A Recursive Workspace Balancing Controller for a 3D Multibody Model of a Biped*

Jeffrey M. Wendlandt†

Intelligent Machines and Robotics Laboratory
University of California at Berkeley

Abstract

This paper describes a balancing controller for a 3D multibody model of a human biped. This model-based controller forms a model using a recursive formulation of workspace control. The recursive techniques free the control designer from generating the complicated equations of motion. The paper describes first the biped model and then the recursive workspace controller. Simulation results of the human model reacting to a disturbance are presented.

1 Introduction

The balancing controller described in this paper is part of a research effort to develop predictive models of human motion. These predictive models must appropriately react to disturbances and produce behaviors that mimic those of human beings. The approach used to develop predictive models is first to create a multibody model of a human biped and then to design control algorithms to command the system to balance, walk, run, jump, adapt to loads, and change directions. These predictive models require the development of efficient simulation techniques to solve numerically the system dynamics in contact with the environment, as well as the development of sophisticated control systems. The equations of motion of multibody systems are nonlinear, lengthy, complex, and can possess a large number of degrees of freedom for a robotic system. The control system must coordinate the degrees of freedom while handling the added complexities caused by the intermittent contact with the environment.

In this paper, the planar results in [1] are extended to create a workspace balancing controller for a 3D model of a biped. The kinematics and dynamic equations are more complex, and there are more degrees of freedom to control.

The paper first describes the biped model and then presents an overview of the recursive workspace controller. The workspace controller is then described in more detail followed by simulation results in a 3D multibody simulator called *Impulse* [2].

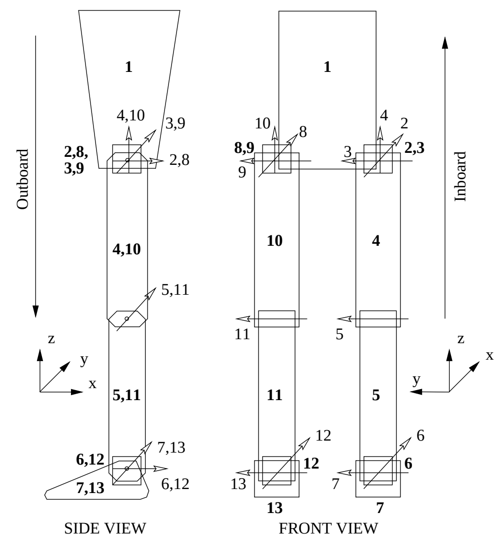

Figure 1: Biped Model

2 Description of Biped Model

The 3D multibody model of the biped consists of a central body and two legs and is shown in Figure 1. Each leg consists of an upper leg, a lower leg, and a foot. The leg is attached to the central body through a 3 degree of freedom (DOF) spherical joint consisting of two intermediate rigid bodies connected with revolute joints. The knee joint has one degree of freedom and connects the upper leg to the lower leg. The foot is attached to the lower leg through a 2 DOF joint consisting of one intermediate body connected with a revolute joint. The total number of rigid bodies in the system is 13 and the total number of DOF is 18. When both feet are in contact with the ground, the contact effectively decreases the total number of DOF to 6. The bodies are labeled in bold text. The revolute joints are shown with arrows pointing in the positive direction and are labeled with plain text. The left leg components are labeled starting at number 2 while the right leg components start at number 8.

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

†wents@eecs.berkeley.edu

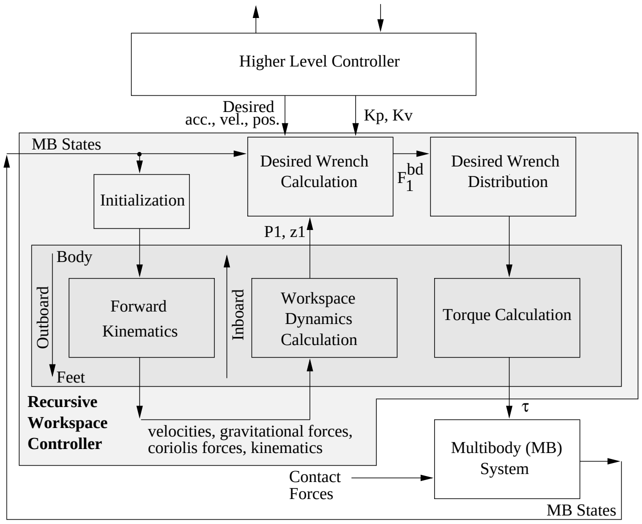

Figure 2: Controller Block Diagram

3 Recursive Workspace Controller

The block diagram for the recursive workspace controller is shown in Figure 2. The terms recursive and iterative refers to 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. Forces and torques due to Coriolis and gravitational forces are also calculated during this stage along with the forward kinematics.

