Recursive Workspace Control of Multibody Systems: A Planar Biped Example*

Jeffrey M. Wendlandt† S. Shankar Sastry

Intelligent Machines and Robotics Laboratory
University of California at Berkeley

Abstract

This paper presents a method of controlling multibody systems using a recursive formulation of workspace control. Using recursive multibody dynamics techniques, we generate the model for a workspace controller avoiding the need to symbolically create the complex equations of motion. The workspace controller is then applied to make a planar biped balance. The paper first describes the biped model and then presents the recursive workspace controller. Finally, the paper presents the results of a simulation in which the controller commands the biped to lower to a standing position.

1 Introduction

Predictive computer models for human motion have applications in animation [1], sports equipment design, soldier equipment design, and soldier training. To be effective, however, the predictive models must react appropriately and dynamically to changing body positions, varying loads, and external forces. Creating predictive models involves realistic simulation as well as the more difficult problem of designing controllers to command the motion of the model. The simulation and control problem is complicated by the fact that the model is often in contact with the environment. The human body is often modelled as a multibody (MB) – a collection of rigid bodies interconnected by joints. Designing controllers for MB systems is difficult since the equations of motion are lengthy, complex, nonlinear, coupled differential equations. An additional difficulty is that the equations of motion change as the model contacts the environment.

In this paper, we create a planar MB model, shown in Figure 1, of a human and design a model-based controller to balance the MB system. The controller creates an approximate model of the MB system through recursive MB techniques. We calculate the workspace dynamics, also known as operational space dynamics [2], for the central body through several passes from rigid body to rigid body in the multibody structure. We formulate the controller in the central body to provide an intuitive relationship between input and output. We then use this model to calculate a desired body force which is translated to joint torques through another pass through each rigid body in the multibody structure. The recursive multibody techniques alleviate the need to symbolically generate the complex differential equations of motion.

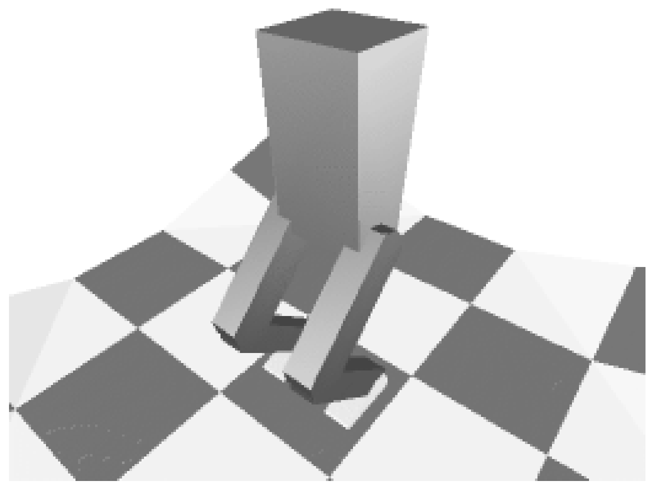

Figure 1: Standing Biped

The controlled system is simulated in 3D using *Impulse* [3] [4], a multibody simulator that is designed to handle contact. *Impulse* simulates the controlled system and provides the inputs to the recursive workspace controller. The simulator calls the controller at a user specified rate.

The first section of the paper describes the model of the planar biped. The second section describes the recursive workspace controller in detail. The next section presents simulation results of the biped lowering the body to a standing position. The paper concludes with a discussion of future work.

*Funded through ARO under grant DAAH-O4-96-1-0119 (AASERT) and DAAH-04-95-1-0588

†wents@eecs.berkeley.edu

2 Description of Biped Model

The model of the planar biped in the zero configuration is shown in Figure 2. The multibody system consists of a central body, a right leg and foot, and a left leg and foot. The central body is labeled as link 1 and the link index increases as one moves outwardly from the central body. The first joint is joint 2 and is the joint between link 1 and link 2. Each joint is a single degree of freedom rotational joint aligned with the $+y$-axis. A positive joint angle corresponds to a rotation of an adjacent outboard link about the $+y$-axis relative to the adjacent inboard link. Outboard indicates a progression from link to link in a direction towards one of the feet. Inboard indicates a progression toward the central body. Positive torques are referenced in the same manner as the joint angles. The location of each joint relative to the inertial frame is given in braces in Figure 2.

