Torso Rotation for Push Recovery Using A Simple Change of Variables

Eric C. Whitman, Benjamin J. Stephens, and Christopher G. Atkeson

***Abstract*— This paper presents a modification for a broad class of controllers based on the LIPM dynamics. We use a change of variables such that instead of controlling the center of mass, we control an "augmented center of mass", which is unaffected by upper body angular accelerations. We use upper body orientation as an additional source of control authority, allowing us to use both upper body rotation and center of pressure modulation for control. We demonstrate an improved robustness to external pushes with this additional control authority through simulated standing and walking experiments. We also demonstrate the modified controller on our force-controlled humanoid robot.**

I. INTRODUCTION

Many researchers design walking controllers based on the Linear Inverted Pendulum Model (LIPM). These dynamics require two assumptions: 1) the center of mass remain at a constant height, and 2) angular momentum is constant. Overcoming the limitations of this second assumption is the subject of this paper. In this paper, we discuss a new method for exploiting angular momentum by applying a simple change of variables. The controller for this simplified model is then mapped to a humanoid, as shown in Fig. 1, using full body torque control, and shown to improve balance.

Many methods that control the LIPM, such as preview control [1] and some forms of model predictive control [2], focus on control of the zero moment point (ZMP). For many of these methods, it is either impossible or computationally expensive to extend them to work with a model that considers angular momentum or upper body rotation.

Due to disturbances and un-modeled dynamics, angular momentum and posture regulation are required, which can directly interfere with a controller based solely on the LIPM. Several authors have derived models of angular momentum for biped robots [1] [3] and it has been shown that exploiting angular momentum can add significant stability to the system [4] [5]. The subject of upper body angular momentum coordination and control for locomotion in position controlled humanoid robots has been considered [6] [7].

Full body torque controllers based on force-based objectives such as desired COM acceleration and change of angular momentum have been presented [8] [9] [10] [11]. Controllers such as these have achieved hip-strategy-like behaviors by making the angular momentum or posture objective less important than COM regulation. However, angular momentum and posture objectives have been mostly limited to regulation tasks.

Authors are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA. ewhitman@cmu.edu

We have identified a mode of motion that rotates the upper body and translates the center of mass without moving the ZMP. We have also identified a point, which we refer to as the augmented center of mass, which is unaffected by these motions, but follows LIPM-like dynamics. By performing a simple change of variables, we can use this point (instead of the center of mass) with traditional ZMP-based algorithms. The result is that the identified mode functions as an additional source of control authority that looks mathematically similar to moving the ZMP. This extra control authority can be utilized by traditional ZMP-based control schemes with no modification to the core algorithm for planning more aggressive maneuvers or greater stabilization.

In this paper, the coupling between COM and angular momentum dynamics is exploited using a simple model and a change of variables. The resulting desired COM and torso accelerations are used to generate full body torques. The unique feature of the method presented here is that it can be added on to existing systems using LIPM dynamics, allowing them to use upper body rotation to improve robustness and allow more aggressive maneuvers. For example, in our previous work [12], we used dynamic programming for control of the LIPM dynamics for walking. Dynamic programming has several advantages, but it is computationally expensive to add the extra state space dimensions that would be necessary to control upper body rotation. This method allows us to utilize upper body rotations for balance without increasing the dimensionality of the dynamic programming.

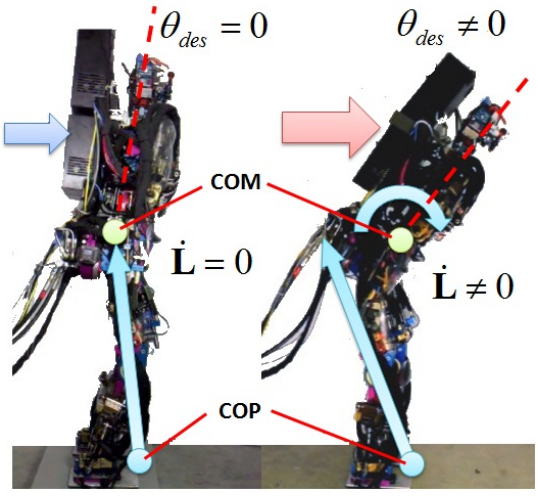

Fig. 1. The system presented in this paper provides a systematic method for exploiting angular momentum for balance by changing the desired torso angle.