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 [3] 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 [3], [4], [5], and [2]. The recursive workspace controller assumes that the feet are fixed to the floor. We approximate this condition from a *big base* assumption presented in [4]. The approximation assigns the feet a large mass and inertia (approximating the Earth) and sets the feet bias forces to zero.

3.1 Mathematical Background

In this section, a brief introduction to the notation used in this paper is presented. See [6] and [7] for more detail and development.

The configuration space of a rigid body is $SE(3)$, the special Euclidean group, formed from the semi-direct product of $\mathbb{R}^3$ and $SO(3)$ where $SO(3)$ is the space of rotation matrices. Consider a frame J and a frame K. The configuration of frame K with respect to frame J is represented by $g_{j,k} \in SE(3)$ where $g_{j,k}$ maps points and vectors in frame K to frame J. $g_{j,k}$ can be represented as a 4x4 matrix in the following form:

$$g_{j,k} = \begin{bmatrix} R_{j,k} & p_{j,k} \\ 0 & 1 \end{bmatrix}, \qquad (3.1)$$

where $p_{j,k} \in \mathbb{R}^3$ is the origin of frame K with respect to frame J and $R_{j,k} \in SO(3)$ is the orientation of frame K with respect to frame J. The inverse is given by

$$g_{j,k}^{-1} = \begin{bmatrix} R_{j,k}^T & -R_{j,k}^T p_{j,k} \\ 0 & 1 \end{bmatrix} \qquad (3.2)$$

$$= \begin{bmatrix} R_{k,j} & -R_{k,j} p_{j,k} \\ 0 & 1 \end{bmatrix} = g_{k,j}. \qquad (3.3)$$

The $\wedge$ operator takes a vector in $\mathbb{R}^3$ and converts it into a 3x3 skew symmetric matrix such that $w \times v = \hat{w}v$. The $\vee$ operator converts a 3x3 skew symmetric matrix to a vector in $\mathbb{R}^3$.

Consider frame K to be moving over time with respect to frame J such that $g_{j,k}$ is a function of time. A 4x4 twist matrix represents the body velocity of frame K with respect to frame J and is given by

$$\hat{V}_{j,k}^b = g_{j,k}^{-1} \dot{g}_{j,k} = \begin{bmatrix} \hat{\omega}_{j,k}^b & v_{j,k}^b \\ 0 & 0 \end{bmatrix}, \qquad (3.4)$$

and the twist vector for the body velocity is

$$V_{j,k}^b = \begin{bmatrix} v_{j,k}^b \\ w_{j,k}^b \end{bmatrix}, \qquad (3.5)$$

where $\omega_{j,k}^b$ is the body angular velocity of frame K with respect to frame J, and $v_{j,k}^b$ is the velocity of the origin of frame K with respect to frame J represented in frame K.

Algorithm 1 Controller Initialization

given: V_1^b, $g_{0,1}$;
$a_1^b = [0\ 0\ 0\ 0\ 0\ 0]^T$;
$b_1^b = -F_1^{be} + \Omega_1^b M_1^b V_1^b$;

Consider another frame L fixed with respect to frame K and let K and L move with respect to frame J. The body velocity of frame L and frame K with respect to frame J is given by

$$V_{j,l}^b = \mathrm{Ad}_{g_{l,k}} V_{j,k}^b, \tag{3.6}$$

where $\mathrm{Ad}_{g_{l,k}}$ is the 6x6 adjoint matrix and is given by

$$\mathrm{Ad}_{g_{j,k}} = \begin{bmatrix} R_{j,k} & \hat{p}_{j,k} R_{j,k} \\ 0 & R_{j,k} \end{bmatrix}. \tag{3.7}$$

