey walked continuously at approximately 0.5 m/s (1.125 mph). The data show approximately six steps in 4 s, for a step time of about 0.5 s. It deviated a maximum of 3 cm from the nominal height of 54 cm, and pitch was confined to ± 0.10 radians (± 5.2 degrees).

6. Sloped-Terrain Implementation

This section describes how the level-ground walking algorithm described earlier can be extended for the biped to

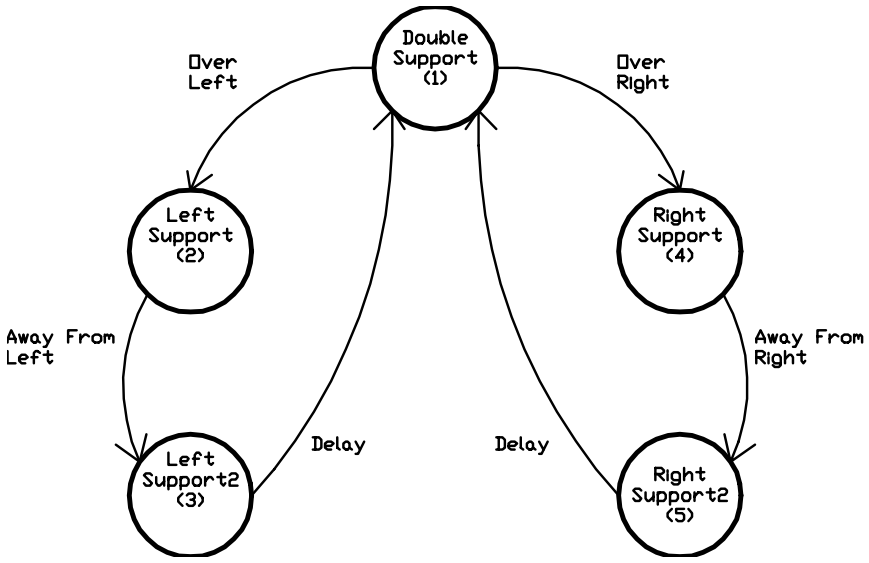

Fig. 5. State machine used in the turkey-walking algorithm.

Table 1. Details of Turkey-Walking State Machine and the Corresponding Virtual Components

State	Trigger Event	Virtual Components
1 Double Support	Delay after left or right support2	Granny walker Dogtrack bunny
2 Left Support	Body nearly over left foot	Granny walker Swing leg linkage
3 Left Support 2	Body away from left foot	Granny walker
4 Right Support	Body nearly over right foot	Granny walker Swing leg linkage
5 Right Support 2	Body away from right foot	Granny walker

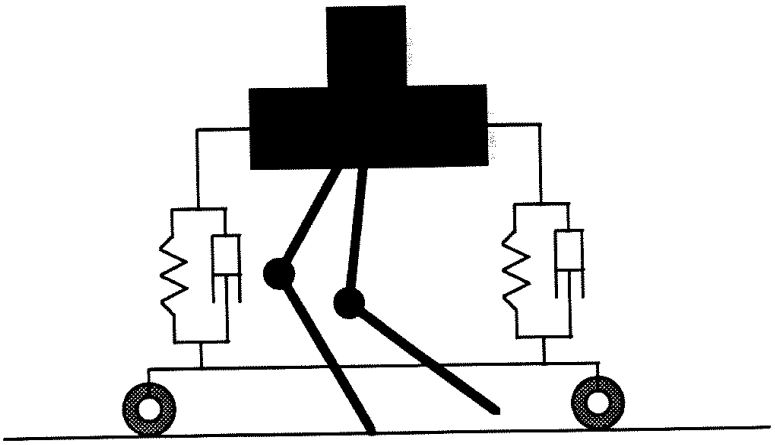

Fig. 6. Spring Turkey with virtual granny walker mechanism.

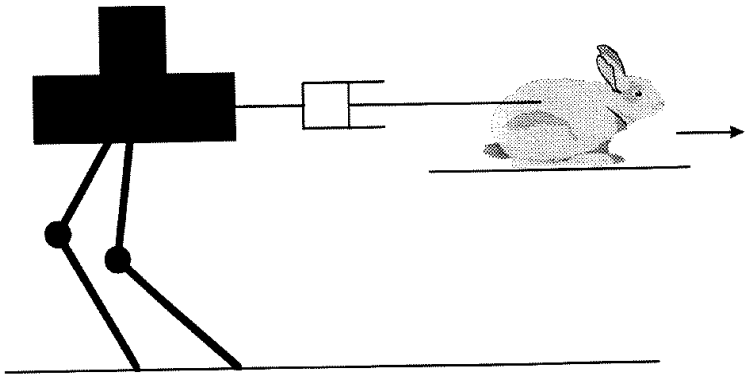

Fig. 7. Spring Turkey with virtual dogtrack bunny mechanism.