The masses of each link and the inertia about the y-axis is also shown in Figure 2. The polygonal models were first designed based on measurements of a human subject. The density of water was used to calculate the mass and inertia of each body. The mass and inertia of each body is calculated in *Impulse*. The central body has a mass of 45 Kg, a weight of 99.2 lbs. Most of the total mass of 72.5 Kg is concentrated in the central body. The location of the center of mass of each link is given in Figure 2 and is given in parentheses. The principal axes of inertia are shown in Figure 2 and are aligned with the inertial coordinate axes except for the two feet. The principal axes of the feet are rotated about the $-y$-axis of the inertial frame by 8.547 degrees.

The sensors available to the controller are the joint angles; joint velocities; central body orientation and position relative to an inertial frame; and the body velocity of the central body. The sensor inputs are obtained from *Impulse*, and *Impulse* calls the controller at a user specified rate.

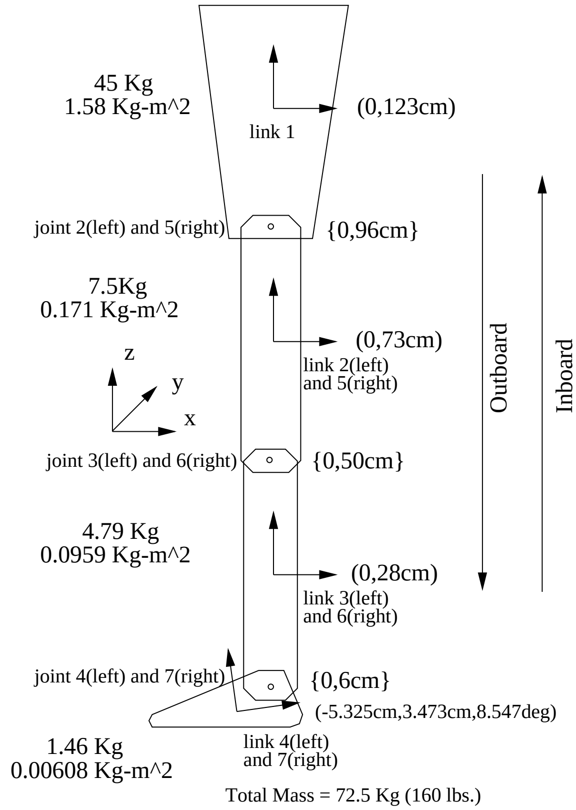

Figure 2: Biped Model

3 Recursive Workspace Controller

The block diagram for the recursive workspace controller is shown in Figure 3. We use the term recursive and iterative to mean successive link to link calculations. The calculations performed for the recursive workspace controller are contained in the outer shaded box. The inboard and outboard recursive steps are indicated in the interior, darker box. The controller outputs joint torques. The inputs to the controller are the multibody (MB) states (central body position and orientation, center of mass velocity, central body angular velocity, joint angles, and joint velocities); desired central body acceleration, velocity, and position; and position and velocity gains. The desired values and gains are provided by a higher level controller. At this time, the designer chooses these values.

The MB states first pass through the velocity propagation stage. The velocity propagation stage consists of an outboard recursion and calculates the body velocity of each link based on the body velocity of the inboard link, the joint position, and the joint velocity. Coriolis and gravitational forces are also calculated during this stage as well as the forward kinematics. We use the term forces to indicate both forces and torques.

The information calculated in the velocity propagation stage is passed to the workspace dynamics calculation stage. The workspace dynamics calculation consists of an inboard recursion from the feet to the central body. The workspace dynamics calculation produces the approximate workspace dynamics for the central body. This stage calculates the articulated body(AB) inertia [5] and bias forces of the inboard link based on the AB inertia and bias forces of the outboard link. The AB inertia and bias forces relate an external force applied to a link to the acceleration of that link and take into account the links outboard to the link. The AB inertia and bias force for the central body provide the approximate workspace dynamics. AB inertias and bias forces are discussed in many references including