Let frame K and frame L be attached to a rigid body. Let a wrench, a force / torque pair, act at the origin of frame K and call this wrench F_k. The wrench is a vector in $\mathbb{R}^6$ formed by stacking the force vector on top of the torque vector. The equivalent wrench acting at the origin of frame L and denoted F_l is given by the following formula:

$$F_l = \mathrm{Ad}_{g_{l,k}^{-1}}^T F_k. \tag{3.8}$$

3.2 Controller 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 provided in Algorithm 1. a_1^b is a kinematic acceleration term and is zero for the free joint between link 1 and the inertial frame. b_1^b consists of gravitational and Coriolis terms for link 1. The term, F_i^{be}, is a 6-vector of external forces acting at the origin of the body frame of link i; V_i^b is the 6-vector body velocity of link i with respect to link 0, the inertial frame;

$$M_i^b = \begin{bmatrix} m_i I & 0 \\ 0 & \mathbb{I}_i \end{bmatrix}; \ \Omega_i^b = \begin{bmatrix} \hat{\omega}_i^b & 0 \\ 0 & \hat{\omega}_i^b \end{bmatrix}; \tag{3.9}$$

m_i is the mass of link i; I is the 3x3 identity matrix; and $\mathbb{I}_i$ is the diagonal inertia matrix of link i. The external forces, F_i^{be}, in this system are the gravitational forces transformed to the body frame of link i.

3.3 Forward Kinematics

The body velocity of each link is now calculated in an outboard recursion. The velocity propagation stage calculates body velocities, gravitational forces, and Coriolis forces and is given in Algorithm 2. The map $g_{j,k} \in SE(3)$ takes coordinates of a point or vector in link k and gives the corresponding coordinates in link j. The symbol, $k.o$, gives the link index of the link outboard to joint k. The symbol, $k.i$, gives the link index of the link inboard to joint k. The adjoint appropriately transforms velocities and forces in different frames. The joint map is given by H_k and is a vector that represents the twist of joint k written in the link frame outboard to joint k, the frame of link $k.o$. The term $H_k\dot{\theta}_k$ is the relative body velocity between link $k.i$ and link $k.o$. The term $A_{k.i}^{k.o} V_{k.i}^b$ transforms the body velocity of link $k.i$ to link $k.o$ coordinates. The Coriolis terms are contained in a_i^b and in the last term of $b_{k.o}^b$. The first term in $b_{k.o}^b$ is the gravitational force written in the coordinates of link $k.o$.

Algorithm 2 Forward Kinematics

for $k = 2$ to 13 **do**
$g_{k.o,0} = g_{k.o,k.i}(\theta_k) g_{k.i,0}$
$A_{k.i}^{k.o} = \mathrm{Ad}_{g_{k.o,k.i}(\theta_k)}$
$V_{k.o}^b = A_{k.i}^{k.o} V_{k.i}^b + H_k \dot{\theta}_k$
$a_k^b = \dot{A}_{k.i}^{k.o} V_{k.i}^b + \dot{H}_k \dot{\theta}_k$
$b_{k.o}^b = -F_{k.o}^{be} + \Omega_{k.o}^b M_{k.o}^b V_{k.o}^b$
end for

Algorithm 3 Workspace Dynamics Calculation

for $k = 1$ to 13 **do**
$z_k = b_k^b$;
$P_k = M_k^b$;
end for
Apply fixed base approximation
$P_7 = P_{13} = 10^6 M_1^b$;
$z_7 = z_{13} = [0\ 0\ 0\ 0\ 0\ 0]^T$;
for $k = 13$ to 2 **do**
$D_k = H_k^T P_{k.o} H_k$;
$K_k = H_k (D_k)^{-1} H_k^T$;
$L_k = [I - K_k P_{k.o}] A_{k.i}^{k.o}$;
$P_{k.i} = P_{k.i} + L_k^T P_{k.o} L_k$;
$z_{k.i} = z_{k.i} + L_k^T (P_{k.o} a_k^b + z_{k.o})$;
end for

3.4 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 [3] 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 calculations are given in Algorithm 3.