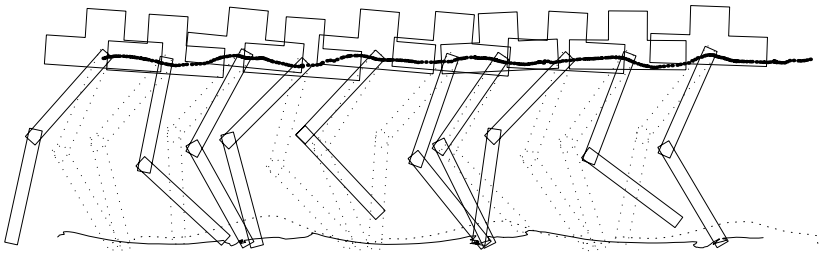

Fig. 8. Elapsed time snapshot of the bipedal walking data in Figure 9. The drawings of the robot are spaced approximately 0.5 s apart. The left leg is dotted, whereas the right leg is solid. Lines show the path of the tips of the foot and the center of the body.

overcome sloped terrain of unknown slopes and transition locations. The resulting algorithm is applied to a simulated Spring Flamingo walking over a rolling terrain. We assume that the terrain has no discontinuous vertical variation. The minimum and maximum slope gradients are assumed to be -20 degrees and $+20$ degrees, respectively. We also assume that no slippage occurs.

The simulated biped has only discrete sensors at the toe and heel to sense the foot's contact with the ground. When encountering a slope, the biped adjusts its gait accordingly to continue stable locomotion on it. In the double-support phase, when one or both of the biped's feet are on a slope, the front and rear supporting legs are resting on different elevations. We call the virtual slope that is formed by joining the ankle of the front and rear legs the global slope. The actual slope that each foot is resting on is called the local slope. The local and global slope gradients are computed based on the joint position data at the beginning of the double-support phase.

6.1. Upslope and Downslope Walking

There are many possible postures and behaviors for the biped to adopt while walking on an upslope or a downslope. In this preliminary study, we choose to keep the desired horizontal velocity unchanged and the body at the same upright posture while the biped walks on the sloped terrain. We also choose to fix the desired step length of the biped. For energy efficiency, we want the biped to walk with a high posture, but without causing the legs to reach singular configurations. The desired hip height of the biped is computed based on geometric considerations (considering the global slope, transition location from double to single support, and the desired step length). This results in a desired body height that varies with the slope gradient.

For upslope walking, the height limit of the biped is computed based on the desired step length, the distance from the front ankle at which the double-support phase transits to the single-support phase, and the singularity consideration of the back supporting leg during the double-support phase (see Fig. 10). We denote h_{limit} to be the hip height limit measured along the direction of the gravitational field from a global slope. By geometric consideration, the hip height limit h_{limit} is computed by eq. (15):

$$h_{limit} = \sqrt{(l_1 + l_2)^2 - r^2} - r \tan \beta, \tag{15}$$

where $l_1 + l_2$ is the total length of the leg (excluding the foot), β is the slope gradient, and r is the horizontal distance of the desired transition plane (where the biped transits from double- to single-support phase) from the stance ankle.

A factor k_{height} is multiplied to the height limit h_{limit} to give the desired height h^d of the hip from a global slope. k_{height} is typically chosen to be around 0.85. For larger slope variations, it should be smaller to accommodate uncertainties.

