

Dynamic Balance Force Control for Compliant Humanoid Robots

Benjamin J. Stephens, Christopher G. Atkeson

Abstract—This paper presents a model-based method, called Dynamic Balance Force Control (DBFC), for determining full body joint torques based on desired COM motion and contact forces for compliant humanoid robots. The center of mass (COM) dynamics are affected directly through contact force control to achieve stable balance. This idea is used to formulate DBFC considering the full rigid-body dynamics of the robot to produce desired contact forces. To achieve generic force control tasks, a virtual model controller, DBFC-VMC, is presented. Examples using this control are presented as results from simulation and experiments on a force-controlled humanoid robot.

I. INTRODUCTION

Humanoid robots must operate in complex environments while interacting closely with people and performing a wide variety of tasks. Many tasks involve the regulation of forces, requiring compliant mechanisms and controllers that are stable but also safe and robust to unknown disturbances. This paper describes a simple method of control for full body balance and other tasks that is suitable for compliant force-controlled humanoid robots.

While humanoid robots are very complex systems, the dynamics that govern balance are often described using simple models of the center of mass (COM) [1]. It has been shown through dynamic simulation that humanoid balance depends critically on controlling the linear and angular momentum of the system [2], quantities that can be directly controlled by contact forces. This suggests that balance is a fundamentally low-dimensional problem that can be solved by contact force control. This idea is the inspiration for the controller presented in this paper.

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) [3]. For environments with small uncertainty or small disturbances, the inverse kinematics can be modified to directly control the contact forces using force feedback [4]. Position-based controllers generally exhibit high impedance, and the speed at which they will comply to an unknown force is limited. Robots with low impedance joints can comply faster. This is useful, but also makes balance control more important and more difficult.

For compliant robots, there are a number of ways that contact force control can be achieved. Virtual model control (VMC) [5] is the simplest method that only uses a kinematic

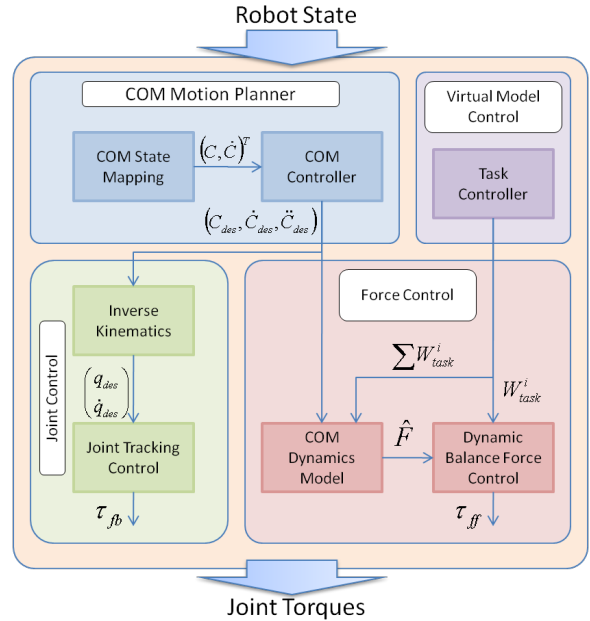


Fig. 1. Block diagram of full control algorithm including DBFC.

model. Desired contact forces are converted into joint torques assuming static loading using a Jacobian-transpose mapping. It has been shown that under quasistatic assumptions and proper damping of internal motions the desired forces can be achieved [6]. In contrast, given the full constrained rigid-body dynamics model, desired joint accelerations can be converted into joint torques using inverse dynamics for improved tracking performance [7].

This paper presents another method, called Dynamic Balance Force Control (DBFC), which is summarized in Figure 1. Like [7], the full dynamic model is used and no quasistatic assumptions are made. However, like [5] and [6], the input is desired contact forces. Contact forces are computed independent of the full robot model based on a simple COM dynamics model and external forces. Because of force-based nature of this controller, it can be modified for the compensation of non-contact forces using VMC-like controls. This modification, called DBFC-VMC, can be used to perform generic tasks such as posture control and manipulation. The output of the DBFC(-VMC) is full body joint torques. Figure 2 offers a comparison these various control methods.

This paper is organized as follows. In Section II, desired contact forces are calculated from a simplified model based on COM dynamics. In Section III, it is shown how full body

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/~bstephel>

	VMC[5]	PBBC[6]	FBID[7]	DBFC
Inertial Model	No	No	Yes	Yes
COM/Force-Based	Yes	Yes	No	Yes
Constraint Handling	No	Damps internal motions	Solves with constraints	Solves with constraints
Model-free Tasks	Yes	Not presented	Not presented	Yes (DBFC-VMC)
Contact Force Method	Hand chosen	Min norm distribution	N/A	Quadratic programming

Fig. 2. Comparison of related control methods: Virtual Model Control (VMC), Passivity-Based Balance Control (PBBC), Floating Body Inverse Dynamics (FBID), and Dynamic Balance Force Control (DBFC).

joint torques can be calculated using DBFC. To perform more general tasks, DBFC-VMC is presented in Section IV. Results are given in Section V from experiments on a Sarcos humanoid robot. Several examples showing a wide range of tasks are presented.