The paper is outlined as follows. In Section II, we derive the model from the LIPM dynamics by relaxing the constraint on zero change of angular momentum and then describe our change of variables strategy. Section III describes the details of how this system can be applied practically. Results from both simulation and hardware are presented in Section IV. Finally, a discussion is offered in Section V.

II. SYSTEM MODEL

Many researchers design walking controllers based on the Linear Inverted Pendulum Model (LIPM), given by

$$\ddot{x} = \frac{g}{h}(x - z) \tag{1}$$

where x is the position of the COM relative to center of the foot, h is a fixed height of the COM above the ground, m is the total mass, and z is the location of the center of pressure (COP) relative to the center of the foot. These dynamics require two assumptions: 1) the center of mass remain at a constant height, and 2) that there be no change in angular momentum about the COM, $\dot{\mathbf{L}} = 0$. The latter assumption is the focus of this paper.

It is often assumed that the majority of the mass is in the torso, and that we can approximately enforce $\dot{\mathbf{L}} = 0$ by maintaining a fixed torso orientation. However, it may not be possible to perfectly control the torso orientation. Undesired torso rotations may be caused by error in the model of the robot or the environment as well as by external pushes. These rotations can be particularly large in torque-controlled systems or systems with relatively low position gains. Any angular acceleration of the torso (or more generally, any change in angular momentum) without changing the COM acceleration will require some foot torque, which means some change in the COP. Conversely, if the COP of the robot is matched to that of a LIPM model, but there is a change in angular momentum, then the COM motion of the robot will not match that predicted by the LIPM model.

In order to model the interaction between upper body rotation and COM motion, we model the upper body as a single rigid flywheel, which rotates about the system center of mass. This model is known as the Linear Inverted Pendulum plus Flywheel Model (LIPFM) [4] with dynamics given by

$$\ddot{x} = \frac{g}{h}(x - z) - \frac{I\ddot{\theta}}{mh} \tag{2}$$

where I is the moment of inertia of the upper body about the system COM and θ is the angle of the upper body relative to the nominal. The upper body angle will also be subject to the kinematic constraint $\theta_{\min} \leq \theta \leq \theta_{\max}$. The LIPFM does not fully account for angular momentum, but many humanoid systems have a large fraction of their mass in the upper body (about 2/3 of the total mass in our system), so a large fraction of the angular momentum will be in the upper body. Additionally, while the lower body motion is highly constrained by walking, we are free to rotate the upper body as desired to aid in control.

We now perform a change of variables so that upper body rotation does not require any change in the COP. Additionally, it will put the LIPFM dynamics in the same form as the LIPM dynamics, allowing us to leverage the existing technology for controlling the LIPM.

A. Change of Variables

We start with the forward (rotational and translational) dynamics for the x-z plane,

$$\begin{bmatrix} m\ddot{x} \\ I\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -h & mg \end{bmatrix} \begin{bmatrix} F_x \\ -z \end{bmatrix} + mg \begin{bmatrix} 0 \\ x \end{bmatrix}, \tag{3}$$

where F_x is the horizontal ground reaction force. We then multiply by a change of variables matrix,

$$\mathbf{D} \begin{bmatrix} m\ddot{x} \\ I\ddot{\theta} \end{bmatrix} = \mathbf{D} \begin{bmatrix} 1 & 0 \\ -h & mg \end{bmatrix} \begin{bmatrix} F_x \\ -z \end{bmatrix} + \mathbf{D}mg \begin{bmatrix} 0 \\ x \end{bmatrix}. \tag{4}$$

where

$$\mathbf{D} = \begin{bmatrix} 1 & \frac{1}{h} \\ 0 & 1 \end{bmatrix}. \tag{5}$$

We now define the augmented COM acceleration, $\ddot{\tilde{x}} \equiv \ddot{x} + \frac{I\ddot{\theta}}{mh}$, and then combine and simplify to get

$$\begin{bmatrix} m\ddot{\tilde{x}} \\ I\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & \frac{mg}{h} \\ -h & mg \end{bmatrix} \begin{bmatrix} F_x \\ -z \end{bmatrix} + \begin{bmatrix} \frac{mg}{h}x \\ mgx \end{bmatrix}. \tag{6}$$

Double integrating $\ddot{\tilde{x}}$ gives us the augmented COM position,

$$\tilde{x} = x + \frac{I\theta}{mh}, \tag{7}$$

which we substitute into (6) to get

$$\begin{bmatrix} m\ddot{\tilde{x}} \\ I\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & \frac{mg}{h} \\ -h & mg \end{bmatrix} \begin{bmatrix} F_x \\ -z \end{bmatrix} + \begin{bmatrix} \frac{mg}{h}\tilde{x} + \frac{gI\theta}{h^2} \\ mg\tilde{x} + \frac{gI\theta}{h} \end{bmatrix}. \tag{8}$$

