2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Inverse Dynamics Control of Floating Base Systems Using Orthogonal Decomposition

Michael Mistry, Jonas Buchli, and Stefan Schaal

Abstract— **Model-based control methods can be used to enable fast, dexterous, and compliant motion of robots without sacrificing control accuracy. However, implementing such techniques on floating base robots, e.g., humanoids and legged systems, is non-trivial due to under-actuation, dynamically changing constraints from the environment, and potentially closed loop kinematics. In this paper, we show how to compute the analytically correct inverse dynamics torques for model-based control of sufficiently constrained floating base rigid-body systems, such as humanoid robots with one or two feet in contact with the environment. While our previous inverse dynamics approach relied on an estimation of contact forces to compute an approximate inverse dynamics solution, here we present an analytically correct solution by using an orthogonal decomposition to project the robot dynamics onto a reduced dimensional space, independent of contact forces. We demonstrate the feasibility and robustness of our approach on a simulated floating base bipedal humanoid robot and an actual robot dog locomoting over rough terrain.**

I. Introduction

A dexterous and skillful humanoid robot that can coexist and interact with humans in our own environment has been a major dream of robotics researchers. Such robots would not only need to be fast and agile, but also compliant enough to be considered safe for human interaction. Robots controlled with traditional joint position control techniques typically require too high stiffness levels in order to maintain accuracy. Greater compliance can be achieved, without sacrificing accuracy, using *model-based* techniques such as inverse dynamics control. In these approaches, an estimated model of the robot's dynamics is used to proactively apply control forces required to track joint or task space trajectories. Such model-based controllers have been well studied in the realm of manipulator robotics (see [1] for a recent review). However, humanoid systems complicate matters since they are typically not fixed to their environments and are free to move about. As a consequence, these *floating-base* systems have under-actuated dynamics with respect to an inertial reference frame, as well as dynamically changing contact states, potential closed loop kinematics, and contact forces that may not be known.

In recent literature, there has been notable progress towards understanding these high-dimensional floating base systems. The works of [2],[3] have extended Khatib's operational space framework [4] to the control of floating base humanoid robot systems in contact with their environments. The advantage of Khatib's framework is that it computes the control forces that decouple task and null-space dynamics. Thus secondary tasks can be achieved without interfering dynamically with higher priority tasks. Additionally, as it is a force control method, it does not use joint position references, and thus does not require explicit inverse kinematics computation. However, these methods rely heavily on accurate modeling, in particular the inertia matrix which is required for pseudoinverse computations and projections.

The work of [5] has applied passivity based approaches to floating base humanoid control and balance, by controlling for optimal contact forces at the feet. They have demonstrated robust disturbance rejection on a real humanoid platform, and have recently shown [6] improved balancing via integration of static and dynamic strategies in a biologically plausible manner.

In this work, we propose a relatively simple technique for full-body model-based control of humanoid robots. Using an orthogonal decomposition of rigid-body dynamics, we are able to express the complete inverse dynamics equations of the robot independently of contact forces. While use of orthogonal decomposition of rigid-body dynamics is not new ([7],[8]), in this work we show how the technique can be used to cope with the under-actuation and dynamically changing contact state inherent in these floating base systems. Additionally, since the decomposition uses only kinematic variables, it avoids the pitfalls of relying on difficult-to-model dynamic projections.

First, we will introduce floating base rigid-body dynamics and notation. We will then show how an orthogonal decomposition can be used to eliminate constraint forces, and produce a simple inverse dynamics control equation. We will evaluate our method on a simulated biped humanoid, as well as an actual robot dog walking over rough terrain. In both cases, our approach increases movement accuracy, allows for the reduction of feedback gains, and enhances compliance and robustness towards perturbations.

II. Floating Base Dynamics

The floating base framework provides the most general representation of a rigid-body system unattached to the world, and is necessary to describe the complete dynamics of the system with respect to an inertial frame. The system configuration is represented as:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_r^T & \mathbf{x}_b^T \end{bmatrix}^T \quad (1)$$

where $\mathbf{q}_r \in \mathbb{R}^n$ is the joint configuration of the rigid-body robot with n joints and $\mathbf{x}_b \in SE(3)$ is the position and orientation of the coordinate system attached to the robot

M. Mistry is with Disney Research Pittsburgh, Pittsburgh, PA 15213 mmistry@disneyresearch.com
J. Buchli and S. Schaal are at the Computational Learning and Motor Control Lab, University of Southern California, Los Angeles, CA 90089 {buchli,sschaal}@usc.edu

base, and measured with respect to an inertial frame. Figure 1 illustrates this representation by showing the 6 virtual degrees of freedom attached from inertial frame to the robot base frame.