II. COM DYNAMICS MODEL

The COM dynamics of a general biped system with two feet in contact with the ground are represented by a system of linear equations which sum the forces and torques on the COM. If $C = (x, y, z)^T$ is the location of the COM, P_R and P_L are the locations of the two feet with respect to the COM, and $F_R, F_L, M_R,$ and M_L are the ground reaction forces and torques, then the dynamics can be written generally as

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} F = \begin{pmatrix} m\ddot{C} + F_g \\ \dot{L} \end{pmatrix} \quad (1)$$

where

$$D_1 = \begin{bmatrix} I & 0 & I & 0 \end{bmatrix} \quad (2)$$

$$D_2 = \begin{bmatrix} (P_R) \times & I & (P_L) \times & I \end{bmatrix} \quad (3)$$

and

$$F = \begin{pmatrix} F_R \\ M_R \\ F_L \\ M_L \end{pmatrix} \quad (4)$$

Here, $r \times$ represents the left cross product matrix, m is the total mass of the system, F_g is the constant gravitational force which points in the $-z$ -direction and \dot{L} is the rate of change of angular momentum. 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 $\dot{L} = 0$, any forces that satisfy these equations do not generate angular momentum about the center of mass. Additionally, if $\ddot{z} = 0$, the dynamics are identical to the well-known Linear Inverted Pendulum Model (LIPM) [8].

These equations can be used to solve for a valid set of desired contact forces, \hat{F} , through the solution of a constrained optimization. If (1) is abbreviated as

$$KF = u \quad (5)$$

then the desired forces, \hat{F} , can be found by solving the quadratic programming problem,

$$\hat{F} = \arg \min_F F^T W F \quad (6)$$

$$\text{s.t. } KF = u \quad (7)$$

$$BF \leq c \quad (8)$$

where $BF \leq c$ represents linear inequality constraints due to the support polygon or friction limits. $W = \text{diag}(w_i)$ can be used to weight certain forces more than others, for example to penalize large horizontal forces. In order to keep the center of pressure under the feet, the constraints,

$$d_Y^- \leq \frac{M_X}{F_Z} \leq d_Y^+ \quad (9)$$

$$d_X^- \leq -\frac{M_Y}{F_Z} \leq d_X^+ \quad (10)$$

must be met for each foot, where d_X^\pm and d_Y^\pm represent the dimensions of the feet. These equations can be re-written as linear constraints on the forces and torques.

Friction constraints can also be considered. However, the general form of these constraints is nonlinear in the forces,

$$\left| \frac{\sqrt{F_X^2 + F_Y^2}}{F_Z} \right| \leq \mu \quad (11)$$

A solution is to write a simple conservative approximation,

$$\left| \frac{F_X}{F_Z} \right| \leq \frac{\mu}{\sqrt{2}} \quad (12)$$

$$\left| \frac{F_Y}{F_Z} \right| \leq \frac{\mu}{\sqrt{2}} \quad (13)$$

which can also be written as linear constraints. Higher order approximations to these constraints can also be used.

During single support, this optimization is not required because there are 6 equations and 6 unknowns, meaning the forces on the stance foot are completely determined by the desired accelerations.

It is possible that, given the constraints, there will be no valid solution to the optimization in (6). If that is the case, it is possible to re-write the quadratic optimization to solve a constrained least squares problem,

$$\hat{F} = \arg \min_F (KF - u)^T (KF - u) + F^T W F \quad (14)$$

$$\text{s.t. } BF \leq c \quad (15)$$

which will always find a solution, but may not be able to achieve the desired accelerations.

III. DYNAMIC BALANCE FORCE CONTROL

The objective of DBFC is to determine the joint torques that will achieve some desired body motion. Rather than perfect joint trajectory control, this paper is concerned with balance-related tasks, meaning that control of the motion of the COM and angular momentum is important. These

quantities can be controlled by the contact forces. Presented below, DBFC is a model-based method for determining full body joint torques based on desired COM motion and contact forces.

Consider the full nonlinear dynamics and constraint equations for a floating-body model,

$$M(q)\ddot{q} + N(q, \dot{q}) = S\tau + J^T(q)F \quad (16)$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = \ddot{P} \quad (17)$$

where $S = [0, I]^T$ selects the actuated degrees of freedom, \ddot{P} is the linear and angular accelerations of the feet, $J(q)$ is the Jacobian of the center of both feet and q is a vector of base coordinates and joint angles.