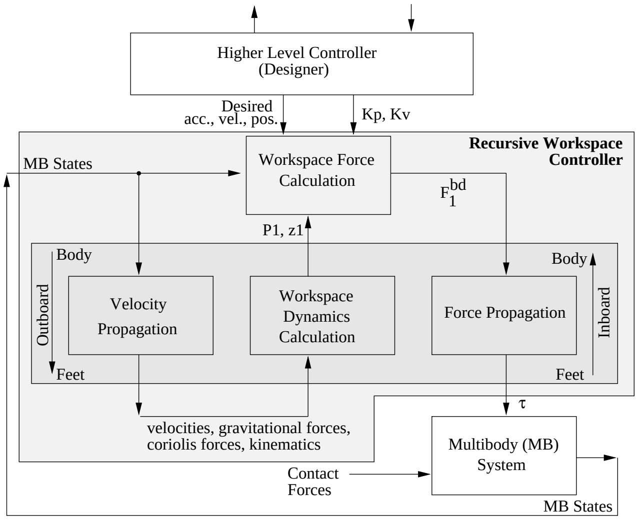

Figure 3: Controller Block Diagram

[5], [6], [7], and [3]. The recursive workspace controller assumes that the feet are fixed to the floor. We approximate this condition from a *big base* assumption presented in [6]. The approximation assigns the feet a large mass and inertia (approximating the Earth) and sets the feet bias forces to zero. There exists exact methods for calculating the workspace dynamics and a method is derived in [8]. The exact workspace inertia is calculated [6]. We have derived an exact workspace dynamics algorithm in terms of Lie group notation but do not present it in this paper. The exact method requires an additional central body to feet recursion. We have achieved satisfactory results with the approximate workspace calculation and may not implement the exact method for the planar biped.

We now present the various components of the recursive workspace controller in more detail.

3.1 Initialization

The inputs to the controller are the desired positions, velocities, and accelerations; body velocity and pose of the central body; gains; and the joint positions and velocities. These inputs are used to initialize the recursive workspace controller. The initialization is as follows:

`Given` V_1^b, $g_{0,1}$, θ, $\dot{\theta}$

`Initialize Variables`

$$a_1^b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.1}$$

$$b_1^b = -g_{1,0}\begin{bmatrix} 0 \\ -m_1 g \\ 0 \end{bmatrix} + \begin{bmatrix} +m_1\dot{\theta}_1 V_{1,z}^b \\ -m_1\dot{\theta}_1 V_{1,x}^b \\ 0 \end{bmatrix}, \tag{3.2}$$

where $g_{0,1}$ is in $SE(2)$, the special Euclidean group in the plane, and maps central body coordinates into the corresponding coordinates in the inertial frame. The inertial frame is link 0. An element of $SE(2)$ is represented as a 3x3 matrix and is in the form:

$$g_{0,1} = \begin{bmatrix} R_{0,1} & p \\ 0 & 1 \end{bmatrix} \tag{3.3}$$

$$g_{1,0} = g_{0,1}^{-1}$$

$$g_{1,0} = \begin{bmatrix} R_{1,0} & -R_{1,0}p \\ 0 & 1 \end{bmatrix}. \tag{3.4}$$

The joint angles are given by θ and the joint velocities by $\dot{\theta}$. The gravitational constant is given in g. The angle of the central body is θ_1 and the first joint angle in the left leg is θ_2. The mass of the ith link is m_i and the inertia of the ith link is $\mathbb{I}_i$. The body velocity, the inertial velocity written in terms of the body frame, of the ith link is V_i^b, and the x component of V_i^b is denoted $V_{i,x}^b$.

3.2 Velocity Propagation

We now compute the body velocity of each link in an outboard recursion. The velocity propagation stage calculates body velocities, gravitational forces, and coriolis forces. The algorithm is given below:

```
For i = (min joint index) to (max joint
index)
```