Multiplying by the inverse of the 2x2 matrix and solving for the controls, F_x and z, gives the inverse dynamics,

$$\begin{bmatrix} F_x \\ -z \end{bmatrix} = \begin{bmatrix} 1 & \frac{-1}{h} \\ \frac{h}{mg} & 0 \end{bmatrix} \begin{bmatrix} m\ddot{\tilde{x}} \\ I\ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{x} + \frac{I\theta}{mh} \end{bmatrix}. \tag{9}$$

The benefit of the change of variables is that we now have a 0 in the lower right of the 2x2 matrix. This 0 means that changes in angular momentum without modifying $\ddot{\tilde{x}}$ have no effect on the COP.

If we define an augmented COP,

$$\tilde{z} = z + \frac{I\theta}{mh}, \tag{10}$$

we are able to put the LIPFM dynamics,

$$\ddot{\tilde{x}} = \frac{g}{h}(\tilde{x} - \tilde{z}), \tag{11}$$

in the same form as the original LIPM dynamics in (1).

Since (1) and (11) have the same form, control methods that work on the LIPM dynamics can also be used on the augmented LIPM dynamics, but by working in the space of $\tilde{x} = x + I\theta/mh$ and its derivatives instead of the space of x and its derivatives. Additionally, controllers will now

be requesting a $\tilde{z}$, and (10) must be used to find z before applying control to the system. The change of variables eliminated a $\ddot{\theta}$ term (by hiding it within $\ddot{\tilde{x}}$), but created a θ term. The angular position term is preferable from a modeling perspective because it changes more slowly and can be more easily measured.

B. Orientation Control

Since θ can be controlled independently of $\tilde{x}$, the θ term in $\tilde{z}$ can be used for more than canceling the effect of small undesired deviations from the desired posture. It can also be actively controlled to something other than 0 and treated as an additional control for the augmented COM. Torso orientation, θ, has the same effect on the system as does z, so a LIPM controller need not differentiate between them; instead, it can simply request a $\tilde{z}$. In addition, the controller can operate under a more lenient constraint (but with the same form) than the ordinary COP constraint, $|z| \leq z_{\max}$. For the LIPFM dynamics, the constraint is

$$|\tilde{z}| \leq z_{\max} + \frac{I\theta_{\max}}{mh} \tag{12}$$

where $\theta_{\max}$ is the maximum allowable lean angle. Use of the more lenient constraint (and the torso rotation necessary to produce the larger requested $\tilde{z}$) makes it possible to plan more aggressive maneuvers that are impossible with a fixed-orientation torso. It also allows for improved feedback in response to unexpected disturbances.

There is also an additional constraint arising from the fact that while contact forces such as z can be changed quickly, torso angle will have a maximum velocity or acceleration. Such constraints become an issue if the requested $\tilde{z}$ rapidly changes. In situations where $\tilde{z}$ changes gradually or on systems where the internal dynamics (dependent on actuators) are much faster than the LIPM dynamics (dependent on g/h), this additional constraint can reasonably be ignored.

There are several options for controlling θ depending on the context. If the increased control authority is being used in a planning context for generating trajectories with large accelerations, then the planner will produce a time varying $\tilde{z}(t)$. In this case, optimization (for example, a quadratic program) can be used to determine a time varying $\theta(t)$ which satisfies (12) as well as whatever hardware constraints exist on $\dot{\theta}$ or $\ddot{\theta}$. A cost on $\theta(t)$ can be added to the optimization to bias it towards upright postures that do not utilize torso rotation wherever possible. However, if (as in the results presented in this paper) the increased control authority is being used for improved responses to unexpected disturbances, then no expected $\tilde{z}(t)$ will be available (because the disturbances are unexpected).

In this case, we use a fixed relationship between $\tilde{z}$ and the desired torso orientation, θ_{des}, then use a high gain PD servo to track that desired angle. Immediately following a large disturbance, it will not be possible to obtain the requested $\tilde{z}$ until the torso has had time to slew, resulting in (11) being inaccurate. However, slewing happens relatively quickly, and rotating the torso at the maximum speed is the best thing to do in this situation. Therefore, the quality of our control does not suffer from neglecting this effect.