These equations can be augmented with the dynamics of the center of mass in (1), where the desired acceleration of the COM, \ddot{C}_{des} , acceleration of the feet, \ddot{P}_{des} , and change of angular momentum, \dot{L}_{des} , are specified. Instantaneously, the result is a set of linear equations,

$$\begin{bmatrix} M & -S & -J^T \\ J & 0 & 0 \\ 0 & 0 & D_1 \\ 0 & 0 & D_2 \end{bmatrix} \begin{pmatrix} \ddot{q} \\ \tau \\ F \end{pmatrix} = \begin{pmatrix} -N \\ \ddot{P}_{\text{des}} - \dot{J}\dot{q} \\ m\ddot{C}_{\text{des}} + F_g \\ \dot{L}_{\text{des}} \end{pmatrix} \quad (18)$$

This linear system can be solved by constrained quadratic programming to determine joint torques, τ , using the same methods described in the previous section.

However, a valid set of contact forces, \hat{F} , that obey the contact constraints can be found by solving (14), then (18) can be simplified by plugging in those values and eliminating the COM equations,

$$\begin{bmatrix} M(q) & -S \\ J & 0 \end{bmatrix} \begin{pmatrix} \ddot{q} \\ \tau \end{pmatrix} = \begin{pmatrix} -N(q, \dot{q}) + J^T \hat{F} \\ -\dot{J}\dot{q} \end{pmatrix} \quad (19)$$

where $\ddot{P}_{\text{des}} = 0$ if both feet are on the ground. This system can be written as $Gz = f$ and solved by a weighted pseudo-inverse,

$$z = (G^T G + W)^{-1} G^T f \quad (20)$$

where W is used to regularize the solution. The solution of this equation, $z^T = [\ddot{q}^T, \tau_{ff}^T]$, contains the joint torques and is much faster to solve than the constrained quadratic programming problem in (18).

IV. DBFC VIRTUAL MODEL CONTROL

In addition to balance control, the ability to perform other tasks, such as posture control and manipulation is useful. Such tasks can be integrated into DBFC by including virtual task forces, just as in Virtual Model Control. These forces are reflected both in the contact forces and the joint torques. In the method described here, the compensating contact forces are first calculated using COM dynamics and then the joint torques are calculated using DBFC.

First, task forces and torques that affect the COM dynamics are used to offset the contact forces,

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} F = \begin{pmatrix} m\ddot{C}_{\text{des}} + F_g + \sum_i F_{\text{task}}^i \\ \sum_i (P^i - C) \times F_{\text{task}}^i + M_{\text{task}}^i \end{pmatrix} \quad (21)$$

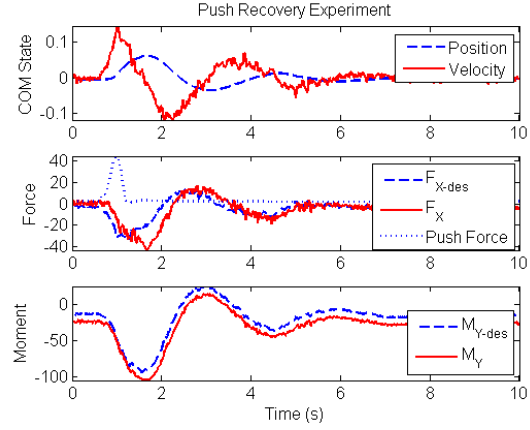


Fig. 3. Example push recovery experiment using the DBFC approach in this paper.

where task forces, F_{task}^i , and torques, M_{task}^i , are applied at some specific point, P^i , on the body. The cross product is included for when a task force also applies a torque about the COM. Solving for the contact forces, \hat{F} , in this case will compensate for the task forces. Now the robot joint torques can be found by solving the DBFC problem with the task forces included,

$$G \begin{pmatrix} \ddot{q} \\ \tau \end{pmatrix} = \begin{pmatrix} -N(q, \dot{q}) + J^T \hat{F} + \sum_i J_{\text{task}}^{iT} W_{\text{task}}^i \\ -\dot{J}\dot{q} \end{pmatrix} \quad (22)$$

where J_{task}^i is the 6-dof Jacobian associated with the point where the i -th task forces are applied and

$$W_{\text{task}}^i = \begin{pmatrix} F_{\text{task}}^i \\ M_{\text{task}}^i \end{pmatrix} \quad (23)$$

V. RESULTS

Several results are presented below using DBFC on a humanoid robot and in simulation. The examples are meant to both describe the performance and implementation details of the controllers. The same rigid body dynamics model is used in simulation and in control of the real robot.

Standing balance control is demonstrated in experiments performed on a Sarcos Primus humanoid robot. The robot uses hydraulic actuators with force feedback to perform compliant torque control on every joint [9]. Power is provided by an off-board pump with tethered hoses that connect to a manifold on the hip. There are potentiometers at every joint, an inertial measurement unit (IMU) mounted on the hip, and 6-axis force/torque sensors on each foot. For push experiments, the push force is measured using a 6-axis force/torque sensor mounted on the end of a stick.

Walking control is presented in simulation only. The simulator assumes a simple spring-mass contact model. The feet are assumed to be point feet, but have the ability to apply torques to the ground when in contact.