$$g_{i.o,0} = g_{i.o,i.i}(\theta_i)g_{i.i,0} \tag{3.5}$$

$$A_{i.i}^{i.o} = \mathrm{Ad}_{g_{i.o,i.i}(\theta_i)} \tag{3.6}$$

$$V_{i.o}^b = A_{i.i}^{i.o}V_{i.i}^b + H_{i.o}\dot{\theta}_i \tag{3.7}$$

$$a_i^b = \dot{A}_{i.i}^{i.o}V_{i.i}^b \tag{3.8}$$

$$b_{i.o}^b = -g_{i.o,0}\begin{bmatrix} 0 \\ -m_{i.o}g \\ 0 \end{bmatrix} + \begin{bmatrix} +m_{i.o}\dot{\theta}_{i.i}V_{i.o,z}^b \\ -m_{i.o}\dot{\theta}_{i.i}V_{i.o,x}^b \\ 0 \end{bmatrix} \tag{3.9}$$

```
end loop
```

where (`min joint index`) = 2 and the (`max joint index`) = 7. The map $g_{j,k}$ takes coordinates of a point or vector in link k and gives the corresponding coordinates in link j. The symbol, $i.o$, gives the link index of the link outboard to joint i. The symbol, $i.i$, gives the link index of the link inboard to joint i. In Equation (3.6), Ad_h is the adjoint for $h \in$SE(2) and is a 3x3 matrix given by

$$\mathrm{Ad}_h = \begin{bmatrix} R & \begin{matrix} -pz \\ +px \end{matrix} \\ 0 & 1 \end{bmatrix}, \tag{3.10}$$

where

$$h = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}. \tag{3.11}$$

The adjoint appropriately transforms velocities and forces in different frames. See [9] for a background in using Lie group theory in robotics. The joint map is given by $H_{i.o}$ and is a vector that represents the twist of joint i written in the link frame outboard to joint i, the frame of link $i.o$. The twist represents the rigid body velocity of a frame and is analogous to angular velocity. For the discussion in this paper, $H_{i.o}\dot{\theta}_i$ gives the velocity component written in terms of link $i.o$ due to the relative velocity between link $i.i$ and link $i.o$. The term $A_{i.i}^{i.o} V_{i.i}^b$ transforms the body velocity of link $i.i$ to link $i.o$ coordinates. The coriolis terms are contained in a_i^b and in the last term of $b_{i.o}^b$. The first term in $b_{i.o}^b$ is the gravitational force written in link $i.o$ coordinates. Again, for a more detailed discussion of the notation of twists and adjoints, see [9]. In [10], the authors derive the solution of the inverse dynamics problem (given joint angles, velocities, and accelerations calculate the necessary torques) for serial chains in terms of Lie group theory and give a background in twists and adjoints.

3.3 Workspace Dynamics Calculation

The *approximate* workspace dynamics are calculated through a recursion from the feet to the body. The algorithm calculates the articulated body (AB) inertia [5] and bias forces of the inboard link based on the AB inertia and bias force of the outboard link. The AB inertia and bias forces for the central body are calculated in the last step in the iteration. The algorithm is given below:

`For j = (min link index) to (max link index)`

$$z_j = b_j^b \tag{3.12}$$

$$P_j = M_j = \begin{bmatrix} m_j & 0 & 0 \\ 0 & m_j & 0 \\ 0 & 0 & \mathbb{I}_j \end{bmatrix} \tag{3.13}$$

`end loop`
`Apply fixed base approximation`

$$M_7 = M_4 = 10^6 M_1 \tag{3.14}$$

$$b_7^b = b_4^b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.15}$$

`For i = (max joint index) to (min joint index)`

$$D_{i.o} = H_{i.o}^T P_{i.o} H_{i.o} \tag{3.16}$$

$$K_{i.o} = H_{i.o}(D_{i.o})^{-1} H_{i.o}^T \tag{3.17}$$

$$L_{i.i}^{i.o} = [I - K_{i.o} P_{i.o}] A_{i.i}^{i.o} \tag{3.18}$$

$$P_{i.i} = P_{i.i} + (L_{i.i}^{i.o})^T P_{i.o} A_{i.i}^{i.o} \tag{3.19}$$

$$z_{i.i} = z_{i.i} + (L_{i.i}^{i.o})^T (z_{i.o} + P_{i.o} a_i) \tag{3.20}$$

`end loop`

The bias forces, z_j, and AB inertias, P_j, are first initialized to the values given above. The fixed base approximation is then applied to the two feet. The next inboard recursion calculates the AB inertia for each link based on the AB inertia of the outboard link. The approximate calculation of the workspace inertia is given in [6] and an exact calculation with coriolis and gravitational forces is given in [8] in terms of the spatial operator algebra. We have derived a recursive algorithm to calculate the exact workspace dynamics in terms of Lie group theory and the steps in this algorithm share many similarities with the algorithm derived by the spatial operator algebra in [8].