We have used a piecewise linear relationship with a dead zone to determine θ_{des} from $\tilde{z}$.

$$\theta_{des}(\tilde{z}) = \begin{cases} \text{if } \tilde{z} > z_{\max}/2 & : \quad (\tilde{z} - z_{\max}/2)mh/I \\ \text{if } \tilde{z} < -z_{\max}/2 & : \quad (\tilde{z} + z_{\max}/2)mh/I \\ \text{else} & : \quad 0 \end{cases} \tag{13}$$

The purpose of the dead zone is to prevent the upper body from rotating unnecessarily, which would both look unnatural and run the risk of exciting unmodeled dynamic modes. For ordinary small values of $\tilde{z}$, θ_{des} will remain at 0, and torso rotation will only be utilized when it becomes necessary to achieve large $\tilde{z}$'s. In simulation, the exact shape of this relationship has very little effect on the system robustness. However, smoother functions perform better on real hardware due to decreased jerk and less excitation of un-modeled higher-order dynamics.

III. USAGE

The intended use of the system presented in this paper is to be added on to existing systems for planning or control of the LIPM dynamics. Many existing systems use the LIPM dynamics for generation and control of COM motion, but do not consider upper body rotation or angular momentum. The use of upper body rotation can improve robustness and allow for more aggressive maneuvers, but for many existing methods, planning or control in the larger space of translations and rotations is difficult. We have developed a means of gaining much of the benefit of upper body rotations without increasing the complexity of the primary algorithm. We first explain how it could be applied to open-loop pattern generation and then torque feedback control.

A. Pattern Generation

Fig. 2 shows how you would use this method for augmenting a simple generic motion planning system for a position controlled robot. The upper flow chart depicts the unmodified system, which uses a planner (e.g. preview control) to generate COM and COP trajectories. Then, an inverse kinematics solver uses the COM trajectory and a fixed upper body orientation, $\theta(t) = 0$, to generate joint angle trajectories, which are used to drive the robot.

The lower flow chart shows how this system would be modified to use the LIPFM dynamics as described in this paper. The same planner and inverse kinematics solver can be used without modification, but the interaction between the two is modified. First, the planner is given the more lenient constraint given by (12). Now, the output of the planner is interpreted as $\tilde{x}(t)$ and $\tilde{z}(t)$. A simple trajectory optimization (denoted "OPT" in Fig. 2) is then used to break $\tilde{z}(t)$ into $z(t)$ and $\theta(t)$. The $\theta(t)$ trajectory is then used in place of the fixed torso orientation given to the inverse kinematics solver. It is also used to compute $x(t)$ from $\tilde{x}(t)$. The inverse kinematics solver then determines joint angle trajectories for driving the robot.

The trajectory optimization for $\theta(t)$ can be accomplished by a quadratic program using a time sequence of angles, $\theta(t_i)$, as the optimization variables. Actuator limits can be enforced using constraints such as $|(\theta(t_i) - \theta(t_{i-1}))/T| \leq \dot{\theta}_{\max}$ for angular velocity limits and $|(\theta(t_i) - 2\theta(t_{i-1}) + \theta(t_{i-2}))/T^2| \leq \ddot{\theta}_{\max}$ for angular acceleration limits. An additional constraint must be added for each time step to enforce (12). Minimizing a simple cost function such as $C = \sum_i (\theta(t_i))^2$ is adequate, though additional terms can be added for a more specific desired behavior.

The change of variables allows the planner to look at the extra capability given by upper body rotation as simply a larger virtual foot. Then, we post-process its output to achieve the extra control authority it requests through upper body rotation.

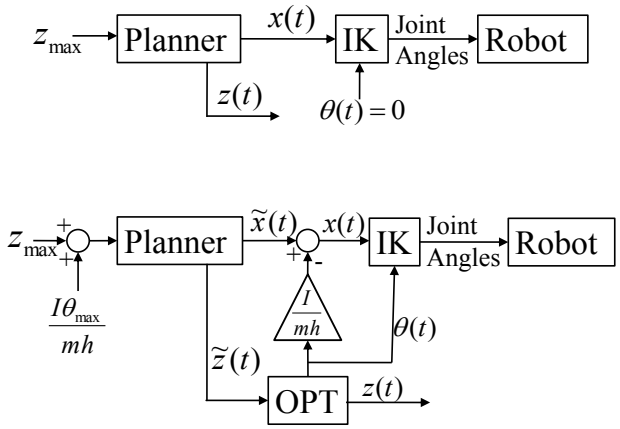

Fig. 2. A flow chart of a simple generic system for humanoid walking pattern generation using standard LIPM dynamics (top) and a flow chart for that same system modified to use the LIPFM dynamics (bottom). "IK" represents inverse kinematics, and "OPT" represents the trajectory optimization discussed above.

