2010 IEEE-RAS International Conference on Humanoid Robots
Nashville, TN, USA, December 6-8, 2010

Push Recovery by Stepping for Humanoid Robots with Force Controlled Joints

Benjamin J. Stephens, Christopher G. Atkeson

***Abstract*— In order to interact with human environments, humanoid robots require safe and compliant control which can be achieved through force-controlled joints. In this paper, full body step recovery control for robots with force-controlled joints is achieved by adding model-based feed-forward controls. Push Recovery Model Predictive Control (PR-MPC) is presented as a method for generating full-body step recovery motions after a large disturbance. Results are presented from experiments on the Sarcos Primus humanoid robot that uses hydraulic actuators instrumented with force feedback control.**

I. INTRODUCTION

Humanoid robots have the unique potential to operate in environments already designed for humans. While performing any given task in these complex environments, robots will encounter uneven ground, dynamic obstacles and humans. Force controlled robots, as opposed to stiff position controlled robots, can be compliant to unknown disturbances, resulting in safer and more robust operation. We have already presented a general framework for force control of legged balance with no stepping [1]. For small disturbances, standing balance is sufficient. However, for locomotion and large disturbances, the robot needs to step. The tight coupling between balance control and choice of footstep location makes this a challenging problem. This paper presents a method for controlling stepping in a force controlled robot.

While humanoid robots are very complex systems, the dynamics that govern balance are often described using simple models of the center of mass (COM) [2]. It has been shown through dynamic simulation that humanoid balance depends critically on controlling the linear and angular momentum of the system [3] [4], quantities that can be directly controlled by manipulating the contact forces.

Given a robot with stiff joint position control and a known environment, the most common approach to balance is to generate a stable trajectory of the COM and then track it using inverse kinematics (IK) [5]. These trajectories are often designed using model predictive control that optimizes over a receding horizon into the future [6] [7]. For unknown environments or small disturbances, the inverse kinematics can be modified to directly control the contact forces using force feedback [8]. However, this requires force-measurement at the point of contact, and the high impedance of the system limits the bandwidth at which it can comply.

For compliant robots with low impedance joints, there are a number of ways that contact force control can be achieved.

B. J. Stephens and C. G. Atkeson are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA. bstephens@cmu.edu, http://www.cs.cmu.edu/~bstephe1

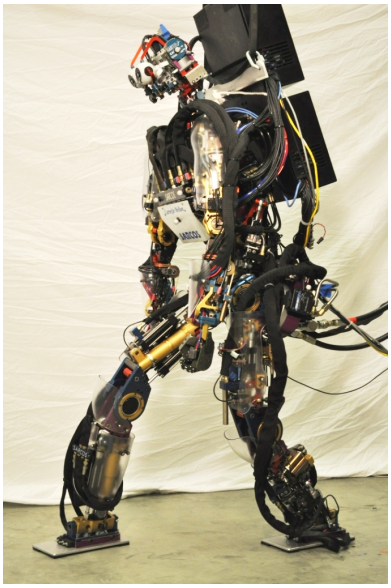

Fig. 1. The controller presented in this paper allows a humanoid robot to recover from large perturbation by stepping. It is applied to the Sarcos Primus hydraulic humanoid robot pictured.

Virtual model control (VMC) [9] is a simple method that only uses a kinematic model. Desired contact forces are converted into joint torques assuming static loading using a Jacobian-transpose mapping, i.e. $\tau = \mathbf{J}^T\mathbf{F}$. It has been shown that under quasistatic assumptions and proper damping of internal motions the desired forces can be achieved [10]. In contrast, given a full constrained rigid-body dynamics model, desired joint accelerations can be converted into joint torques using inverse dynamics for improved tracking performance [11].

Stepping strategies have been considered by several authors. Simple models have been used to define stable footstep locations, known as Capture Points [12]. Robots with stiff position control that expect small disturbances often solve footstep planning separately [13]. For situations when desired footstep locations cannot be known in advance, such as in the presence of large disturbances, motion and footstep planning can be performed simultaneously [14].

In this paper, the technique of planning stepping motions is extended to robots with force controlled joints, such as the Sarcos humanoid robot shown in Figure 1. Planning is performed by a linear model predictive controller called Push Recovery Model Predictive Control (PR-MPC). This is accomplished using a simple model presented in Section II and a carefully chosen objective function and constraints given in Section III. However, instead of tracking trajectories

as if the system had stiff position control, force control is used to add feed-forward joint torques to allow for the use of low gain joint trajectory tracking. The full body robot control is described in Section IV and some experimental results are shared in Section V.