3.4 Workspace Force Calculation

The previous step provides the approximate workspace inertia and workspace bias forces. The workspace inertia is the AB inertia for link 1, P_1. The workspace coriolis forces and gravitational forces are contained in z_1. The desired body force on the central body, F_1^{bd}, is calculated in the following way:

`Given` x_1^d, z_1^d, θ_1^d, V_1^{bd}, and $\dot{V}_1^{bd}$

$$e = \begin{bmatrix} R_{1,0} \begin{bmatrix} x_1 - x_1^d \\ z_1 - z_1^d \end{bmatrix} \\ \theta_1 - \theta_1^d \end{bmatrix} \tag{3.21}$$

$$\dot{e} = V_1^b - V_1^{bd} \tag{3.22}$$

$$F_1^{bd} = P_1(\dot{V}_1^{bd} - K_v \dot{e} - K_p e) + z_1. \tag{3.23}$$

The desired force is calculated with a computed-torque control law in the workspace. The desired values are denoted with a superscript d.

3.5 Force Propagation

The desired body force is then converted into actuator torques through an outboard iteration. The recursion calculates the torques given the desired body force. The torques calculated satisfy, $\tau = J_b^T F_1^{bd}$, where J_b is a function of the leg joint positions and maps joint velocities into body velocities. There is one body Jacobian for each leg. Calculating $\tau = J_b^T F_1^{bd}$ is a static equilibrium calculation and can be calculated by finding the equilibrium forces on each link as though each link were at rest. The algorithm is given below:

`Divide desired force to each leg`

$$F_{2.i}^{bd} = F_{5.i}^{bd} = \frac{F_1^{bd}}{2} \tag{3.24}$$

`For i = (min joint index) to (max joint index)`

$$F_{i.o}^{bd} = -(A_{i.o}^{i.i})^T F_{i.i}^{bd} \tag{3.25}$$

$$\tau_i = H_{i.o}^T F_{i.o}^{bd} \tag{3.26}$$

```
 if link i.o is not a leaf
```

$$F^{bd}_{(i+1).i} = -F^{bd}_{i.o} \tag{3.27}$$

```
 endif
end loop
```

The desired force is first divided into a component along each leg. For simplicity, the algorithm evenly divides the force among the legs. More sophisticated force division algorithms can be used in the future. The force, $F^{bd}_{i.o}$, is the desired force acting on link $i.o$ from link $i.i$ and is written in terms of the frame of link $i.o$. The force, $F^{bd}_{i.i}$, is the desired force acting on link $i.i$ from link $i.o$ and is written in terms of the frame of link $i.i$. Equation (3.25) is based on equal and opposite reaction forces and transforming the forces to the same coordinate system. Equation (3.27) is an equilibrium equation for link $i.o$ which is link $(i+1).i$ if link $i.o$ is not a leaf. The two feet are leafs in the tree structure of the planar biped multibody.

4 Simulation Results

The controlled multibody system is simulated in *Impulse* and the recursive workspace controller is called at 100Hz. The desired body angle is 0 degrees and the desired position of the body is $x = 10$ cm and $z = 95$ cm. The desired central body velocities and accelerations are set to 0. The initial body position is $x = 0$ cm and $z = 105$ cm, and the initial body angle is -10 degrees. The initial joint velocities are zero and the initial joint angles are $(\theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = (40.0, -70.4595, 40.4595, 40.0, -70.4595, 40.4595)$ degrees. The coefficient of restitution is 0.1, and the friction coefficient is 3.0. The position gains for each of the three coordinates is 3.0. The velocity gains for each of the three coordinates is 2.46. These gains correspond to complex poles with equal imaginary and real parts for the linear error dynamics. The controller is coded in the C programming language and interfaced to the simulator *Impulse*. *Impulse* evolves the multibody system dynamics in contact with the floor. A more detailed description of *Impulse* can be found in [4].