B. Torque Control

We use an analogous process to modify a LIPM-based controller for a torque controlled robot, as shown in Fig. 3. The upper flow chart depicts a generic LIPM-based controller. A high-level controller (e.g. MPC or dynamic programming), denoted "HLcon" in Fig. 3, uses the current COM state, x and $\dot{x}$ to determine the desired COP, z. A PD servo is used to keep the torso angle at a fixed orientation, and a low-level controller, denoted "LLcon" in fig. Fig. 3, generates the joint torques necessary to achieve the desired z and $\ddot{\theta}$.

The lower flow chart shows how this system was modified to use the LIPFM dynamics. Again, the primary algorithms, the high-level and low-level controllers, remain unchanged. As for pattern generation, we modify the COP constraint according to (12). The inputs to the high level controller are now the augmented COM state, $\tilde{x}$ and $\dot{\tilde{x}}$, and the output is now interpreted as $\tilde{z}$. We use θ and $\tilde{z}$ to determine z by rearranging (10), which is passed to the low-level controller. The bottom box represents the function given in (13), and is used to pick a θ_{des} that makes $\tilde{z}$ achievable.

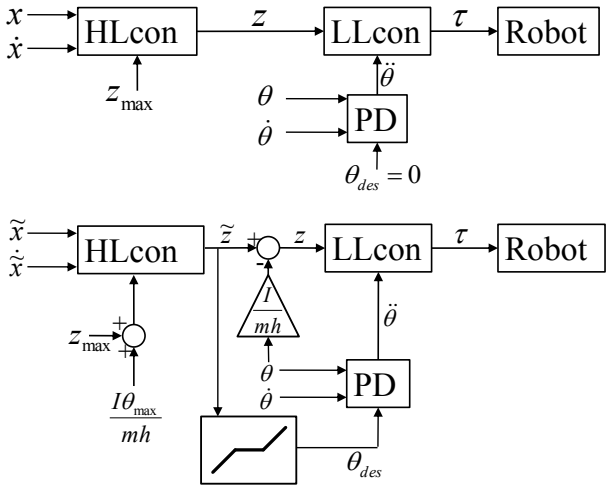

Fig. 3. A flow chart of a simple generic system for torque control of a humanoid robot using standard LIPM dynamics (top) and a flow chart for that same system modified to use the LIPFM dynamics (bottom). "HLCon" represents a high-level controller, "LLcon" represents a low-level controller, and the bottom box represents the function given in (13).

IV. RESULTS

We applied a modified LIPM-based controller to simulated standing, simulated walking, and standing on the Sarcos Primus system force controlled humanoid robot. Both our simulation and robot are torque controlled, so all of our experiments used the configuration depicted in Fig. 3.

A. Simulation

Our simulation is a 10-link 3-dimensional rigid-body simulation based on our Sarcos humanoid robot. The simulation has a mass of 78 kg (45 kg in the torso). It has one spine joint located between the hips, which allows the torso to rotate relative to the pelvis in the coronal plane. Each leg has 2 hip joints, a knee joint, and a calf rotation joint along the long axis of the calf. The hip joints can be used to rotate the pelvis and torso together in the sagittal plane. In addition, each point foot can apply torque directly between the ground and the shin, which simulates feet that are always parallel to the ground plane. Finite size feet are simulated by enforcing the COP constraint,

$$\begin{aligned} |\boldsymbol{\tau}_x| &\leq w_f \mathbf{f}_z \\ |\boldsymbol{\tau}_y| &\leq l_f \mathbf{f}_z \end{aligned} \tag{14}$$

on each foot, where $\mathbf{f}_z$ is the vertical force, and $w_f = 0.05\,\text{m}$ and $l_f = 0.1\,\text{m}$ are half the width and length of the foot. Ground contact is modeled as a spring-damper system between the feet and the ground.

B. Simulated Standing Results

We started with a simple standing balance controller, then modify it as in Fig. 3. We measured how large an external push the unmodified and modified system can withstand

without failing. For our high-level controller, we used two independent PD controllers in the sagittal and coronal plane, each of which uses the COP to servo the COM position.