II. COM DYNAMICS MODEL

Balance is controlled using a simple model of the COM dynamics. At any instant, the sum of all forces and torques on a system of rigid bodies results in an acceleration of the COM, $\ddot{\mathbf{C}}$, and a change in angular momentum, $\dot{\mathbf{H}}$. This relationship is linear in the contact forces and torques, $\mathbf{F}$,

$$\begin{bmatrix} \mathbf{D_1} \\ \mathbf{D_2} \end{bmatrix} \begin{pmatrix} \mathbf{F} \end{pmatrix} = \begin{pmatrix} m\ddot{\mathbf{C}} + \mathbf{F}_g \\ \dot{\mathbf{H}} \end{pmatrix} \tag{1}$$

where

$$\mathbf{D_1} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \tag{2}$$

$$\mathbf{D_2} = \begin{bmatrix} (\mathbf{P}_R - \mathbf{C})\times & (\mathbf{P}_L - \mathbf{C})\times & \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}, \tag{3}$$

and $\mathbf{F}$ can be partitioned to the right and left feet, respectively,

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_R \\ \mathbf{F}_L \\ \mathbf{M}_R \\ \mathbf{M}_L \end{pmatrix}. \tag{4}$$

$\mathbf{P}_R$ and $\mathbf{P}_L$ are the positions of the feet, $\mathbf{r}\times$ represents the left cross product matrix, m is the total mass of the system, and $\mathbf{F}_g$ is the constant gravitational force which points in the $-z$-direction. The first three equations of (1) sum the forces on the center of mass due to gravity and the ground contact points. The last three equations sum the torques about the center of mass to give the resulting change in angular momentum. Note that these equations can be extended easily to more than two contacts, but will be limited to two contacts in this paper. If $\dot{\mathbf{H}} = \mathbf{0}$, any forces that satisfy these equations do not generate angular momentum about the center of mass. If, additionally $\ddot{z} = 0$, the dynamics are identical to the well-known Linear Inverted Pendulum Model (LIPM) [15].

A. Linear Inverted Pendulum Model with External Force

The LIPM is often used to simplify planning and control for humanoid robots. This specially-chosen model is based on COM dynamics and assumes a constant height of the COM and zero angular momentum. In this paper, a slightly modified version is used which includes an external force, such as a push. The dynamics are given by

$$m\ddot{x} = \frac{mg}{z_0}(x - x_c) + f_x \tag{5}$$

$$m\ddot{y} = \frac{mg}{z_0}(y - y_c) + f_y \tag{6}$$

$$\dot{x_c} = u_x \tag{7}$$

$$\dot{y_c} = u_y \tag{8}$$

where (x, y) is the horizontal position of the COM at a constant height, z_0, (x_c, y_c) is the center of pressure (COP), (f_x, f_y) is an external force and (u_x, u_y) is the control signal for the center of pressure. This system can be re-written in discrete state-space form

$$\mathbf{X}_{t+1} = \mathbf{A}\mathbf{X}_t + \mathbf{B}\mathbf{U}_t \tag{9}$$

where $\mathbf{X}_t = (x_t, \dot{x}_t, x_{ct}, f_{Xt}, y_t, \dot{y}_t, y_{ct}, f_{Yt})^T$, $\mathbf{U}_t = (\dot{u}_{xt}, \dot{u}_{yt})^T$,

$$\mathbf{A}_t = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ \omega^2 T & 1 & -\omega^2 T & T/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega^2 T & 1 & -\omega^2 T & T/m \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

and

$$\mathbf{B}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{11}$$

$\omega^2 = g/z_0$ and T is the lookahead timestep.

Given a sequence of control inputs, $\bar{\mathbf{U}}$, the linear model in (9) can be converted into a sequence of states, $\bar{\mathbf{X}}$, for the next N timesteps,

$$\bar{\mathbf{X}} = \bar{\mathbf{A}}\mathbf{X}_t + \bar{\mathbf{B}}\bar{\mathbf{U}}, \tag{12}$$

where

$$\bar{\mathbf{X}} = \left(\mathbf{X}_{t+1}^T, \ldots, \mathbf{X}_{t+N}^T\right)^T, \tag{13}$$

$$\bar{\mathbf{U}} = \left(\mathbf{U}_{t}^T, \ldots, \mathbf{U}_{t+N-1}^T\right)^T, \tag{14}$$

and $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are defined recursively from (9).

The linear dynamics of the LIPM allow for some analytic insight into the behavior of the system. In particular, stability margins can be derived describing the states from which the system can recover without stepping [2], as shown in Figure 2. These analytic bounds are useful for deciding if and when a step is required.

III. STEP PLANNING

Stable walking patterns for walking robots are often generated using model predictive control, which optimizes a trajectory over a receding horizon. Using the LIPM, the trajectory optimization simplifies to a quadratic programming (QP) problem. In this section, Push Recovery Model Predictive Control (PR-MPC) is described using a special objective function and carefully-chosen constraints.

A. Objective Function

The goal of PR-MPC is to bring the COM to rest over the centroid of the support region with both feet on the ground. The footstep locations are made variable, allowing the controller to adaptively choose step locations for push recovery. The timings of the stepping phases, including both single and double support, are fixed ahead of time, as is the

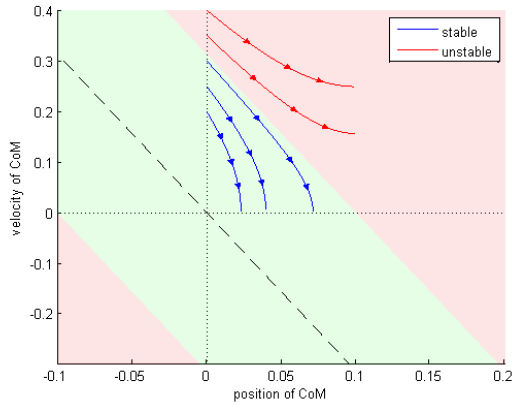

Fig. 2. For standing balance, the LIPM defines analytic stability regions in the COM position-velocity phase plane. Here the width of the support region is 20cm.