The body pose and orientation relative to the inertial frame over the 10 second simulation time is shown in Figure 4. The body position and orientation are converging to the desired values. The values at 9.99 seconds are $x = 9.99935$ cm, $z = 95.016$ cm, $\theta_1 = 0.00169$ degrees. The final configuration of the biped is shown in Figure 1.

The six joint angles are shown over time in Figure 5. The dashed lines correspond to the joints for the right leg. There is a noticeable drift in the right leg joint

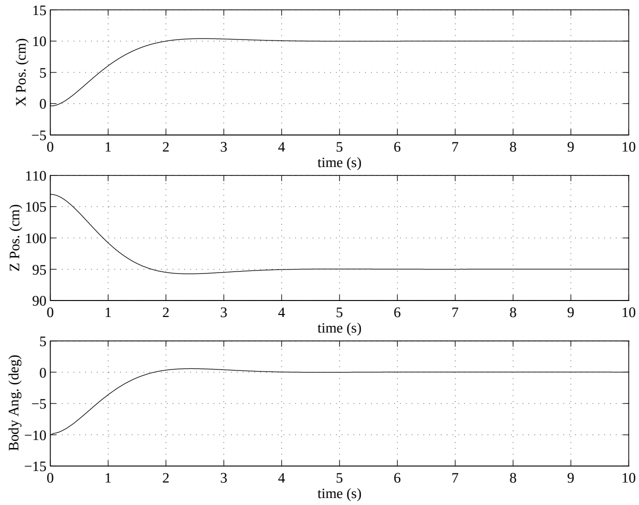

Figure 4: Body Pose Versus Time

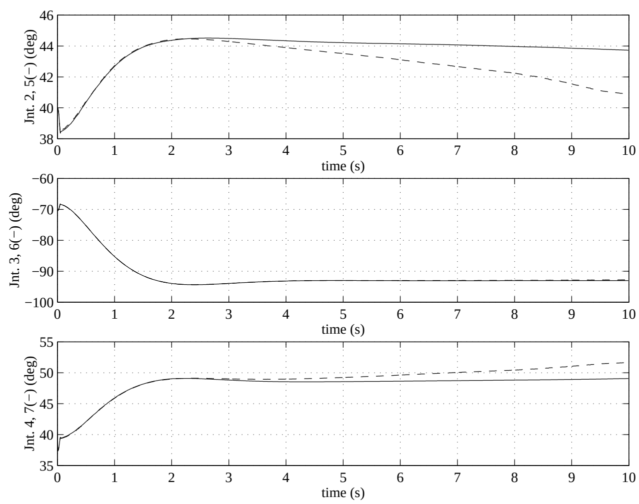

Figure 5: Joint Angle Versus Time

angles. The graphical display reveals that the right foot is slipping slowly backward. The left foot is also slipping but more slowly than the right foot. We believe that this is an artifact of the simulation method used in *Impulse*. We currently are developing a simulator better suited to handle the continuous, planar contact occurring in simulating humans standing, walking and running.

The joint torque versus time is shown in Figure 6. The joint torques drift over time and this is believed to be caused by the feet slipping. Notice that the torque required in the knee joint increases as the knee joint angle becomes more negative.

The location of the center of mass of the body in the $x - z$ plane is shown in Figure 7. The slight overshoot in e_x and e_z cause the curl at the end of the simulation. The body moves along the x direction, toward the back, as it crouches down to the desired height.