A. EXAMPLE: BALANCE CONTROL

For standing balance control, the objective is simply to regulate the position of the COM, which is achieved by a

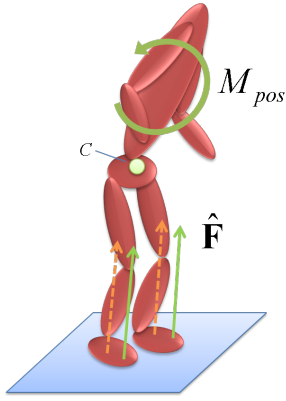


Fig. 4. Posture control is achieved by a virtual torque on the torso. The original contact forces (dashed) are modified (solid) to compensate for this virtual torque.

PD controller,

$$\ddot{C}_{des} = -K_p(C_{des} - C) - K_d\dot{C} \quad (24)$$

This desired acceleration can be converted into contact forces using the method in Section II. Solving (19) gives the appropriate joint torques. In practice, small feedback torques, τ_{fb} , are also added to bias the joint angles and velocities to desired angles and velocities,

$$\tau_{fb} = K_p(q_{des} - q) + K_d(\dot{q}_{des} - \dot{q}) \quad (25)$$

For a standing balance task where the robot is pushed from behind, the result of the DBFC controller is shown in Figure 3. This figure shows the total desired and measured F_x and M_y forces, as well as the state of the robot COM after the push.

B. EXAMPLE: TORSO POSTURE CONTROL

Maintaining torso posture can be a desirable goal during standing balance. Often this is handled by special consideration of the hip joint torque [10]. This example shows how torso posture control can be achieved via DBFC-VMC and implemented on a humanoid robot. The posture, or torso angle, can be corrected by a torque, M_{torso} , to the torso, as shown in Figure 4. This torque can be written as a simple PD controller,

$$M_{torso} = K_p(\theta_{des}^{pos} - \theta_{pos}) - K_d\dot{\theta}^{pos} \quad (26)$$

where θ^{pos} is the orientation of the torso. Using (21), \hat{F} can be determined by solving

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} F = \begin{pmatrix} m\ddot{C}_{des} + F_g \\ M_{torso} \end{pmatrix} \quad (27)$$

and (22) can be used to determine the joint torques,

$$G \begin{pmatrix} \ddot{q} \\ \tau \end{pmatrix} = \begin{pmatrix} -N(q, \dot{q}) + J^T \hat{F} + \sum_i J_{torso}^T M_{torso} \\ -J\dot{q} \end{pmatrix} \quad (28)$$

where J_{torso} is the Jacobian associated with the torso body to which the virtual torque is applied.

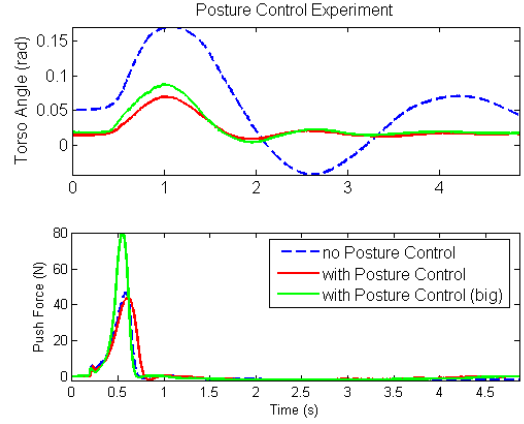


Fig. 5. Torso angle with and without torso posture control. The force of the push was measured to compare the performance relative to push size.

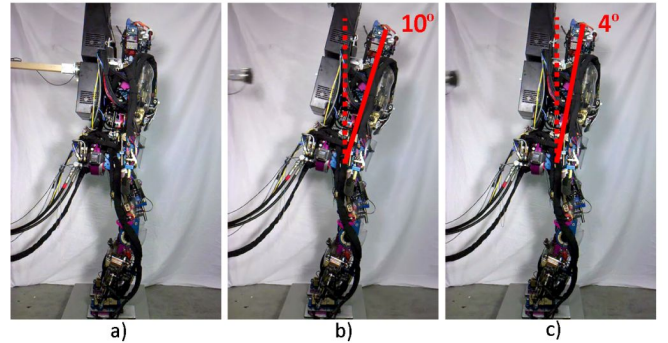


Fig. 6. Comparing torso posture control responses. In a) the robot is pushed from behind to the middle of the torso with a force sensing stick. The other two figure shows the maximum displacement first with b) no posture control and finally c) with posture control.

This controller was applied to the Sarcos Primus humanoid robot, which used an inertial measurement unit (IMU) attached to the hip to measure torso angle with respect to ground. Figure 5 shows the result of the balance controller when the robot is pushed from behind at the middle of the torso with and without posture control. Without posture control, the torso is underdamped and has a large deviation from upright. This is corrected by torso posture control with $K_p = 150$ and $K_d = 150$. The robot is also able to handle much larger pushes with this controller. Photos of the robot during this experiment are shown in Figure 6.