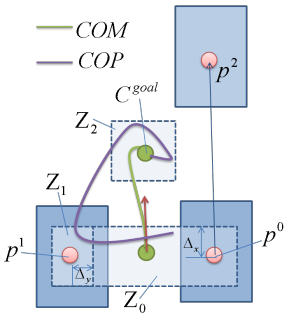

Fig. 3. Illustration of one step lookahead with right leg stepping.

number of steps to look ahead and the choice of initial swing foot.

The objective function used in this paper is

$$J = \frac{w_1}{2}\left|\left|\mathbf{C}_X^{\text{goal}} - \mathbf{C}_X\right|\right|^2 + \frac{w_2}{2}\left|\left|\dot{\mathbf{C}}_X\right|\right|^2 + \frac{w_3}{2}\left|\left|\mathbf{U}_X\right|\right|^2 + \frac{w_4}{2}\left|\left|p_X^{\text{ref}} - p_X\right|\right|^2 + \frac{w_1}{2}\left|\left|\mathbf{C}_Y^{\text{goal}} - \mathbf{C}_Y\right|\right|^2 + \frac{w_2}{2}\left|\left|\dot{\mathbf{C}}_Y\right|\right|^2 + \frac{w_3}{2}\left|\left|\mathbf{U}_Y\right|\right|^2 + \frac{w_4}{2}\left|\left|p_Y^{\text{ref}} - p_Y\right|\right|^2 \quad (15)$$

where $\mathbf{C}_X$ and $\mathbf{C}_Y$ are vectors of COM positions in 2D over the next N timesteps, $\left(\mathbf{C}_X^{\text{goal}}, \mathbf{C}_Y^{\text{goal}}\right)$ is the goal position, $\dot{\mathbf{C}}_X$ and $\dot{\mathbf{C}}_Y$ are the velocities, $\mathbf{U}_X$ and $\mathbf{U}_Y$ are the inputs, and p^{ref} and p are vectors of the next M reference and actual footstep locations, respectively. For the examples in this paper, it will be assumed that $M = 1$, but it is easily generalized to more footsteps. Reference footstep locations can be used to bias towards a desired step width or step length.

As shown in Figure 3, if p_X^0, p_X^1 and p_X^2 are the swing foot, stance foot and next footstep locations, respectively, then the COM goal position, C_X^{goal} is given by

$$\mathbf{C}_X^{\text{goal}} = \frac{\mathbf{O}_N}{2}\left(p_X^1 + p_X^2\right) \quad (16)$$

$$\mathbf{C}_Y^{\text{goal}} = \frac{\mathbf{O}_N}{2}\left(p_Y^1 + p_Y^2\right) \quad (17)$$

where $\mathbf{O}_N$ is a vector of ones N long. This means that the desired position is located half way between the location of the final footstep locations. The objective function is quadratic in the inputs: control input U over the next N timsteps and the next M footstep locations. Therefore, the dimensionality of the quadratic programming problem is $2(N + 2M)$.

B. Constraints

Preview control and stepping would be a simple problem if it were not for the constraints on the COP and footstep locations. This difficulty is compounded by not knowing the locations of the footsteps ahead of time, meaning the constraints on the COP after the first step are unknown, and can only be solved for exactly using nonlinear optimization. In this paper, simple conservative constraints are chosen to approximate the true constraints. The chosen constraints are conservative in that they more restrictive than the true constraints.

As shown in Figure 3, a COP constraint region, Z_i, is defined for each phase of the step. The first phase is assumed to be a short double support phase used to shift the weight over to the stance foot. The constraint region associated with this phase, Z_0, is known exactly because the current locations of the feet are known. Likewise, the constraint region during the swing phase, Z_1, is also known. However, the final constraint region, Z_2, is not known because the location of the footstep is not known in advance. A hand-crafted linear approximation of the final constraint region is shown. Notice also that the constraints assume smaller-than-actual-size feet. This is very important to help compensate for some of the unmodeled dynamics of the humanoid robot.

These constraints can be written in a compact form for use in standard quadratic programming software. Let $\mathbf{O}_i$ be defined as a vector of zeros and ones with the ones corresponding to the timesteps of the i-th phase, with the first phase being the initial double support phase. For a single footstep lookahead, i will range from 0 to 2. Notice that $\sum_i \mathbf{O}_i = \mathbf{O}_N$. The constraints on the X-trajectory of the COP, $\mathbf{X}_C$, can be written as

$$\mathbf{O}_0\left(\min\left(p_X^0, p_X^1\right) - \Delta_x\right) + \mathbf{O}_1\left(p_X^1 - \Delta_x\right) + \mathbf{O}_2\left(\mathbf{C}_X^{\text{goal}} - \Delta_x\right) \leq \mathbf{X}_C \quad (18)$$

$$-\mathbf{O}_0\left(\max\left(p_X^0, p_X^1\right) + \Delta_x\right) - \mathbf{O}_1\left(p_X^1 + \Delta_x\right) - \mathbf{O}_2\left(\mathbf{C}_X^{\text{goal}} + \Delta_x\right) \leq -\mathbf{X}_C \quad (19)$$

where Δ_x is the distance from the center of the foot to the edge in the x-direction.