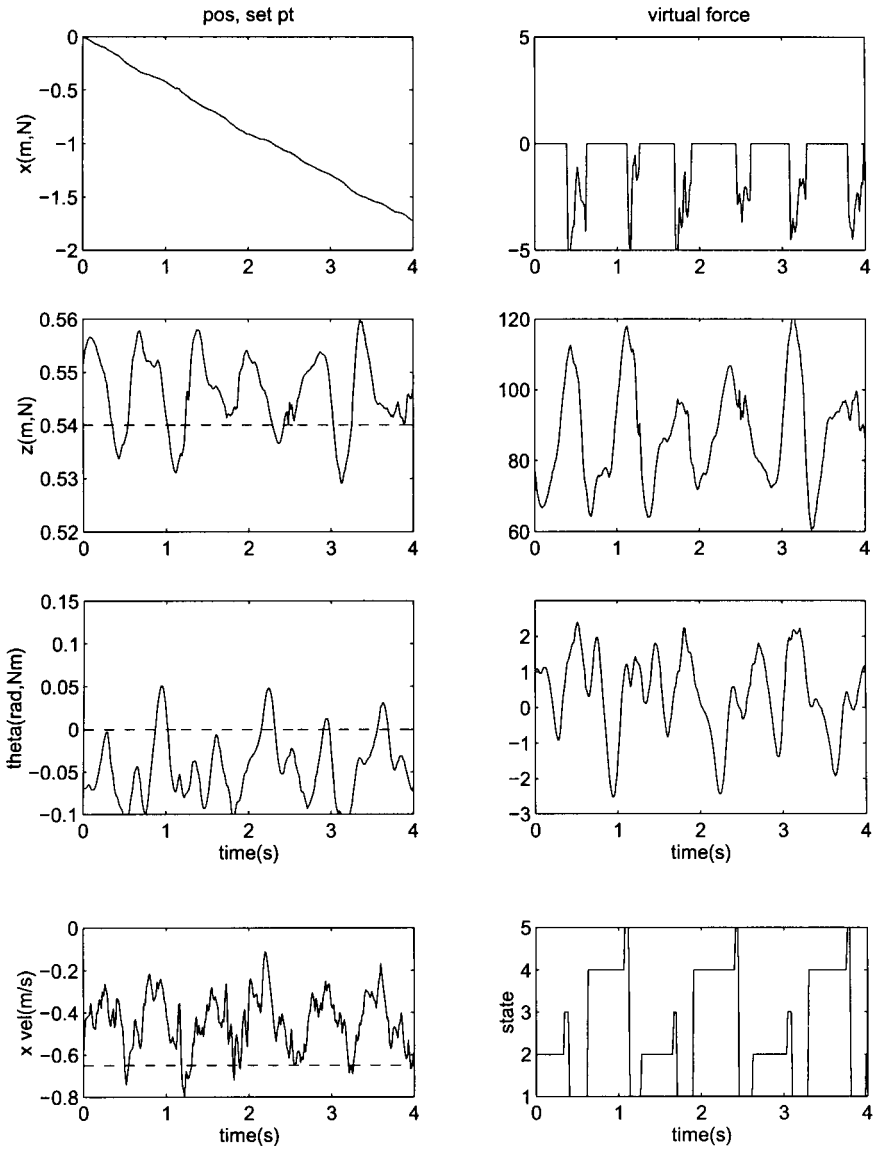

Fig. 9. Turkey-walking data. Upper left graphs display the x, z, and θ positions and virtual spring set points (dashed). Upper right graphs display the resultant forces applied to the body due to the virtual components. The lower left-hand graph shows the body velocity and the dogtrack bunny velocity (dotted). The lower right-hand graph shows the state machine transitions.

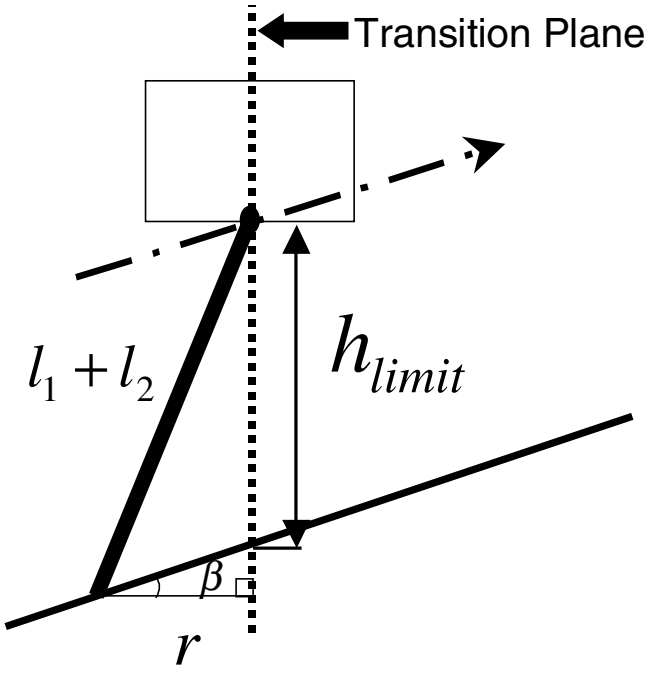

Fig. 10. Geometric constraint to calculate h_{limit}: level and upslope.

For the downslope, depending on the swing leg's touch-down location and slope gradient, both the back supporting leg during the double-support phase and the swing leg during the single-support phase may reach singular configuration as shown in Figures 11a and 11b, respectively. In the algorithm, we have set the swing leg to touch down when the vertical projection of the hip is between both legs. For such a strategy, the case depicted in Figure 11b can occur and the corresponding height limit h_{limit} is computed as in eq. (16):

$$h_{limit} = \sqrt{(l_1 + l_2)^2 - r_1^2} - r_1 \tan\beta, \qquad (16)$$

where r_1 is the expected horizontal distance of the swing leg touch-down location measured from the hip.