C. EXAMPLE: HEAVY LIFTING

This task further demonstrates the capabilities of the DBFC-VMC controller. When lifting a heavy object as shown in Figure 7, DBFC-VMC can be used to generate the compensating joint torques. The VMC task controller can be defined to regulate the position of the object,

$$F_{lift} = K_p(P_{des}^{object} - P^{object}) \quad (29)$$

Now (21) takes the form

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} F = \begin{pmatrix} m\ddot{C}_{des} + F_g + F_{lift} \\ (P^{object} - C) \times F_{lift} \end{pmatrix} \quad (30)$$

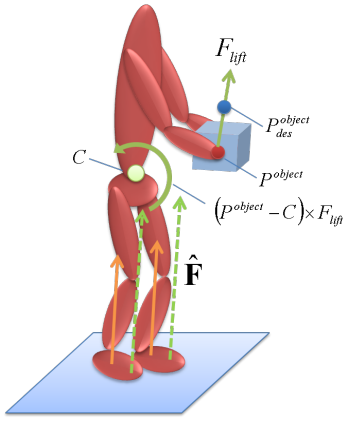


Fig. 7. Lifting a heavy object requires both a task force and task torque to compensate for the gravitational force on the object.

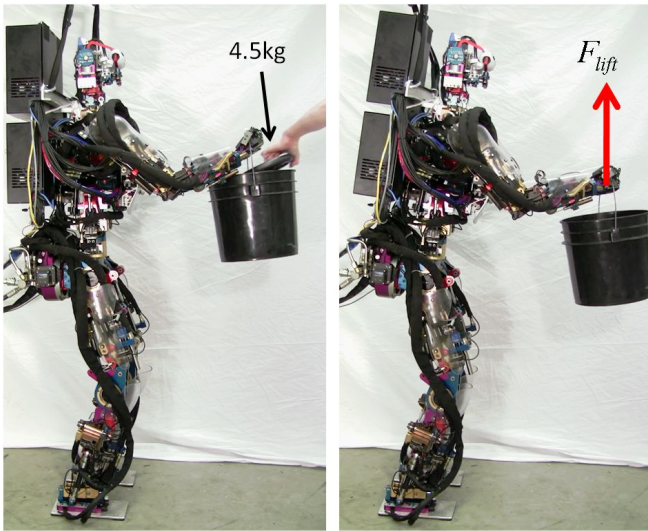


Fig. 8. Experimental setup for evaluating the lifting task. A 4.5kg mass is dropped into the bucket. The robot applies a force, F_{lift} , to the bucket to control the height of the bucket.

The mass of the object is not needed for VMC control, but could be implemented in a feedforward manner if known, $\hat{F}_{\text{lift}} = F_{\text{lift}} + m^{\text{object}}g$. In addition to the task force, a task torque is required because the gravitational force on the object applies a torque about the COM. (22) is again used to determine the joint torques,

$$G \begin{pmatrix} \ddot{q} \\ \tau \end{pmatrix} = \begin{pmatrix} -N(q, \dot{q}) + J^T \hat{F} + \sum_i J_{\text{lift}}^T F_{\text{lift}} \\ -\dot{J}\dot{q} \end{pmatrix} \quad (31)$$

where J_{lift} defines the point on the arms where the lift force is applied.

Figure 8 shows the experimental setup used to test this controller. A bucket is attached to the hand so only a vertical force is required. A 4.5kg mass is dropped into the bucket from a height of approximately 22cm. Without the VMC, the robot falls forward. However, when VMC is enabled with $K_p > 0$, the robot maintains its balance. A comparison of the performance for different K_p values is shown in Figure 9. As expected, F_{lift} increases with K_p and the steady state error

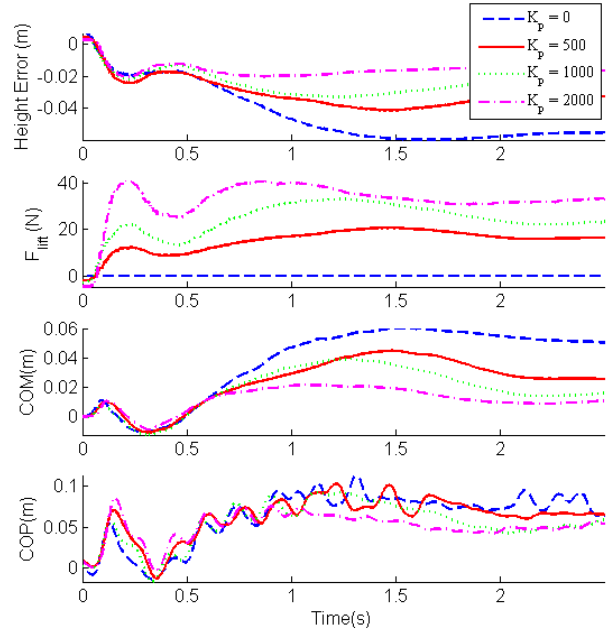


Fig. 9. Comparison of heavy lifting experiments with a range of K_p values.

is reduced. To eliminate the steady-state offset, an integrator could be added to (29).