C. Solving the QP

To solve for the step recovery trajectory, constrained quadratic programming is used to solve for the minimum of the objective function in (15) subject to the constraints in

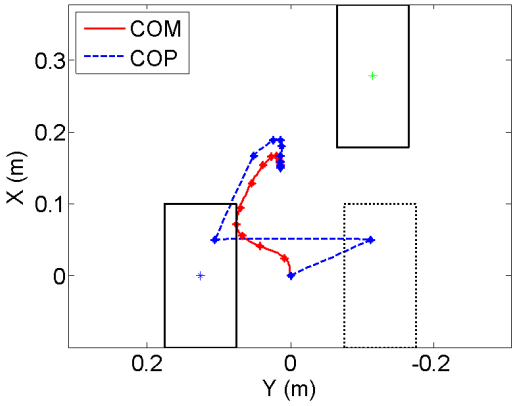

Fig. 4. Example trajectory solved using the PR-MPC with $T = 0.2$, $N = 15$. The system is perturbed from rest with both feet initially on the ground.

(18) and (19). It can be shown that when (12) is substituted into these equations, the result takes the form of

$$\bar{\mathbf{U}}^* = \arg\min_{\bar{\mathbf{U}}} \bar{\mathbf{U}}^T \mathbf{H} \bar{\mathbf{U}} + \mathbf{f}^T \bar{\mathbf{U}} \quad (20)$$

$$\text{s.t. } \mathbf{D}\bar{\mathbf{U}} \leq \mathbf{d} \quad (21)$$

which is a standard form and can be solved by many existing constrained QP solvers. In this paper, an active set dual method is used [16]. $\bar{\mathbf{U}}^*$ can then be substituted back into (12) to obtain the COM trajectory. The N points are connected using quintic splines to give continuous position, velocity and acceleration values along the trajectory. An example solution to this problem is shown in Figure 4 which resembles the expected behavior illustrated in Figure 3.

IV. ROBOT CONTROL

In this section, simplified model is directly applied to the state estimation and control of full body step recovery. A Kalman filter is designed using the linear dynamics model from (10) and (11). The control objective is simplified to tracking the desired COM trajectory and placing the feet at the desired footstep locations. If the robot had stiff position-controlled joints, this problem could be solved by inverse kinematics and joint trajectory tracking control. However, with compliant joints and unknown disturbances, feed-forward torques are required and dynamics play a larger role.

A. COM Kalman Filter

Accurate estimation of the COM state (position and velocity) is important for balance control. For better COM state estimation, the process model from (9) can be combined with a measurement model,

$$\mathbf{Y}_t = \begin{pmatrix} x \\ x_c \\ \ddot{x} \\ y \\ y_c \\ \ddot{y} \end{pmatrix} = \mathbf{C}\mathbf{X}_t \quad (22)$$

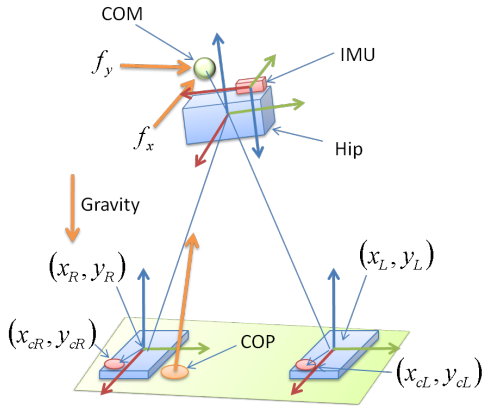

Fig. 5. A simplified model of the robot using only hip and feet states is used for state estimation of the humanoid robot.

where

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \omega^2 & 0 & -\omega^2 & 1/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \omega^2 & 0 & -\omega^2 & 1/m \end{bmatrix} \quad (23)$$

The COM position measurement is determined by the robot kinematics, the COP is sensed by force-torque sensors on the feet, and acceleration is measured by an IMU mounted at the hip, as shown in Figure 5. The COP is calculated by

$$x_c = \frac{(x_L + x_{cL})\, F_{ZL} + (x_R + x_{cR})\, F_{ZR}}{F_{ZL} + F_{ZR}} \quad (24)$$

$$y_c = \frac{(y_L + y_{cL})\, F_{ZL} + (y_R + y_{cR})\, F_{ZR}}{F_{ZL} + F_{ZR}} \quad (25)$$

where x_{cL} and x_{cR} are the position of the COP under each foot individually.