For low-level control, we used Dynamic Balance Force Control (DBFC) as presented in [13] to generate joint torques, $\boldsymbol{\tau}_j$. DBFC uses a weighted pseudo-inverse with regularization to solve

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & -\mathbf{S} \\ \mathbf{J}(\mathbf{q}) & 0 \end{bmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\tau}_j \end{pmatrix} = \begin{pmatrix} N(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}(\mathbf{q})\hat{\mathbf{f}} \\ -\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} + \ddot{\hat{\mathbf{p}}} \end{pmatrix} \quad (15)$$

where $\mathbf{q}$ is a vector of base coordinates and joint angles, $\mathbf{M}(\mathbf{q})$ is the mass matrix, $\mathbf{J}(\mathbf{q})$ is the Jacobian of both feet, $\mathbf{S} = [0, \mathbf{I}]$ selects the actuated elements of $\mathbf{q}$, $\hat{\mathbf{f}} = [{\mathbf{f}_L}^\mathsf{T}, {\boldsymbol{\tau}_L}^\mathsf{T}, {\mathbf{f}_R}^\mathsf{T}, {\boldsymbol{\tau}_R}^\mathsf{T}]^\mathsf{T}$ is the ground reaction force, and $\ddot{\hat{\mathbf{p}}} = [\ddot{\mathbf{p}}_L^\mathsf{T}, \ddot{\mathbf{p}}_R^\mathsf{T}]^\mathsf{T}$ is the foot acceleration. We enforce a 20% margin for the COP, leaving us with an effective $w_f = 0.04\,\mathrm{m}$ and $l_f = 0.08\,\mathrm{m}$. For standing, the controller attempts to maintain an equal vertical force on the left and right feet.

Fig. 4 shows the effect of modifying this controller to use the LIPFM dynamics. Modifying it in only the sagittal plane increased the resistance to forward or backward pushes, but had no significant effect on the response to lateral pushes, while the opposite is true for modifying it in only the coronal plane. Modifying the system in both planes generally combined the advantages, but as can be seen for back left and back right pushes, there can be some negative effects from coupling between the two planes when there are large angles in both of them.

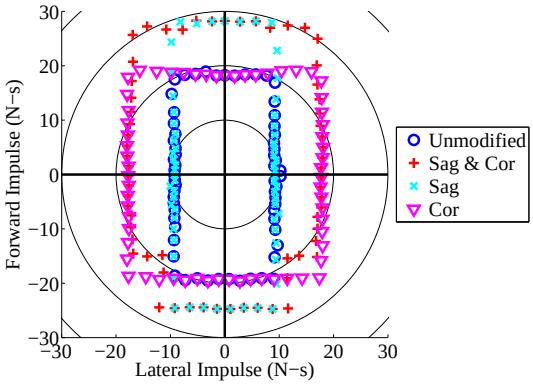

Fig. 4. Polar plot of the maximum survivable perturbation of our standing simulation as a function of push angle. A point represents the maximum survivable perturbation in a given direction. Concentric circles are in increments of 10 Newton-seconds. Data is shown for the unmodified system, the system modified in both the sagittal and coronal planes, only the sagittal, and only the coronal plane.

Fig. 5 shows the response of the original and modified system to a forward push of 18 Ns, which is near the edge of the survivable region for the unmodified system. The modified system recovers from the push more quickly and its COM, x, deviates less far from 0. The COM of the modified system actually moves backwards temporarily in response to the large upper body rotation. While the COM motion of the modified system is more complex, the augmented COM, $\tilde{x}$, displays the approximately exponential decay that we expect from a PD controller.

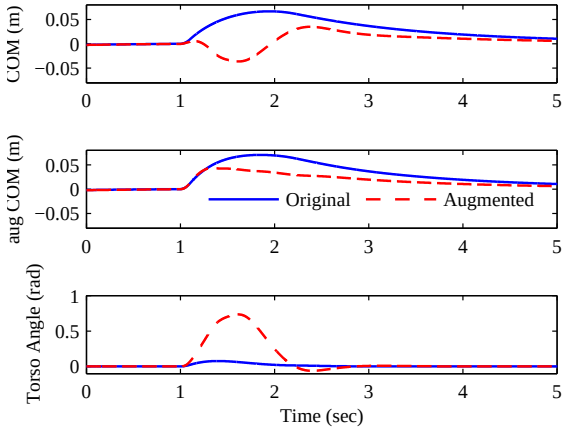

Fig. 5. Simulation: Response of the original and modified system to a forward push of 18 Ns. The push begins at 1 second and lasts for 0.1 seconds.