The minimum of the computed hip height limit h_{limit} between eqs. (15) and (16) is used to compute the desired hip height h^d during the downslope walking. These equations can also be used for level walking where β is zero.

6.2. Transition Cases

This subsection considers transitional walking from level to slope ground and from slope to level ground, where the slope can be either ascending or descending. We first discuss the level-to-upslope and downslope-to-level transitions. Both result in premature landing of the swing leg.

For the level-to-upslope transition, we employ a strategy similar to the one adopted by Zheng and Shen (1990). However, instead of using the force feedback approach, we set the ankle torque of the swing leg to zero when the swing foot touches the upslope surface. This allows the swing foot to orient itself and adapt to the slope according to the natural compliance (for details, see Chew 1998).

When both the heel and the toe of the swing foot are on the slope, they trigger the transition from the single-support phase to the double-support phase. The biped then computes the gradient of the global slope based on the joint angles. Note that the global slope gradient is not the same as the real slope gradient, since both feet are not on the same slope yet (see Fig. 12b). The global slope is an imaginary intermediate slope whose gradient is between the level ground gradient (equal to zero) and the actual upslope gradient. However, the biped considers the global slope to be the actual terrain slope during the double-support phase. It computes the desired walking height based on the global slope.

When the biped switches to the single-support phase, it continues to compute the desired walking height based on the global slope. The swing leg trajectory of the single-support phase is also planned using the global slope. However, when the biped is in the double-support phase again, both its legs are on the actual slope, and the global slope will have the same gradient as the actual slope (see Fig. 12c).

Note that during transition walking, the actual step length of the biped may vary significantly from the desired step length. For the level-to-upslope transition, the variation in the step length is due to the premature landing of the swing foot. The biped will reach a steady walking gait as it continues to walk on the same slope.

The usage of global slope instead of local slope to compute the desired height of the biped is analogous to the use of a low-pass filter to get rid of high-frequency noise. For example, the biped may walk on a ground with many local variations in elevation. Using global slope information will reduce unnecessarily large variation in trajectory during locomotion. The strategy for the level-to-upslope transition is also applied to the downslope-to-level transition.

We now consider the level-to-downslope and upslope-to-level transitions. During the level-to-downslope transition, the biped executes the usual walking control for single-support phase until the swing time has expired. When the swing time expires, if the swing foot has not touched down, the biped will exert a preset downward force exerted at the swing leg's ankle so that the swing foot will continue on its way down. The downward force will cause the swing leg's ankle to penetrate through a virtual surface (which is the global slope) (see Fig. 13a). When the virtual surface is penetrated, the biped recomputes the gradient of the global slope based on the instantaneous position of the swing leg's ankle (see Fig. 13b) and adjusts the desired hip height accordingly. This is a continuous process. After the swing leg has touched down, the state machine will switch to the double-support phase, and the biped will compute the intermediate global slope (see Fig. 13c). In this phase, the biped will

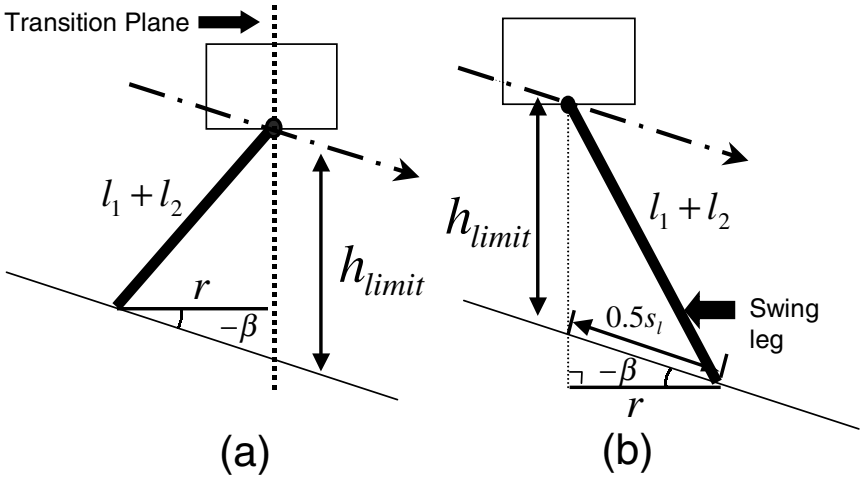

Fig. 11. Geometric constraint to calculate h_{limit} during downslope walking: (a) case 1, (b) case 2.