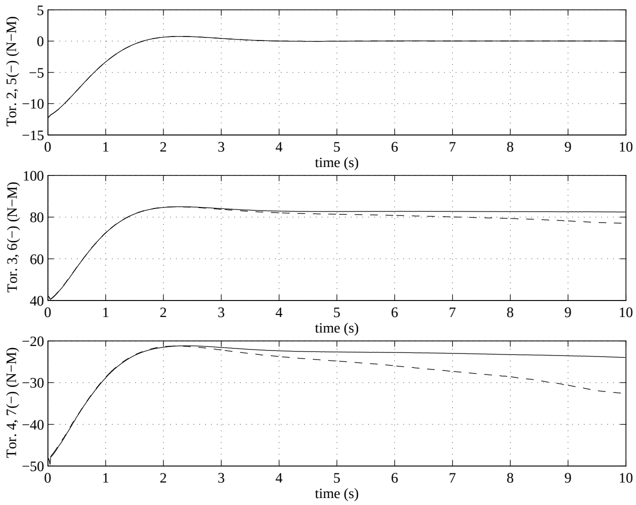

Figure 6: Joint Torque Versus Time

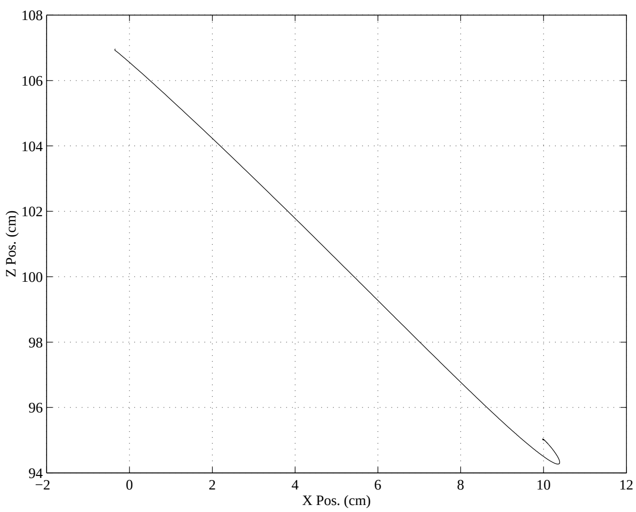

Figure 7: X-Z Body Position Parameterized by Time

5 Conclusion

This paper has demonstrated a control technique for multibody systems and has applied the technique to a multibody model of a planar biped. Recursive multibody dynamic algorithms are used to produce a model in the space that is important to the particular problem. In this case, we designed the controller to stand and the important object in the multibody system is the central body. We, therefore, formed errors in this space and were rewarded with an intuitive relationship between body motion and controller gains. We are pleased with the approximate workspace calculation and may not implement the exact recursive techniques to calculate the workspace dynamics.

There are several areas for future work. We are currently 100 to 200 times slower than interactive speeds and this hinders our ability to tune the controller quickly. We are currently creating a new multibody simulator to better handle the continuous, planar contact in our problem. We will also implement this controller on a 3D biped model. The move from SE(2) to SE(3) will involve more joints, but the controller block diagram remains the same. The 3x3 matrices in the SE(2) algorithm will be replaced by 6x6 matrices for the SE(3) algorithm. Accounting for errors in orientation is another complication of a 3D biped, but it can be dealt with in several ways. We also need to modify the existing controller or design a new controller to appropriately handle kinematic singularities. We then will begin to implement controllers to take steps, walk, run, carry loads, jump, and change direction.

Acknowledgements: We would like to thank Brian Mirtich for his help and for allowing us to use *Impulse*.

References

[1] J.K Hodgins, W.L. Wooten, D.C. Brogan, and J.F. O'Brien. Animating human athletics. In *Computer Graphics Proceedings. SIGGRAPH '95*, pages 71–78, New York, 1995. ACM Press.

[2] O. Khatib. A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE Journal of Robotics and Automation*, RA-3(1):43–53, 1987.

[3] B. Mirtich. *Impulse-based Simulation of Rigid Body Systems*. PhD thesis, U.C. Berkeley, 1996. To be filed in 1996.

[4] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*, New York, 1995. ACM Press.

[5] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research*, 2(1):13–30, 1983.

[6] K. W. Lilly. *Efficient Dynamic Simulation of Robotic Mechanisms*. Kluwer Academic Publishers, Boston, 1993.

[7] A. Jain. Unified formulation of dynamics for serial rigid multibody systems. *Journal of Guidance Control and Dynamics*, 14(3):531–542, May-June 1991.

[8] K. Kreutz-Delgado, A. Jain, and G. Rodriguez. Recursive formulation of operational space control. *International Journal of Robotics Research*, 11(4):320–8, 1992.

[9] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, Fl, 1994.

[10] F.C. Park, J.E. Bobrow, and S.R. Ploen. A Lie group formulation of robot dynamics. *The International Journal of Robotics Research*, 14(6):609–618, December 1995.