D. EXAMPLE: FORCE FEEDBACK

The previous heavy object lifting task can be modified to incorporate force feedback control. The mass of the heavy object causes the effective COM of the entire system to move forward. This can be sensed by measuring the center of pressure (COP) using the force-torque sensors on the feet. The desired COP, COP_{des} , is the center of the base of support. To compensate for the extra mass, the COM of the robot is moved backwards using an integral control,

$$C_{\text{des}} = \frac{K_i}{T_0} \int (COP_{\text{des}} - COP) dt \quad (32)$$

where $K_i > 0$ and T_0 is a scaling factor.

Figure 10 shows the effect of the force feedback controller in (32). Without feedback ($K_i = 0$) the COP moves to the front of the foot where balance control is poor. As K_i is increased, the COP is moved back to the center of the foot and the COM of the robot is moved backward. This also results in better task performance because the robot has greater force control authority in this state. If K_i is increased too much, oscillations in the COP can occur due to the dynamics of the robot. A COP controller that combines integral control and the dynamics of the robot [11] could be implemented for improved control.

Figure 11 demonstrates control with and without force feedback when multiple weights are added to the bucket one at a time. For the $K_p = 0$ case, the robot falls over after the second weight is added. With $K_p = 2000$, the robot can remain standing. Without force feedback, both the COM and COP move forward with each weight. With force feedback,

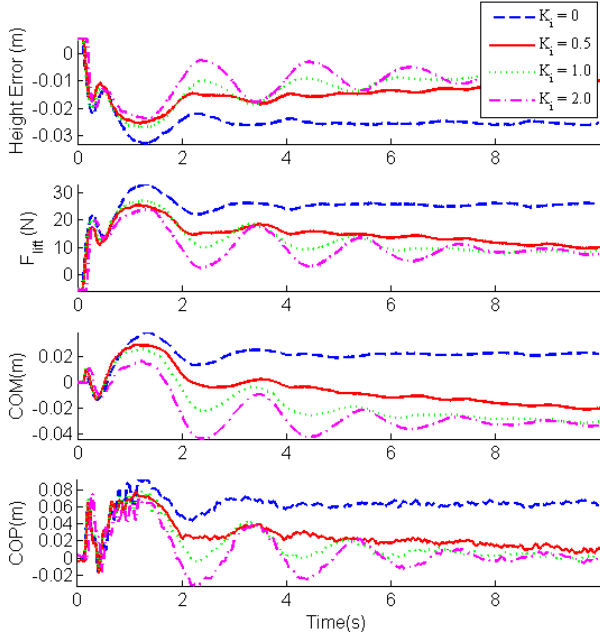


Fig. 10. Comparison of force feedback controls for a range of K_i values. $K_p = 1000$ and $T_0 = 700$ for all experiments.

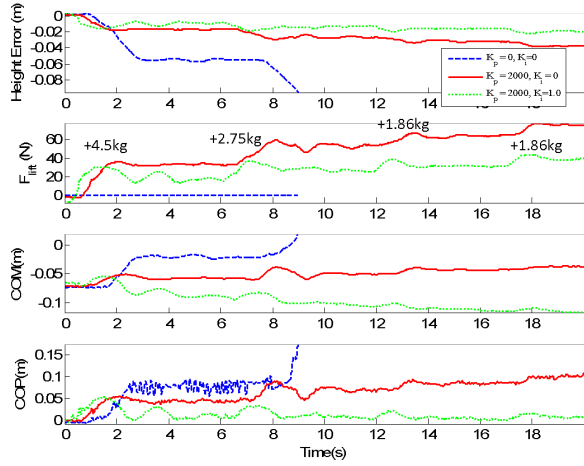


Fig. 11. Comparison of different K_p and K_i gains for lifting multiple weights.

the COM is moved backward and the COP is regulated to the middle of the foot.

E. EXAMPLE: WALKING CONTROL

This controller can be extended beyond standing balance to the control of locomotion such as walking. For this example, walking trajectories of the COM are generated using preview control [12] with adaptive foot placement [13]. Preview control is a model predictive controller that uses the LIPM dynamics to generate COM trajectories for walking. The details of this algorithm will not be presented here. What will be presented is how DBFC-VMC can use preview

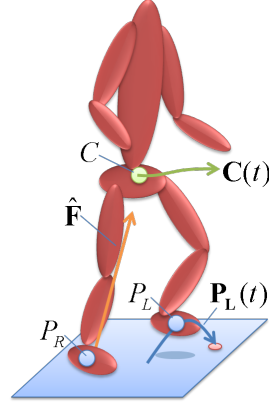


Fig. 12. Walking control is achieved by tracking the desired COM and swing foot trajectories. Not only are inverse kinematics used, but DBFC-VMC is used to generate full body joint torques.

control, or other similar motion generators, to generate fully body joint torques for walking.