The bias forces, z_k, and AB inertias, P_k, are first initialized as shown in Algorithm 3. The fixed base approximation is then applied to the two feet. The inboard recursion calculates the AB inertia for each link based on the AB inertia of the outboard link. The approximate workspace inertia is P_1, and the approximate Coriolis and gravitational forces are given in z_1. Given an external wrench, F_1, acting at the center of mass of the central body, the approximate workspace

Algorithm 4 Desired Wrench Calculation

given: trajectory data, MB states, workspace dynamics, and gains.
$e_c^p = p_{0,1}^s - p_{0,1}^{sd}$; $\dot{e}_c^p = \dot{p}_{0,1}^s - \dot{p}_{0,1}^{sd}$; $\{$*position error*$\}$
$X_c^s = \ddot{p}_{0,1}^{sd} - K_c^v \dot{e}_c^p - K_c^p e_c^p$;
$X_c^b = R_{0,1}^T (X_c^s - \omega_{0,1}^s \times p_{0,1}^s)$;
$q_{et} = q_{0,1} \star \bar{q}_{0,d} = (q_{et}^s, q_{et}^v)$; $\{$*quaternion error*$\}$
if $q_{et}^s \geq 0$ **then**
  $q_e = (q_e^s, q_e^v) = (q_{et}^s, q_{et}^v)$;
else
  $q_e = (q_e^s, q_e^v) = (-q_{et}^s, -q_{et}^v)$;
  $\{R_e \in SO(3)$ *is represented by* $\pm q_{et}\}$
end if
$\omega_e^s = \omega_{0,1}^s - R_e \omega_{0,d}^s$;
$X_R^s = R_e \dot{\omega}_{0,d}^s + \omega_{0,1}^s \times (R_e \omega_{0,d}^s) - K_R^v \omega_e^s - K_R^p q_e^v$;
$X_R^b = R_{0,1}^T X_c^s$;
$X^b = \begin{bmatrix} X_c^b \\ X_R^b \end{bmatrix}$;

$F_1^{bd} = P_1 X^b + z_1$;

dynamics are $F_1 = P_1 V_1^b + z_1$.

The approximate calculation of the workspace inertia is given in [4] based on the *big base* assumption. An exact calculation of the workspace dynamics for serial chains is given in [8] in terms of the spatial operator algebra.

3.5 Desired Wrench Calculation

In this section, the algorithm to calculate the desired wrench acting at the center of mass and written with respect to the body frame is given. The desired wrench is denoted by F_1^{bd} and is realized through the joint torques that are calculated in Section 3.7. The desired force is designed to be in the form

$$F_1^{bd} = P_1 X^b + z_1, \tag{3.10}$$

where X^b is calculated in Algorithm 4. The desired wrench calculation is divided into an orientation component (represented with unit quaternions) and a position component. Refer to [7] for more details.

3.6 Desired Wrench Distribution

In this section, the desired body wrench, F_1^{bd}, is divided into two contributions: one from the left leg and one from the right leg. Distributing the wrench needs to be done carefully to avoid creating unnecessary and detrimental internal forces and torques.

Let F_r be the wrench acting on the right foot from the ground and represented in the body frame of the right foot (link 13). Let F_l denote the corresponding wrench for the left foot (link 7). The desired body wrench is given by

Algorithm 5 Desired Wrench Distribution

given: $F_1^{bd}, g_{1,7}, g_{1,13}$
$G = [\mathrm{Ad}_{g_{1,7}^{-1}}^T \;\; \mathrm{Ad}_{g_{1,7}^{-1}}^T]$; $\{G$ *has full row rank*$\}$
$\begin{bmatrix} F_l \\ F_r \end{bmatrix} = G^T (GG^T)^{-1} F_1^{bd}$;

$\tilde{F}_2^{bd} = \mathrm{Ad}_{g_{1,7}^{-1}}^T F_l$;
$\tilde{F}_8^{bd} = \mathrm{Ad}_{g_{1,13}^{-1}}^T F_r$; $\{F_1^{bd} = \tilde{F}_2^{bd} + \tilde{F}_8^{bd}\}$

$$\begin{aligned} F_1^{bd} &= \mathrm{Ad}_{g_{1,7}^{-1}}^T F_l + \mathrm{Ad}_{g_{1,13}^{-1}}^T F_r \\ &= \begin{bmatrix} \mathrm{Ad}_{g_{1,7}^{-1}}^T & \mathrm{Ad}_{g_{1,13}^{-1}}^T \end{bmatrix} \begin{bmatrix} F_l \\ F_r \end{bmatrix} \triangleq G \begin{bmatrix} F_l \\ F_r \end{bmatrix} \end{aligned} \tag{3.11}$$

and is a result of transforming F_l and F_r to the frame of link 1 and adding. First note that the matrix G has full row rank and then solve Equation (3.11) for F_l and F_r by computing a minimum norm solution in the following way:

$$\begin{bmatrix} F_l \\ F_r \end{bmatrix} = G^T (GG^T)^{-1} F_1^{bd}. \tag{3.12}$$