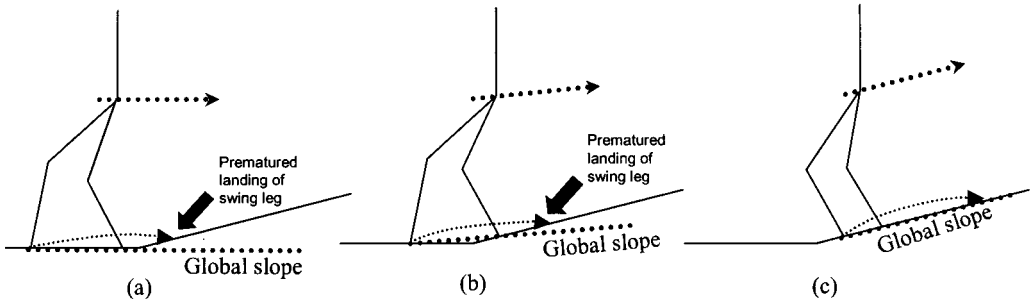

Fig. 12. Sequence for level-to-upslope transition. Note that in (b), global slope gradient is different from the actual slope gradients underneath both feet. In (c), all the detected slopes are the same.

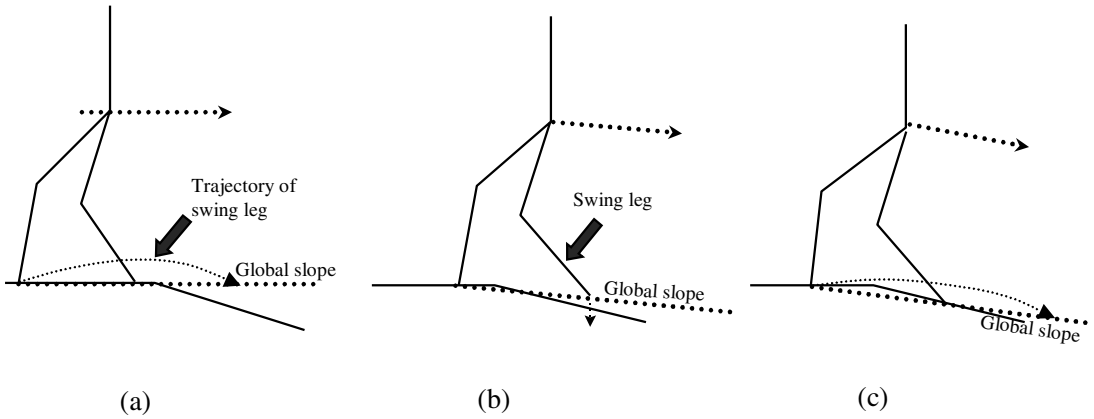

Fig. 13. Sequence for level-to-downslope transition.

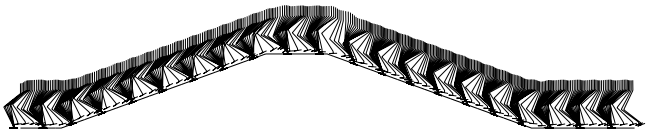

Fig. 14. Stick diagram of the biped walking over the sloped terrain Profile One from left to right (spaced approximately 0.08 s apart and showing only the left leg).

compute the desired hip height based on the global slope. The swing leg of the next single-support state will also follow this global slope. The biped then goes through the whole sequence again before both feet are on the same slope. The strategy for the level-to-downslope transition can also be applied to the upslope-to-level transition.

For both the level-to-downslope and upslope-to-level transitions, we have to control the swing leg so that it will not hit the ground prematurely. To avoid such an event, the desired lift height for the swing leg is set to a value that will enable it to clear all the edges of the level-to-downslope and upslope-to-level transitions. This is based on the assumption that we know the maximum change in the gradients for each slope transition.

Although these subsections only describe the strategies used to handle four simple slope transition cases, they are applicable to other transition cases as long as there is no discontinuous vertical variation.

6.3. Simulation Results

The simulated biped was tested over several terrain profiles, two of which we discuss below. Sloped terrain Profile One (shown in the top graph of Fig. 15) was used in the simulation to observe and compare the key output variables for the upslope, downslope, and transition terrain walking of the biped. Sloped terrain Profile Two (shown in the top graph of Fig. 16), which is rougher than Profile One, was used to illustrate the robustness of our approach for the biped to walk over a sloped terrain that consisted of a series of unknown slopes.

The desired values of the key gaits' variables and the virtual components' parameters were set as in Table 2. The desired hip height of the biped was computed as discussed earlier.

6.3.1. Walking over Terrain Profile One

Figure 14 shows a stick diagram of the biped walking over the sloped terrain Profile One from left to right. Figure 15 shows the profiles of the key variables. In Figure 15, the vertical dash-dot and dotted lines represent the start of the single-support phase and the double-support phase, respectively. Note that the variables α and M_α correspond to θ and f_θ, respectively, introduced in Section 4.