It is possible to implement a Kalman filter by combining this measurement model with the forward dynamics given by (10) and (11) and assuming noisy process and measurement models

$$\mathbf{X}_{t+1} = \mathbf{A}\mathbf{X}_t + \mathbf{B}\mathbf{U}_t + \mathbf{I}_{8\times 8}\mathbf{w} \quad (26)$$

$$\mathbf{Y}_t = \mathbf{C}\mathbf{X}_t + \mathbf{I}_{6\times 6}\mathbf{v} \quad (27)$$

where $\mathbf{w}$ and $\mathbf{v}$ represent the standard process and measurement noise variables which feature Gaussian noise characteristics with zero mean and covariances given by $\mathbf{Q}$ and $\mathbf{R}$, respectively.

B. Floating Body State Estimation

A floating body model of the robot is used for control. In order to use this model, it is important to have an estimate of the full state. Joint positions are measured directly and the hip orientation and angular velocity are sensed using an IMU. Joint velocities are calculated by differentiating the positions and filtering with 2nd order Butterworth filters. At all times, it is assumed that at least one foot is flat and fixed on the ground. Hip linear position and velocity are calculated with respect to this fixed foot. The procedure for calculating the state is:

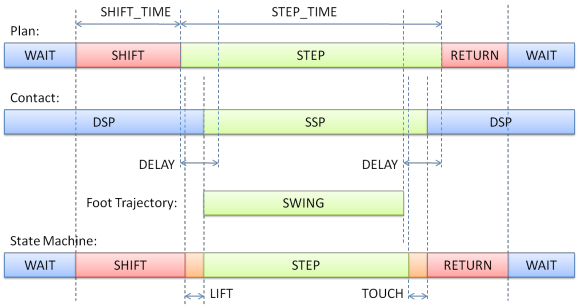

Fig. 6. The stepping state machine is chosen to handle errors in how well the robot follows the expected plan

1) Transform IMU measurements into the hip coordinate system.
2) Copy the joint positions, joint velocities, hip orientations and hip angular velocities into the state.
3) Assume the yaw orientation of the fixed foot is known and rotate the hip to correct for drift in the IMU.
4) Transform the IMU accelerations into the global coordinate system aligned with the fixed foot.
5) Calculate hip position and velocity relative to the fixed foot.
6) Calculate the COP from force sensors and feet positions.
7) Update the COM Kalman Filter.

C. Stepping State Machine

Special attention was given to the relationship between planning and execution of a stepping motion. The state machine is illustrated in Figure 6. Most significantly, it is assumed that the time spent in single support (SSP) is always less than the planned time. For liftoff (LIFT), this gives the robot extra time, if needed, in the SHIFT phase to continue shifting its weight towards the stance foot. For touchdown (TOUCH), this prevents the plan from asking for forces that cannot be applied before or just as the foot touches the ground.

Additionally, short transition, or "twilight", phases are added to trigger the switch between single and double support (DSP). By default, all phases are run off a clock. These transition phases can also be initiated through contact force detection. For example, if it is after the planned start of single support and the swing foot force goes to zero, the liftoff phase can be initiated early.

Multiple steps can be performed by re-evaluating the state at the end of the step and re-planning if necessary. In this case, the clock is reset and the state machine is re-initiated.

D. Inverse Kinematics

Given a planned trajectory of the COM and footstep locations, robot joint trajectories can be generated using inverse kinematics. The planner only specifies the final footstep locations, so footstep trajectories are defined by fifth order spline trajectories starting and ending with zero velocity and acceleration.

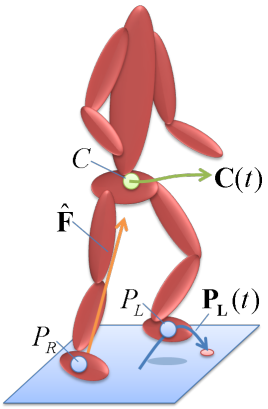

Fig. 7. Stepping control is achieved by tracking the desired COM and swing foot trajectories. Traditional joint tracking control is used with low gains with added feed-forward joint torques to achieve dynamic COM trajectory-tracking control.

A free-floating kinematic model is used to determine both desired joint angles and joint velocities. For full-body behaviors such as this, an optimization-based inverse kinematics solution is useful. In addition to COM and swing foot trajectories, the torso angle is desired to be vertical. A vector of feature velocities, $\dot{X}_f$, is related to the joint velocities, $\dot{q}$, by a full body Jacobian,

$$\dot{\mathbf{X}}_f = \mathbf{J}(q)\dot{\mathbf{q}} \tag{28}$$

At every timestep, gradient descent is performed to determine the joint velocities from the objective function

$$\dot{\mathbf{q}}^* = \arg\min_{\dot{\mathbf{q}}} \left|\left| \dot{\mathbf{X}}_f^{\text{des}} - \mathbf{J}(q)\dot{\mathbf{q}} \right|\right|^2. \tag{29}$$

The desired feature velocities, $\dot{\mathbf{X}}_f^{\text{des}}$, can be defined to prevent drift by

$$\dot{\mathbf{X}}_f^{\text{des}} = \dot{\mathbf{X}}_f^{\text{plan}} + \mathbf{K}_{pf}\left(\mathbf{X}_f^{\text{plan}} - \mathbf{X}_f^{\text{int}}\right) \tag{30}$$

where $\dot{\mathbf{X}}_f^{\text{plan}}$ and $\mathbf{X}_f^{\text{plan}}$ are determined by the plan and $\mathbf{X}_f^{\text{int}}$ is calculated from the robot state found by integrating $\dot{\mathbf{q}}^*$.