Let $\tilde{F}_k^{bd}$ be the desired wrench acting on body $k.i$ from body $k.o$ and written with respect to frame $k.i$. Let F_k^{bd} be the desired wrench acting on link $k.o$ from link $k.i$ and written with respect to frame $k.o$. The wrenches acting on the central body, link 1, from the left and right legs are then

$$\tilde{F}_2^{bd} = \mathrm{Ad}_{g_{1,7}^{-1}}^T F_l \quad \text{and} \quad \tilde{F}_8^{bd} = \mathrm{Ad}_{g_{1,13}^{-1}}^T F_r. \tag{3.13}$$

In this way,

$$F_1^{bd} = \tilde{F}_2^{bd} + \tilde{F}_8^{bd}, \tag{3.14}$$

and the desired wrench is divided into a contribution from the left and right legs. The calculation is given and summarized in Algorithm 5.

3.7 Torque Calculation

The joint torques are now calculated in an outboard recursion given the wrench contributions from the left and right legs.

Let τ_l be the joint torques for the left leg and τ_r be the joint torques for the right leg. Let J_l be the body Jacobian for the left leg and J_r be the body Jacobian for the right leg. If the feet are fixed, J_l maps joint velocities for the left leg to the body velocity of link 1, and J_r maps right leg joint velocities to the body velocity of link 1. The joint torques are then $\tau_l = J_l^T \tilde{F}_2^{bd}$ and $\tau_r = J_r^T \tilde{F}_8^{bd}$. The Jacobian transpose relationship is a static equilibrium calculation and can be calculated by finding the equilibrium forces on each link as though

Algorithm 6 Torque Calculation

given: $\tilde{F}_2^{bd}$ and $\tilde{F}_8^{bd}$; $g_{k.o,k.i}$ and H_k for all $k \in \{2, \cdots, 13\}$;
for $k = 2$ to 13 **do**
  $F_k^{bd} = -\mathrm{Ad}^T_{g^{-1}_{k.o,k.i}} \tilde{F}_k^{bd}$; {*equal and opposite forces across joint* k}
  $\tau_k = H_k^T F_k^{bd}$; {*Joint Projection*}
  if link $k.o$ is not a leaf **then**
    $\tilde{F}_{k+1}^{bd} = -F_k^{bd}$; {*body* $k.o$ *is in equilibrium*}
  end if
end for

each link were at rest. The recursive algorithm to calculate the joint torques is given in Algorithm 6. The joint torques are calculated in an outboard iteration over the joints. First create F_k^{bd} by using equal and opposite forces across joint k:

$$F_k^{bd} = -\mathrm{Ad}^T_{g^{-1}_{k.o,k.i}} \tilde{F}_k^{bd}.$$

Then project F_k^{bd} across the joint axis to get the joint torque:

$$\tau_k = H_k^T F_k^{bd}.$$

Then update $\tilde{F}_{k+1}^{bd}$ by noting that link $k.o$ is in equilibrium for the Jacobian calculation:

$$\tilde{F}_{k+1}^{bd} = -F_k^{bd}.$$

4 Simulation Results

The controlled multibody system is created and simulated in *Impulse* [2], and simulation results are presented in this section. *Impulse* is a multibody simulator that handles contact through impulses. The controller is written in C and interfaced to the simulator. The coefficient of friction is 0.3, the coefficient of restitution is 0.2, and the gravitational acceleration is $9.81\frac{\mathrm{m}}{\mathrm{s}^2}$. The multibody system initially has zero velocity. The center of mass of the central body is initially placed at $(0, 0, 103.5)$ cm, and the central body is initially rotated about the $-y$ axis by 5 degrees. The system is initially slightly above the ground and collides with the ground right after the start of the simulation. The initial joint angles are $\{\theta_2, \cdots, \theta_{13}\} = \{0.0, 40.0, 0.0, -77.4595, 0.0, 42.4595, 0.0, 40.0, 0.0, -77.4595, 0.0, 42.4595\}$ degrees. The controller gains are

$$K_c^p = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 200 \end{bmatrix}, K_c^v = \begin{bmatrix} 7.07 & 0 & 0 \\ 0 & 7.07 & 0 \\ 0 & 0 & 14.142 \end{bmatrix},$$

$$K_R^p = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 50 \end{bmatrix}, \text{ and } K_R^v = \begin{bmatrix} 7.07 & 0 & 0 \\ 0 & 7.07 & 0 \\ 0 & 0 & 7.07 \end{bmatrix}.$$