The third graph from the top of Figure 15 shows the actual hip height profile (solid line) of the biped measured from the global slope. This graph shows that the biped was able to track the desired hip height (dashed line) within a tolerable range. The ripple found in the actual hip height profile was mainly due to the swing leg dynamics and the variation in the effective mass of the biped as perceived at the hip.

We observe that when the global slope gradient was negative, the desired hip height (dashed line in the top third graph of Fig. 15) of the biped was higher than when the global slope gradient was positive. This was the result of the geometric considerations when we computed the desired hip height of the biped.

Table 2. Desired Values of Some of the Gait Variables and the Parameters of the Virtual Components Set in the Simulation

Variable/Parameter	Value
Gait variables	
Desired pitch angle of the body, θ_d	0 rad
Desired horizontal velocity of the hip, $\dot{x}_d$	0.4 m/s
Desired step length, s_l	0.28 m
Distance from the front ankle at which double support phase transits to single support phase, l_t	−0.03 m
Desired lift height of swing leg, h_l	0.07 m
Virtual components' parameters	
Spring stiffness in z direction, k_z	500 N/m
Damping coefficient in z direction, b_z	200 Ns/m
Damping coefficient in x direction, b_x	200 Ns/m
Spring stiffness in θ direction, k_θ	50 Nm
Damping coefficient in θ direction, b_θ	20 Nms

6.3.2. Walking over Terrain Profile Two

The results from the terrain Profile Two demonstrate the robustness of the algorithm developed for the sloped terrain walking. Figure 16 shows the profiles of the key variables. Note that the profile of the terrain was not known to the biped in advance. The biped was required to *feel* its way through the terrain. From Figure 16, we observe that the biped can cope with this rougher terrain without much difficulty. The actual hip height, hip horizontal velocity, and pitch angle of the body were well behaved in the simulation.

7. Conclusions

This paper has demonstrated the successful application of virtual model control to two bipeds for walking tasks. We stress here that we *augmented* the natural dynamics of the robot with simple virtual components rather than attempt to *cancel* the natural dynamics. In no case did we assume linear dynamics.

This paper has also demonstrated the successful application of virtual model control to a simulation of Spring Flamingo walking dynamically and steadily over sloped terrain with unknown slope gradients and transition locations.