E. Feed-Forward Control

Feed-forward torques are generated using Dynamic Balance Force Control (DBFC) [1]. The step planner uses the current COM state, $(\mathbf{C}, \dot{\mathbf{C}})$ and foot locations, $\mathbf{P}_L$ and $\mathbf{P}_R$, to calculate step recovery trajectories for the COM, $\hat{\mathbf{X}}(t)$ and feet, $\hat{\mathbf{P}}_L(t)$ and $\hat{\mathbf{P}}_R(t)$, over the next M footsteps,

$$\left\{C, \dot{C}, P_L, P_R\right\} \rightarrow \{\mathbf{X}(t), \mathbf{P}_L(t), \mathbf{P}_R(t)\} \tag{31}$$

as illustrated in Figure 7.

The controller for this task is written as trajectory-tracking controller with feed-forward accelerations,

$$\ddot{X}_{\text{des}} = \ddot{\mathbf{X}}(t) + K_p\left(\mathbf{X}(t) - X\right) + K_d\left(\dot{\mathbf{X}}(t) - \dot{X}\right) \tag{32}$$

The desired contact forces, $\hat{F}$, are determined by solving (1) for a vector of contact forces that achieve the desired

acceleration in (32). Feed-forward joint torques can then be solved using DBFC, which can be re-written as

$$G \begin{pmatrix} \ddot{q} \\ \tau \end{pmatrix} = \begin{pmatrix} -N(q,\dot{q}) + J^T \hat{F} \\ \ddot{\mathbf{P}}(t) - \dot{J}\dot{q} \end{pmatrix} \quad (33)$$

where $\ddot{\mathbf{P}}(t)$ is non-zero for the swing leg. This set of equations is solved giving the desired feed-forward torques.

F. Hydraulic Actuator Torque Control

Hardware experiments are performed on a Sarcos humanoid robot. The robot uses linear hydraulic actuators with force feedback to perform compliant torque control on every joint [17]. Power is provided by an off-board pump with tethered hoses that connect to a manifold on the hip. There are potentiometers and force sensors at every joint, an inertial measurement unit (IMU) mounted on the hip, and 6-axis force sensors on each foot.

Each joint is controlled independently using a valve command that controls the rate at which hydraulic fluid enters and exits the piston chambers. This valve command is made up of two parts,

$$v_i = v_i^{fb} + v_i^{ff} \quad (34)$$

where v_i^{fb} is calculated locally on the robot at 5kHz and v_i^{ff} is calculated offboard and communicated to the robot at 400Hz. The local controller does simple linear load feedback on each joint,

$$v_i^{fb} = -K_{ui} u_i \quad (35)$$

while the offboard controller adds the necessary command that results in proportional control of the load. In addition, a positive velocity feedback term is added to cancel the actuator dynamics,

$$v_i^{ff} = K_{ui} u^{\text{des}} + K_{vi} \Psi_i \left(\dot{\theta}_i\right) \quad (36)$$

where the function $\Psi_i \left(\dot{\theta}_i\right)$ turns joint angular velocity into linear velocity of the piston. This function takes into account the kinematics of the joint and represents an instantaneous moment arm. Likewise, the desired load is calculated from the desired joint torque by a function, $u = \Phi(\tau)$, and a slow integrator that compensates for any valve bias

$$u_i^{\text{des}} = \Phi_i \left(\tau_i^{\text{des}} + K_{\tau i} \int \left(\tau_i^{\text{des}} - \Phi_i^{-1}(u_i) \right) dt \right) \quad (37)$$

Finally, the desired torque is a combination of the feedforward torques and low gain PD controls,

$$\tau_i^{\text{des}} = \tau_i^{\text{ff}} + K_{pi}(\theta_i^{\text{des}} - \theta_i) + K_{di}(\dot{\theta}_i^{\text{des}} - \dot{\theta}_i) \quad (38)$$

where τ_i^{ff} is calculated from (33).

V. RESULTS

Preliminary step recovery experiments have been performed on the Sarcos humanoid robot. The robot was pushed from behind with a force-measuring device to sense the magnitude of the push. Impulses are calculated by integrating the push force. Only large magnitude and short duration pushes, as opposed to low magnitude and long duration pushes, are presented here because they result in more dynamic motions. Figure 8 and Figure 9 show COM position and velocity and COP trajectories of a single push compared to the plan generated by PR-MPC. Figure 10 shows a top-down view of the COM and COP positions during the step.

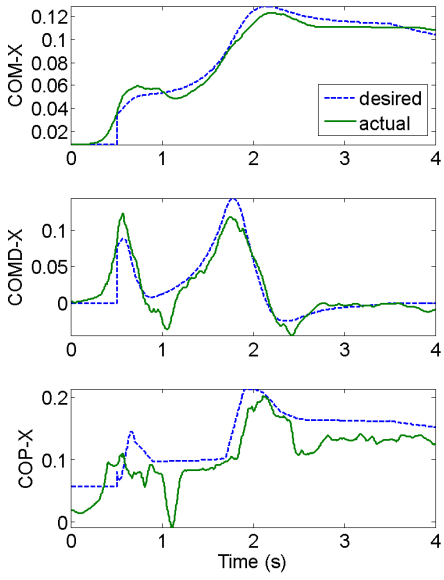

Fig. 8. X trajectories of the COM and COP from a push recovery experiment.