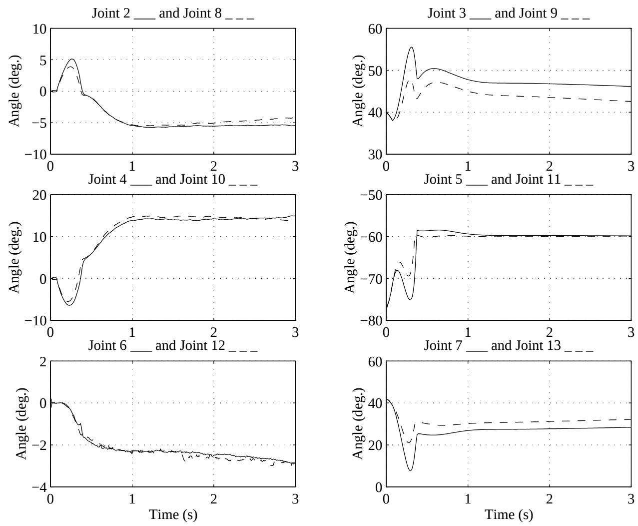

Figure 3: Joint Angles

The vertical gain in the Z position is chosen based on the experiments on astronauts in [9]. The remaining gains are chosen arbitrarily with a damping ratio of 0.5. The controller is called at 500 Hz. On an SGI Indigo (R4000, 100 MHz) workstation, each control loop takes 7 milliseconds to compute. A three second simulation took 2066 seconds (34 minutes 26 seconds) to compute.

The set point for the center of mass of link 1 is $(-1, 0, 114)$ cm, and the desired orientation is $q_{0,d} = (0.99144486, -0.13052619, 0, 0)$ corresponding to a 15 degree rotation about the $-Y$ axis. The system begins rising to the desired pose while a 1.8 Kg block is thrown at the body from the position of $(50, 10, 120)$ cm with a velocity of $(-450, 0, 20)$ cm/s (approximately 4 lbs. thrown at 10 mph). The block collides with the body at approximately 0.08 seconds. The two feet begin to rise at approximately 0.13 seconds, and the system rests on the front of the two feet as the heels rise. The right foot makes contact with the ground at approximately 0.36 seconds, and the left foot makes contact at approximately 0.38 seconds. During this simulation, the feet drift slowly relative to the ground. The drifting is believed to be an artifact of the contact model. The controller compensates for the collisions, the feet leaving contact with the ground, the impulsive contact forces, and the sliding feet to drive the error to a neighborhood of zero.

The joint angles over time are shown in Figure 3. The sharp change in joint angles due to the collision can be seen at 0.08 seconds. The feet making contact with the ground at 0.36 and 0.38 seconds is also seen in the data. After these disturbances, the joint angles settle with a small drifting due to the sliding feet.

The joint torques over time are shown in Figure 4. The torques fluctuate after the collision and have sharp