Preview control can be thought of as a function, Φ , of the current COM state, (C, \dot{C}) and foot locations, P_L and P_R , that returns trajectories for the COM, $\mathbf{C}(t)$ and feet, $\mathbf{P}_L(t)$ and $\mathbf{P}_R(t)$, over the next several footsteps,

$$\Phi : \{C, \dot{C}, P_L, P_R\} \rightarrow \{\mathbf{C}(t), \mathbf{P}_L(t), \mathbf{P}_R(t)\} \quad (33)$$

as illustrated in Figure 12. In this example, footsteps are being planned 3 steps into the future and replanned just after each touchdown.

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

$$\ddot{\mathbf{C}}_{\text{des}} = \ddot{\mathbf{C}}(t) + K_p (\mathbf{C}(t) - C) + K_d (\dot{\mathbf{C}}(t) - \dot{C}) \quad (34)$$

When substituted into (21), this generates the desired contact forces, $\hat{\mathbf{F}}$. Joint torques can be found using (22), which can be re-written as

$$G \begin{pmatrix} \ddot{q} \\ \tau \end{pmatrix} = \begin{pmatrix} -N(q, \dot{q}) + J^T \hat{\mathbf{F}} \\ \dot{\mathbf{P}}(t) - \dot{J}\dot{q} \end{pmatrix} \quad (35)$$

where $\dot{\mathbf{P}}(t)$ is non-zero for the swing leg. The footstep trajectories can be defined in a number of ways. In this paper, minimum jerk trajectories are defined between the takeoff position and desired footstep location with a predefined height of 5cm starting and ending with zero velocity. Joint trajectories are calculated using inverse kinematics [14], determining both desired joint angles and joint velocities, and tracked using low gain PD control.

This walking controller has been successfully applied in simulation. Figure 13 shows the COM positions and velocities in the horizontal plane. The desired COM trajectories calculated by preview control are tracked very closely even though joint position gains are low and joint trajectory tracking is not perfect. This is thanks to the use of torques calculated by DBFC. Figure 14 shows the desired contact forces used by DBFC and the actual contact forces that result.

VI. DISCUSSION AND FUTURE WORK

The controller presented in this paper is a model-based controller. It uses a 3D rigid-body dynamics model of the entire robot to determine forward and inverse dynamics. One of the problems with this type of controller is that the robot does not match the model. Not only can kinematics and mass properties be wrong, but unmodeled effects such as hydraulic fluid flow, deformable joints, and ground contact compliance can cause unpredicted behavior. This is a significant problem for force-controlled robots performing dynamic motions. While the robot model used to control the robot in this paper is not perfect, reasonable force control is achieved, for example in Figure 3.

The simplest method for overcoming some modeling error is to include the feedback controller in (25). The gains used for feedback control can be a PD controller, which assumes all joints are independent, or an LQR full-state feedback controller derived from a linearized model. While in this paper, this joint feedback controller is simply appended to the DBFC torques, a prioritized control approach could be used to limit the interaction between the two controllers [15].

The role of angular momentum is not emphasized in this paper. Angular momentum is generated by applying a torque about the COM, for example by bending quickly at the hip or swinging the arms. Doing so can momentarily generate higher horizontal forces to aid balance. However, the humanoid form is not built to store angular momentum as in a flywheel. This means that any angular momentum added to the system must also be removed quickly. Coordinating this tradeoff is difficult and the subject of future work. In this paper, the desired behavior is to minimize the generation of angular momentum by setting the desired change of angular momentum to zero, $\dot{L}_{des} = 0$. In fact, it is generally useful to dissipate angular momentum by setting $\dot{L}_{des} = -K_p L$.

The DBFC-VMC controller can accommodate a wide variety of tasks. It also inherits from traditional VMC the ability to define tasks in a model-free way. As shown in Figure 7, a model of the object isn't needed, only a definition of a controller to grasp it. However, if a model is provided, DBFC-VMC can take advantage of the model to achieve improved control.

A slightly modified version of the task in Figure 7 would include grasping forces, which are internal to the system. To grasp the object steadily, both arms could apply an equal and opposite force. This force has no effect on the ground contact forces but does affect the joint torques in the two arms. The lifting example also highlights a possible problem with the DBFC-VMC as presented. If the object is heavy enough to tip the robot over, adjustments such as leaning backwards or moving a foot to reshape the base of support may be required [16]. However, with proper placement of the COM and feet, DBFC is a simple choice for determining full body joint torques.