C. Simulated Walking Results

We performed a similar experiment for simulated walking using the same simulation. Our low-level controller was also the same except that we do not require the vertical force on the left and right feet to be equal. For high-level control of walking, we used a combination of dynamic programming policies similar to that described in [12], except with fixed footstep timing (0.4 second single support and 0.1 second double support) and location (16 cm step width and 50 cm step length). With the footstep timing and locations fixed, walking can be modeled as two independent sets of LIPM dynamics, in the sagittal and coronal plane. Additionally, we can compute swing foot accelerations independent of the COM motion. Therefore, each subsystem: sagittal LIPM, coronal LIPM, swing foot X acceleration, swing foot Y acceleration, and swing foot Z acceleration could be controlled by its own independent dynamic programming controller. We could then improve the performance by modifying the two LIPM controllers to use the LIPFM dynamics.

Fig. 6 shows results similar to those in Fig. 4, but for walking. Perturbations are applied midway through left foot single support. This time we see a significantly larger improvement in the coronal plane than in the sagittal plane because large sagittal plane rotations cause kinematic problems where the swing foot can not reach the desired touch down location. We also see a greater degree of coupling between the two planes, as demonstrated by more rounding of the corners of the survivable rectangles.

D. Hardware Robot Results

The controller from Fig. 3 was implemented on the Sarcos Primus hydraulic humanoid robot. The robot has a mass of

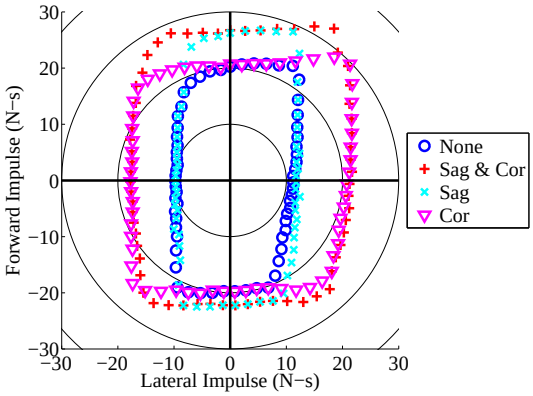

Fig. 6. Polar plot of the maximum survivable perturbation of our walking simulation as a function of push angle. A point represents the maximum survivable perturbation in a given direction. Pushes occur midway through right single support. Concentric circles are in increments of 10 Newton-seconds. Data is shown for the unmodified system, the system modified in both the sagittal and coronal planes, only the sagittal, and only the coronal plane.

95 kg and uses fast force feedback control at every joint to achieve the desired joint torques. The torso orientation is measured by an inertial measurement unit (IMU) attached to the pelvis. COM position and velocity are estimated using a state estimator that accounts for modeling error [14]. The robot is pushed forward while standing still with a force-measuring stick applied to the rear of the black backpack (pictured in Fig. 1) at a height of about 120 cm.

As for the simulation in Section (IV-B), the robot is pushed with and without the proposed controller. The results for two 14 Nm pushes are shown in Fig. 7. In the case where torso rotation is employed (dashed lines), a greater initial decceleration of the COM is achieved. It is also apparent that the controller is actively rotating the torso forward in this case, as opposed to performing pure regulation.

V. DISCUSSION

The change of variables and associated upper body orientation control is a purely dynamic method; it makes no attempt to handle kinematic considerations such as joint limits or reachability of the desired contact locations. The implication is that even if used to post-process (as in Fig. 2) a plan that was originally kinematically feasible, the resulting augmented plan may not be kinematically feasible.

For standing, we handled this problem by continuously controlling the COM height such that the knees remained reasonably straight, but without completely straightening them and pulling the feet off of the ground. For walking, we made certain that the original plan had plenty of margin for reaching the desired touchdown locations so that any torso rotations would not render them unreachable. However, in order to use this method for walking where the footstep locations are allowed to continuously adjust in response to disturbances, kinematic issues would have to be addressed