Many such push recovery experiments have been performed. Figure 11 shows 2 trials from forward impulses of approximately 23Ns. One trial exhibits a short step while the other exhibits a longer step. This apparent change in stepping strategy for similar-size pushes can be explained by Figure 12 which shows the forward step distance predicted by PR-MPC for a given initial X velocity and constant external force. As the initial velocity increases, there is a change in strategy requiring drastically larger steps. This is largely due to the activation of additional constraints on the COP during the step.

VI. DISCUSSION AND FUTURE WORK

Some extensions that are being investigated are multiple footstep lookahead and walking control. Currently, re-planning has only been initiated after each touchdown, but this could be done more often to more quickly correct for deviations from the planned trajectory due to additional disturbances during the step or large modeling error. Figure 14 shows the potential benefit of multiple footstep lookahead and re-planning. For a simulated robot, which does not necessarily have the same dynamics as the real robot, optimizing over multiple steps into the future allows for recovery from much larger disturbances.

There are several limits of the current algorithm. As presented, it does not directly address footstep rotations or cross-stepping (stepping to the side by crossing legs). In order to use the efficient quadratic programming solution of PR-MPC,

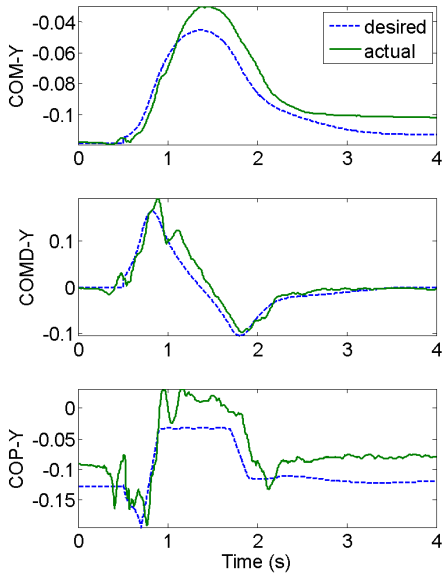

Fig. 9. Y trajectories of the COM and COP from a push recovery experiment.

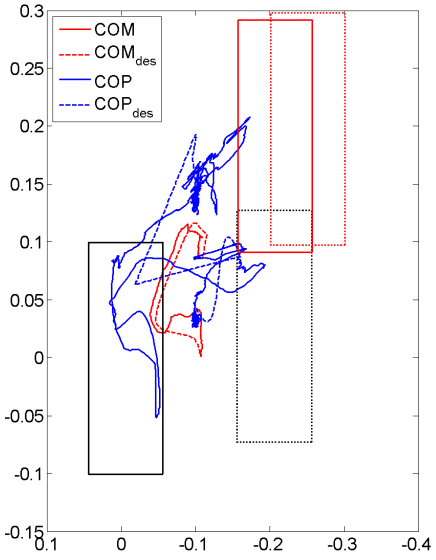

Fig. 10. Top-down view of the COM and COP trajectories and footsteps from a push recovery experiment running PR-MPC.

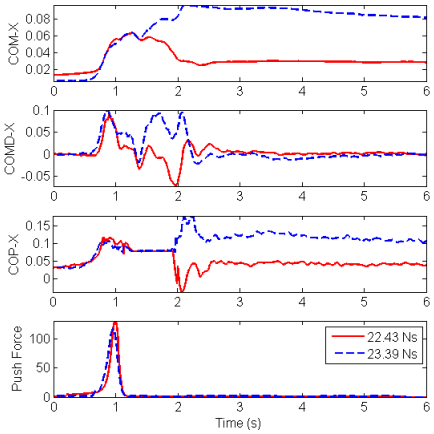

Fig. 11. X trajectories of COM including measured push forces from experiment. Impulses were calculated by integrating the push force curve.

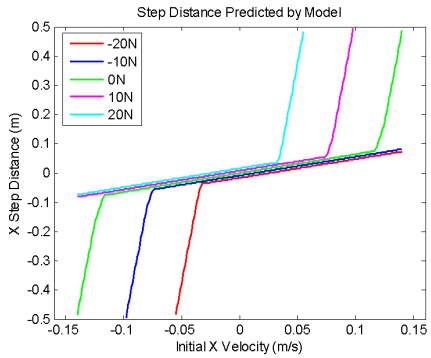

Fig. 12. Relationship between initial forward velocity, constant external force and forward step distance using PR-MPC reveals an apparent change in strategy.