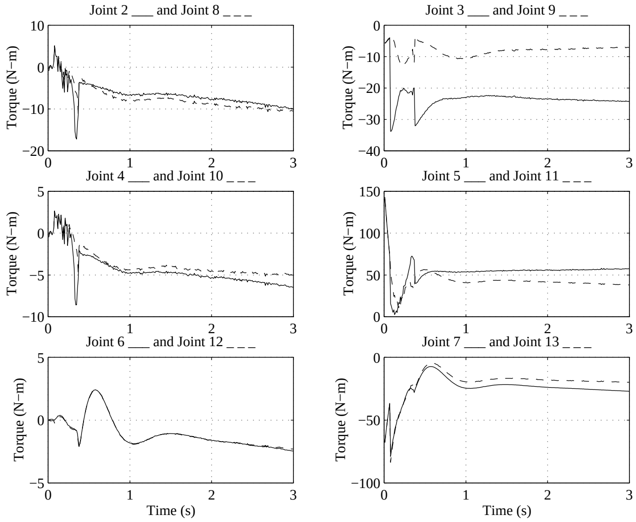

Figure 4: Joint Torques

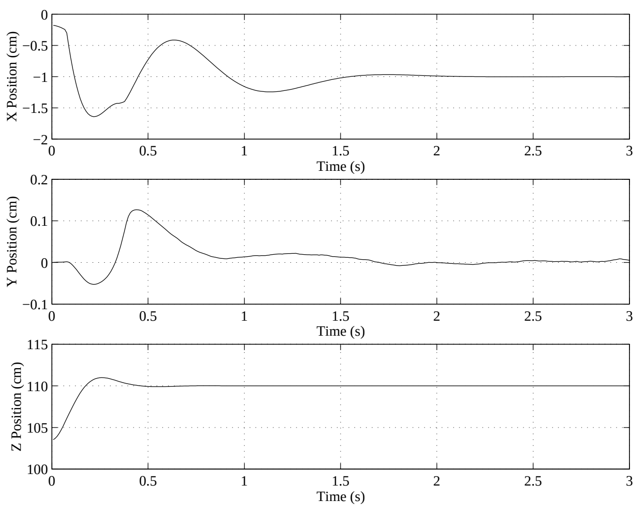

Figure 5: Body Position

changes after the changes in the contact of the feet with the ground.

The position of the center of mass relative to the inertial frame is shown in Figure 5. The errors converge to a neighborhood of zero. The convergence is faster in the Z coordinate. The Y position is seen to drift about zero, and this is believed to be due to the sliding feet. The collision disturbance is seen to push the body in the $-X$ direction and then the controller compensates for the disturbance.

The orientation of the central body is shown in Figure 6. The collision causes the body to rotate about the $-Y$ axis and to rotate about the Z axis as can be seen in the plot of q_y and q_z. The controller compensates for the disturbances and drives the error to a neighborhood of zero.

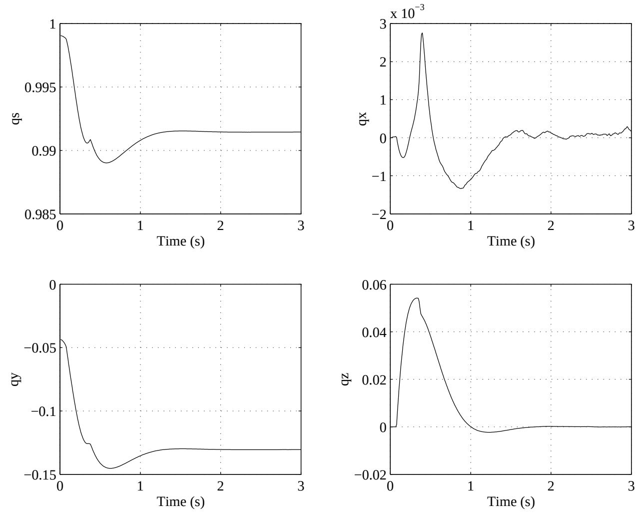

Figure 6: Body Orientation

5 Conclusion

The balancing controller extends the planar results in [1] to create a workspace balancing controller for a 3D multibody model of a biped. A model-based controller is formed in the workspace of the 3D biped, and the model is efficiently calculated through link to link iterations in the multibody structure. The recursive calculations free the control designer from having to symbolically create the complicated, nonlinear model. The research presented here brings us closer to the goals of our research effort to create predictive models for human motion.

References

[1] J.M. Wendlandt and S.S. Sastry. Recursive workspace control of multibody systems: A planar biped example. In *IEEE Control and Decision Conference*, Kobe, Japan, 1996. Dec. 11-13, 1996.

[2] B. Mirtich. *Impulse-based Simulation of Rigid Body Systems*. PhD thesis, U.C. Berkeley, 1996.

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

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

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

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

[7] J. M. Wendlandt. *Control and Simulation of Multibody Systems*. PhD thesis, University of California at Berkeley, 1997.

[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] D. J. Newman, D. K. Jackson, and J. J. Bloomberg. Altered astronaut performance in downward jumping following spaceflight-ii. mass center kinematics. submitted to Exp Brain Res, 1996.