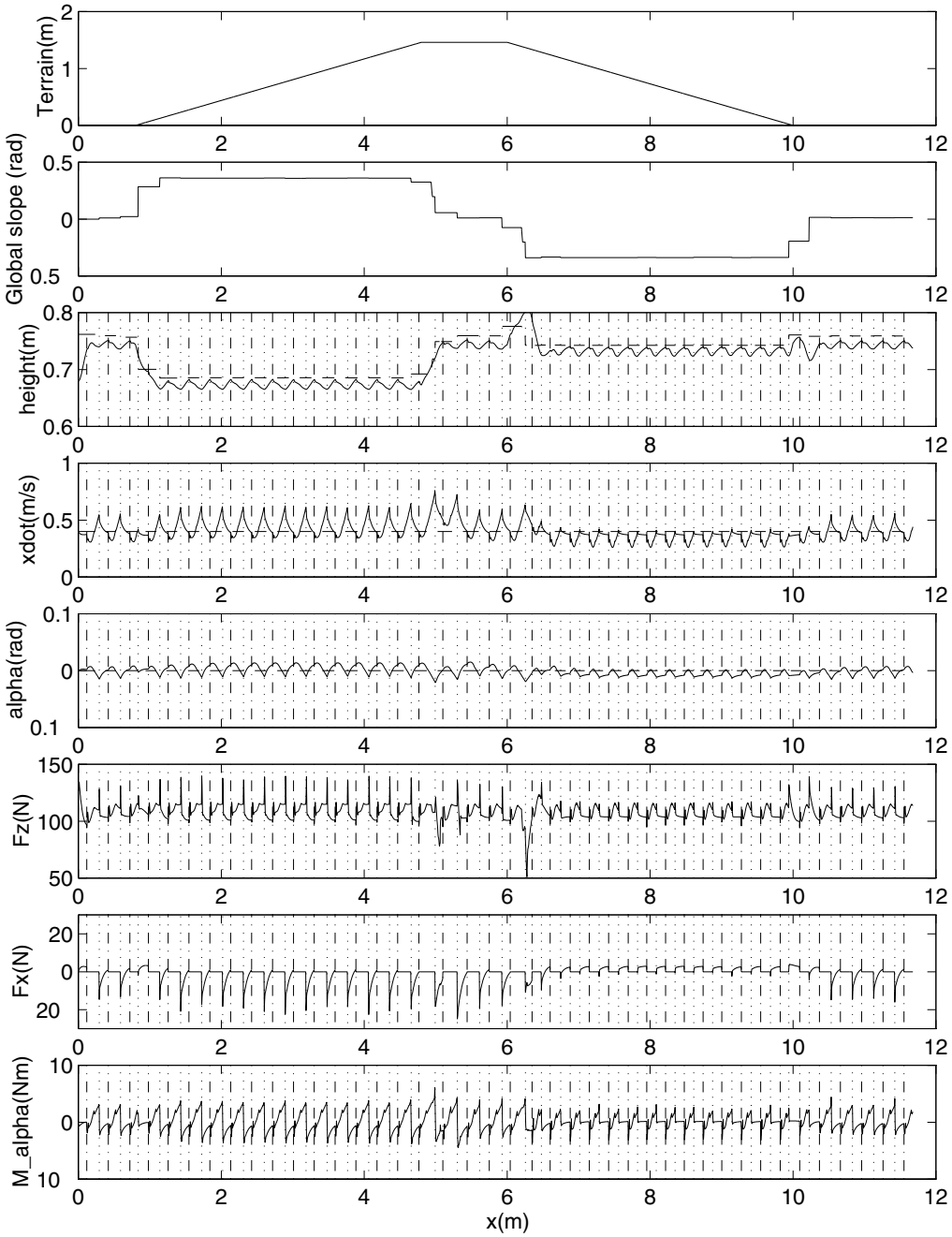

Fig. 15. Profiles of the key variables when the biped walked over the sloped terrain Profile One.