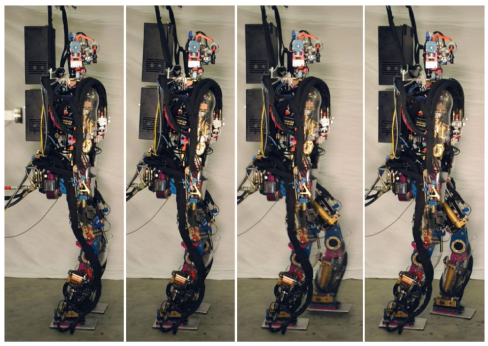

Fig. 13. Frames captured from a stepping experiment on the Sarcos Primus humanoid robot. See video at http://www.cs.cmu.edu/˜bstephe1/videos/humanoids2010.avi

Lookahead	Simulation (0.1s Impulse)	Robot (Measured Impulse)
No Step	18 Ns	22 Ns
1-step	21 Ns	23 Ns
1-step w/ replanning	29 Ns	N/A
2-step	36 Ns	N/A
2-step w/ replanning	42 Ns	N/A

Fig. 14. Comparison of multiple-step lookahead push recovery performance in simulation and on the robot. Currently only a single step lookahead is implemented on the robot.

all of the constraints must be linear. While the true double support constraints after touchdown are nonlinear, this paper only presents an approximation. Likewise, step timings are fixed in advance to maintain the compact quadratic problem structure. Nonlinear constraints can be solved using special QP algorithms [18] or sequential quadratic programming (SQP).

The computation speed of the algorithm has not been considered in this paper. The PR-MPC QP problem can be solved very efficiently by readily available algorithms and scales primarily with the number of lookahead timesteps, N. This should be as small as possible, but the lookahead time must also be large enough. In this paper, $N = 15$ with a timestep of $0.2s$ is used resulting in a lookahead of $3.0s$ and solution times just under 1 millisecond on a 2GHz CPU.

The role of angular momentum in step recovery is not discussed in this paper. One of the limitations of the LIPM is that it does not model angular momentum. To change this, the point mass of the LIPM can instead be modeled as a flywheel [12] [2], allowing a torque about the COM. Using this model in PR-MPC requires only minor changes.

VII. CONCLUSION

This paper presented a stepping controller suitable for a robot with compliant joints. Using low gain PD joint tracking controls combined with feedforward torques generated by Dynamic Balance Force Control (DBFC), the robot can track a desired COM trajectory generated by Push Recovery Model Predictive Control (PR-MPC). This allows the robot to comply to and recover from a large push. Results were presented as pushing experiments on the Sarcos humanoid robot.

VIII. ACKNOWLEDGEMENTS

This material is based upon work supported in part by the US National Science Foundation under grants ECCS-0325383, DGE-0333420, EEC-0540865, and ECCS-0824077.

REFERENCES

[1] B. J. Stephens and C. G. Atkeson, "Dynamic Balance Force Control for Compliant Humanoid Robots," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2010.

[2] B. J. Stephens, "Humanoid Push Recovery," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2007.

[3] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 1, 2009.

[4] Y. Abe, C. K. Liu, and Z. Popovic, "Momentum-based Parameterization of Dynamic Character Motion," *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pp. 194–211, 2004.

[5] A. Takanishi, T. Takeya, H. Karaki, and I. Kato, "A control method for dynamic biped walking under unknown external force," in *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, vol. 29, no. 6. IEEE, 1990, pp. 795–801.

[6] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *Proceedings fo the International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 2003, pp. 1620–1626.

[7] P.-B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," in *Proceedings of the International Conference on Humanoid Robots*, vol. 25, no. 4. IEEE, 2006, pp. 137–142.

[8] Y. Fujimoto and A. Kawamura, "Proposal of biped walking control based on robust hybrid position/force control," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, no. 2. IEEE, 1996, pp. 2724–2730.

[9] J. Pratt and G. Pratt, "Intuitive control of a planar bipedal walking robot," in *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998, pp. 2014–2021.

[10] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-Body Compliant Human Humanoid Interaction: Balancing in the Presence of Unknown External Forces," *IEEE Transactions on Robotics*, vol. 23, pp. 884–898, Oct. 2007.

[11] M. Mistry, J. Buchli, and S. Schaal, "Inverse Dynamics Control of Floating Base Systems using Orthogonal Decomposition," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[12] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Proceedings of the International Conference on Humanoid Robots*. IEEE, Dec. 2006, pp. 200–207.

[13] J. Chestnutt, M. Lau, K. M. Cheung, J. Kuffner, J. K. Hodgins, and T. Kanade, "Footstep Planning for the Honda ASIMO Humanoid," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr. 2005.

[14] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online Walking Gait Generation with Adaptive Foot Positioning Through Linear Model Predictive Control," in *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1121–1126.

[15] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation andapplication of the linear inverted pendulum mode," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, Apr. 1991, pp. 1405–1411.

[16] D. Goldfarb and a. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, Sep. 1983.

[17] D. C. Bentivegna and C. G. Atkeson, "Compliant control of a hydraulic humanoid joint," *Proceedings of the International Conference on Humanoid Robots*, pp. 483–489, Nov. 2007.

[18] D. Dimitrov, P.-B. Wieber, H. J. Ferreau, and M. Diehl, "On the implementation of model predictive control for on-line walking pattern generation," *Proceedings of the International Conference on Robotics and Automation*, pp. 2685–2690, May 2008.