Future work using DBFC will include the implementation of walking and step recovery on the Sarcos humanoid robot. Modeling error and state estimation make hardware

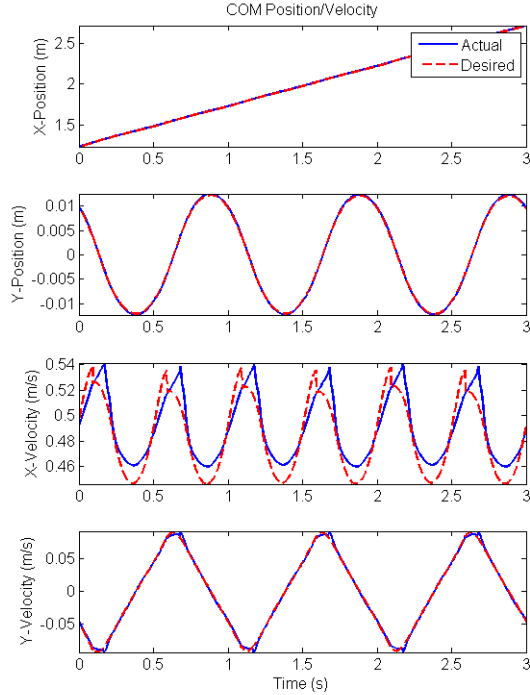


Fig. 13. Trajectory of the COM generated by preview control for a 0.5 m/s desired walking speed in simulation. Dashed lines represent desired positions/velocities and solid lines represent true state of the robot.

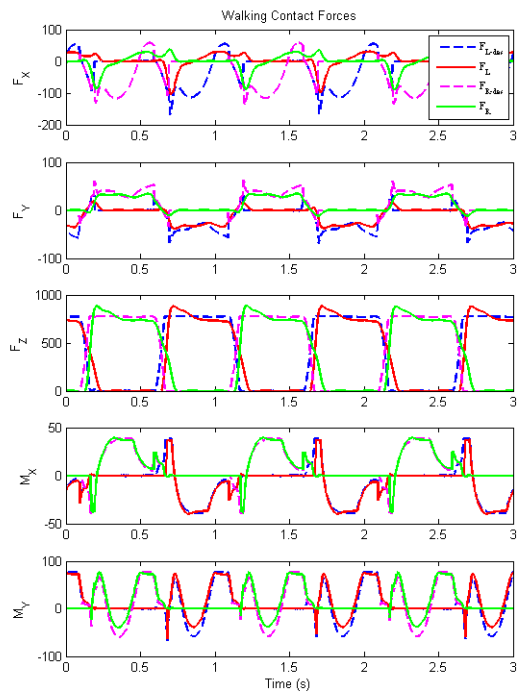


Fig. 14. Contact forces during forward walking at 0.5 m/s in simulation. The dashed lines are the desired forces used by DBFC to create the full body joint torques whereas the solid lines are the actual contact forces.

implementation difficult, though these behaviors have been shown to work very well in simulation, even when subjected to very large disturbances. By focusing on push recovery in a force-control context, the goal of this research is to develop very robust controllers for balance and locomotion that can work even in uncertain or rough terrain environments.

VII. CONCLUSION

This paper has presented a controller called Dynamic Balance Force Control that is suitable for performing a wide variety of tasks in the presence of unknown perturbations. Using the same simple models that are often used in other non-force-controlled humanoid robots, desired contact forces can be predicted. DBFC uses these contact forces to determine full body joint torques. Furthermore, virtual model control is integrated into this framework, allowing model-free task execution. The effectiveness of this control was demonstrated in standing balance experiments on a force-controlled humanoid robot and walking control in simulation.

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. Stephens, "Humanoid Push Recovery," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2007.
- [2] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 1, 2009.
- [3] 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.
- [4] 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.
- [5] J. Pratt and G. Pratt, "Intuitive control of a planar bipedal walking robot," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*. Leuven, Belgium: IEEE, 1998, pp. 2014–2021.
- [6] 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, October 2007.
- [7] M. Mistry, J. Buchli, and S. Schaal, "Inverse Dynamics Control of Floating Base Systems using Orthogonal Decomposition," in *Proceedings of the 2010 IEEE Int. Conference on Robotics and Automation*, 2010.
- [8] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, April 1991, pp. 1405–1411.
- [9] D. C. Bentevegna and C. G. Atkeson, "Compliant control of a hydraulic humanoid joint," *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pp. 483–489, November 2007.
- [10] K. Yin, "SIMBICON : Simple Biped Locomotion Control," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 1–10, 2007.
- [11] B. Stephens, "Integral Control of Humanoid Balance," in *Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems*, 2007.
- [12] 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 of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 1620–1626.
- [13] 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 *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1121–1126.
- [14] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2008.
- [15] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, "A unifying framework for robot control with redundant DOFs," *Autonomous Robots*, vol. 24, no. 1, pp. 1–12, 2007.
- [16] E. Yoshida, O. Kanoun, C. Esteves, and J.-P. Laumond, "Task-driven Support Polygon Reshaping for Humanoids," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 208–213.