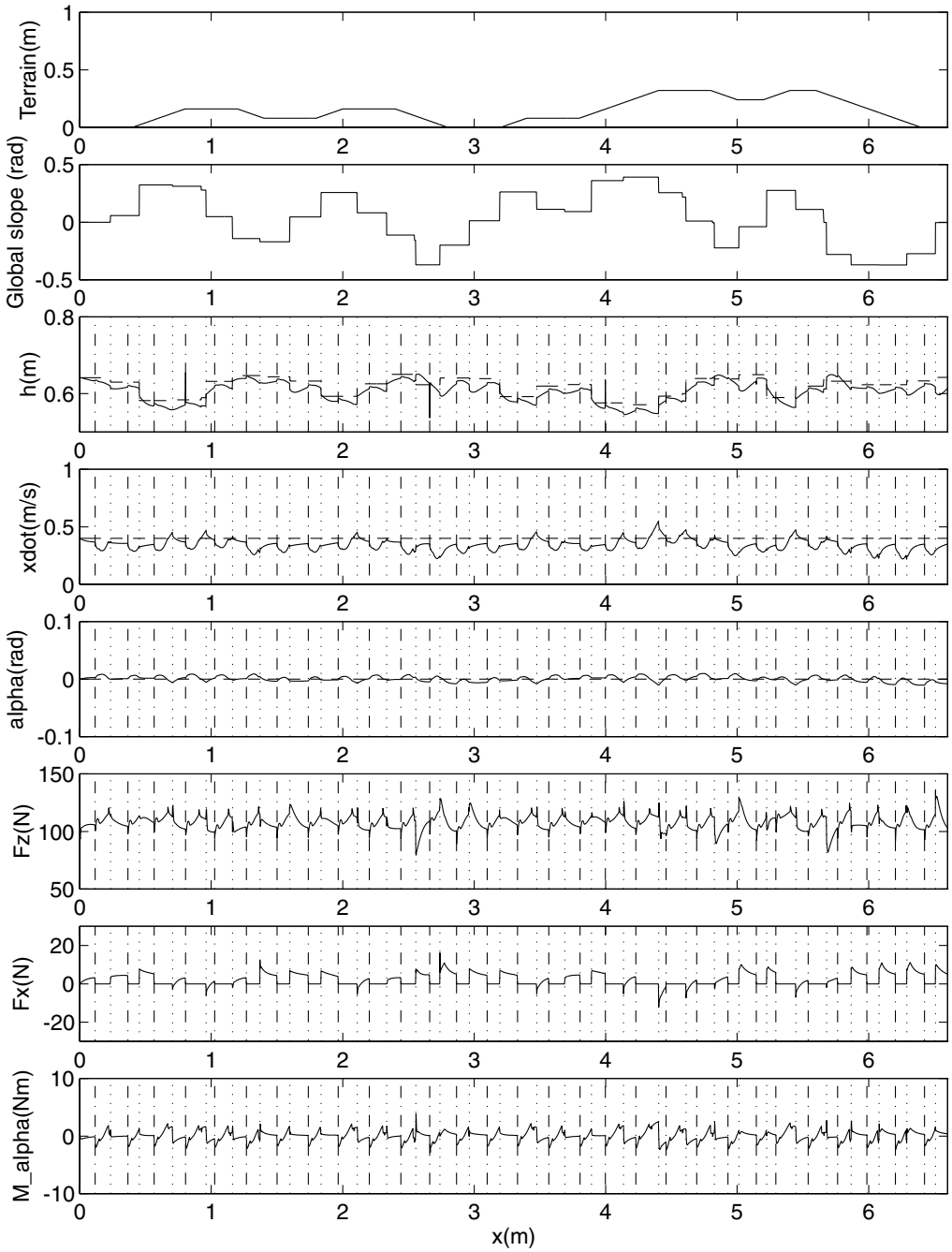

Fig. 16. Profiles of the key variables when the biped walked over the sloped terrain Profile Two.

It was assumed that the slope gradients were between −20 degrees and +20 degrees, and that the sloped terrain had maximum transitional gradient change of less than 20 degrees. In the implementation, the global slope was used to compute the desired hip height based on geometric considerations, which resulted in a straight line trajectory parallel to the global slope. It was very simple to implement and did not require an extensive sensory system to achieve blind walking. The algorithms can be extended to more abrupt terrain changes, such as stair climbing, if properties such as the general location and layout of the terrain are known or detected visually.

The ease of implementing virtual model control is promising. Although some intuition is needed to identify the types and locations of virtual components in a virtual model control implementation, the resulting algorithm usually has low computational requirement. One of our goals is to automate the design process.

Acknowledgment

This work was supported in part by the Office of Naval Research, Grant #N00014-93-1-0333.

References

Bay, J. S., and Hemami, H. 1987. Modeling of a neural pattern generator with coupled nonlinear oscillators. *IEEE Transactions on Biomedical Engineering* BME-34:297–306.

Chew, C.-M. 1998. *Blind Walking of a Planar Biped over Sloped Terrain.* Master's thesis, Massachusetts Institute of Technology.

Chew, C.-M., and Pratt, G. A. 1999. A minimum model adaptive control approach for a planar biped. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyongju, Korea.

Furusho, J., and Sano, A. 1990. Sensor-based control of a nine-link biped. *International Journal of Robotics Research* 9(2):83–98.

Golliday, C. L., and Hemami, H. 1977. An approach to analyzing biped locomotion dynamics and designing robot locomotion controls. *IEEE Transactions on Automatic Control* AC-42:963–973.

Gubina, F., Hemami, H., and McGhee, R. B. 1974. On the dynamic stability of biped locomotion. *IEEE Transactions on Biomedical Engineering* BME-21:102–108.

Hogan, N. 1985. Impedance control: An approach to manipulation: Part I—Theory, Part II—Implementation, Part III—Applications. *Journal of Dynamic Systems, Measurement and Control* 107:1–24.

Katoh, R., and Mori, M. 1984. Control method of biped locomotion giving asymptotic stability of trajectory. *Automatica* 20:405–414.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *IEEE Journal of Robotics and Automation* 5(1):90–98.

Mita, T., Yamaguchi, T., Kashiwase, T., and Kawase, T. 1984. Realization of a high speed biped using modern control theory. *International Journal of Control* 40(1):107–119.

Miura, H., and Shimoyama, I. 1984. Dynamic walk of a biped. *International Journal of Robotics Research* 3(2):60–74.

Pratt, G. A., and Williamson, M. M. 1995. Series elastic actuators. *IEEE International Conference on Intelligent Robots and Systems* 1:399–406.

Pratt, J. E. 1994. *Learning Virtual Model Control of a Biped Walking Robot.* Unpublished project report, Massachusetts Institute of Technology.

Pratt, J. E. 1995. *Virtual Model Control of a Biped Walking Robot.* Master's thesis, Massachusetts Institute of Technology.

Pratt, J., Torres, A., Dilworth, P., and Pratt, G. 1996. Virtual actuator control. *IEEE International Conference on Intelligent Robots and Systems,* Osaka, Japan.

Raibert, M. H., and Craig, J. J. 1981. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control* 102:126–133.

Salisbury, K. 1980. Active stiffness control of a manipulator in Cartesian coordinates. *19th IEEE Conference on Decision and Control*, pp. 83–88.

Torres, A. L. 1996. *Implementation of Virtual Model Control on a Walking Hexapod.* Undergraduate thesis, Massachusetts Institute of Technology.

Vukobratovic, M., Borovac, B., Surla, D., and Stokic, D. 1990. *Biped Locomotion: Dynamics, Stability, Control, and Applications.* Berlin: Springer-Verlag.

Zheng, Y. F., and Shen, J. 1990. Gait synthesis for the SD-2 biped robot to climb sloping surfaces. *IEEE Transactions on Robotics and Automation* 6(1):86–96.