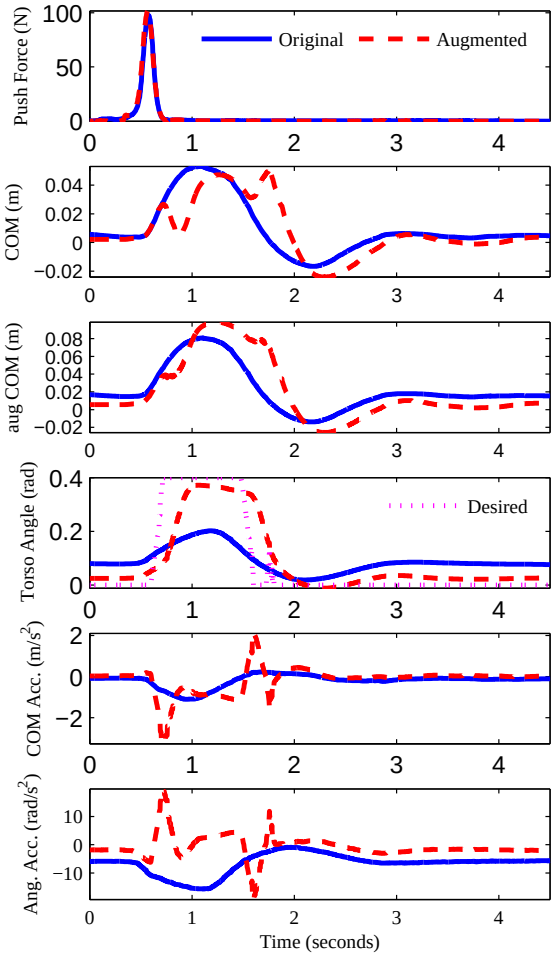

Fig. 7. Hardware Results: Comparing trajectories for two 14 Ns pushes. The solid line corresponds to the robot using the unmodified controller, and the dashed line represents the robot using the modified controller. From top to bottom, the plots represent 1) the applied pushing force, 2) the center of mass position in the forward direction, 3) the augmented center of mass position, as described by (7), 4) the torso pitch angle (including the desired angle for the modified controller), 5) the commanded center of mass acceleration in the forward direction, and 6) the commanded torso pitch acceleration.

explicitly. In that case, it is necessary to balance the tradeoff between the short term benefit of immediately accelerating the center of mass through upper body rotation and the long term benefit of being able to reach a better touchdown location. Additionally, the mechanism for selecting touchdown locations must be aware of what locations are reachable given the torso orientation, which moves the hips relative to the COM.

What we have presented here is a specific instance of a more general principle: a transformation of the coordinate system that simplifies constraints can ease decoupling and simplify or improve control. However, controllers acting on a system that is not perfectly decoupled will interact.

This interaction is less problematic if it does not involve constraints, meaning that each of the active constraints is only governed by one of the controllers. By identifying a mode of motion that did not affect the ZMP constraints, we made it much simpler to use rotation without violating those constraints.

VI. CONCLUSION

In this paper, we present a change of variables into a coordinate system that includes an "augmented COM" which is unaffected by upper body rotation without change in the COP. In the new coordinate system, we can easily compute the motion of the "augmented COM" without considering upper body rotation as well as use upper body rotation in addition to COP modulation for balance. This change of variables and the associated upper body orientation control is intended as a modification for existing LIPM-based controllers, and we show how to modify a generic controller to use it. We demonstrate improved robustness to external pushes using our modification in simulated standing and walking experiments. We also demonstrate this method on our force-controlled humanoid robot in standing balance experiments.

REFERENCES

[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: humanoid motion planning based on the linear and angular momentum," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, no. 5, 2003.

[2] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2006.

[3] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, vol. 4, Honda Res. Inst., Mountain View, CA, USA, 2004, pp. 3785–3790.

[4] 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.

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

[6] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, "Real time motion generation and control for biped robot -4th report: Integrated balance control," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee, Oct. 2009, pp. 1601–1608.

[7] D. Kaynov, P. Soueres, P. Pierro, and C. Balaguer, "A practical decoupled stabilizer for joint-position controlled humanoid robots," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3392–3397, Oct. 2009.

[8] 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.

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

[10] A. Hofmann, M. Popovic, and H. Herr, "Exploiting angular momentum to enhance bipedal center-of-mass control," in *2009 IEEE International Conference on Robotics and Automation*. Ieee, May 2009, pp. 4423–4429.

[11] S.-h. Lee and A. Goswami, "Ground reaction force control at each foot : A momentum-based humanoid balance controller for non-level and non-stationary ground," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3157–3162.

[12] E. Whitman and C. Atkeson, "Control of Instantaneously Coupled Systems Applied to Humanoid Walking," in *Proceedings of the IEEE International Conference on Humanoid Robots*, 2010.

[13] 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.

[14] B. J. Stephens, "State Estimation for Force-Controlled Humanoid Balance using Simple Models in the Presence of Modeling Error," in *IEEE International Conference on Robotics and Automation*, 2011.