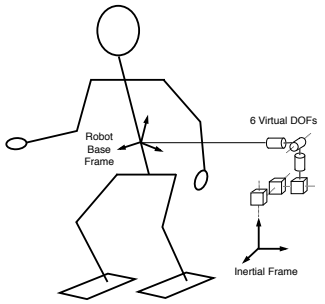

Fig. 1. The base frame attached to the robot is connected to the inertial frame via 6 unactuated virtual DOFs

When the robot is in contact with the environment, the equations of motion with respect to an inertial frame are given by:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T\tau + \mathbf{J}_C^T(\mathbf{q})\lambda \tag{2}$$

with variables defined as follows:

- $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n+6\times n+6}$: the floating base inertia matrix
- $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n+6}$: the floating base centripetal, Coriolis, and gravity forces.
- $\mathbf{S} = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times 6} \end{bmatrix}$: the actuated joint selection matrix
- $\tau \in \mathbb{R}^n$: the vector of actuated joint torques
- $\mathbf{J}_C \in \mathbb{R}^{k\times n+6}$: the Jacobian of k linearly independent constraints
- $\lambda \in \mathbb{R}^k$: the vector of k linearly independent constraint forces

By constraints, we mean locations on the robot in environmental contact, where external forces or torques are applied, and no motion is observed with respect to the inertial frame. Calling $\mathbf{x}_C$ the positions and orientations of these locations, we will have:

$$\dot{\mathbf{x}}_C = \mathbf{J}_C\dot{\mathbf{q}} = \mathbf{0} \tag{3}$$

$$\ddot{\mathbf{x}}_C = \mathbf{J}_C\ddot{\mathbf{q}} + \dot{\mathbf{J}}_C\dot{\mathbf{q}} = \mathbf{0} \tag{4}$$

Note that the structure and units of $\mathbf{J}_C$ and λ may depend on the particular kinematic structure used to model the robot. For example, a flat foot in surface contact with the ground can be equivalently represented as a plane contact (with 3 forces and 3 moments), or as a lattice of several point contacts. Without loss of generality, we assume that $\mathbf{J}_C$ and λ represent *k linearly independent* constraint forces. For example, a humanoid robot with one flat foot on the ground has 6 linearly independent constraints. With two flat feet on the ground, it will have 12 linearly independent constraints.

For the purpose of our model and derivation, we will assume that all constraints are multilateral, and friction is sufficient such that (3) and (4) hold for all desired motion. However, it is often acceptable for these assumptions to be violated in practice, as will be discussed in the next section.

III. Floating Base Inverse Dynamics Control

Given some desired joint motion we wish for our robot to perform $(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$, and a model of robot dynamics (2), we would like to compute the actuated joint torques, τ that will realize the desired motion. While such inverse dynamics control methods have been used extensively for fixed base robot manipulators, the problem is ill-posed for floating base systems. Solving for τ using (2) requires full knowledge of the constraint forces, λ. However, these constraint forces depend on the actuation torques applied. Due to this co-dependance between λ and τ, there are potentially infinitely many solutions for τ, and because of under-actuation, certain values of $\ddot{\mathbf{q}}_d$ may yield no solution at all.

A. Inverse Dynamics with Contact Force Feedback

In order to resolve these issues, we can attempt to project the equations of motion into actuated joint space [2]. If we assume that the contact points do not move with respect to the inertial frame (i.e. (3) and (4)), then contact forces can be uniquely determined in the following way. We multiply (2) by $\mathbf{J}_C\mathbf{M}^{-1}$, replace $\mathbf{J}_C\ddot{\mathbf{q}}$ with $\ddot{\mathbf{x}}_C - \dot{\mathbf{J}}_C\dot{\mathbf{q}}$, and solve for $\ddot{\mathbf{x}}_C$:

$$\ddot{\mathbf{x}}_C = \dot{\mathbf{J}}_C\dot{\mathbf{q}} - \mathbf{J}_C\mathbf{M}^{-1}\left(\mathbf{h} - \mathbf{S}^T\tau\right) + \mathbf{J}_C\mathbf{M}^{-1}\mathbf{J}_C^T\lambda \tag{5}$$

Thus the only λ for which $\ddot{\mathbf{x}}_C = 0$ is:

$$\lambda = \left(\mathbf{J}_C\mathbf{M}^{-1}\mathbf{J}_C^T\right)^{-1}\left(\mathbf{J}_C\mathbf{M}^{-1}\left(\mathbf{h} - \mathbf{S}^T\tau\right) - \dot{\mathbf{J}}_C\dot{\mathbf{q}}\right). \tag{6}$$

Inserting (6) into (2) and projecting the result into actuated joint space (by multiplying by $\bar{\mathbf{S}}^T = \left(\mathbf{S}\mathbf{M}^{-1}\mathbf{S}^T\right)^{-1}\mathbf{S}\mathbf{M}^{-1}$) results in the equations of constrained dynamics in actuated joint space:

$$\begin{aligned}\left(\mathbf{S}\mathbf{M}^{-1}\mathbf{S}^T\right)^{-1}\ddot{\mathbf{q}}_r + \bar{\mathbf{S}}^T\left(\mathbf{I} - \mathbf{J}_C^T\bar{\mathbf{J}}_C^T\right)\mathbf{h} + \bar{\mathbf{S}}^T\mathbf{M}\bar{\mathbf{J}}_C\dot{\mathbf{J}}_C\dot{\mathbf{q}} \\ = \bar{\mathbf{S}}^T\left(\mathbf{I} - \mathbf{J}_C^T\bar{\mathbf{J}}_C^T\right)\mathbf{S}^T\tau\end{aligned} \tag{7}$$

where $\bar{\mathbf{J}}_C^T = \left(\mathbf{J}_C\mathbf{M}^{-1}\mathbf{J}_C^T\right)^{-1}\mathbf{J}_C\mathbf{M}^{-1}$. Note that the result is independent of the constraint forces, λ. If the expression $\bar{\mathbf{S}}^T\left(\mathbf{I} - \mathbf{J}_C^T\bar{\mathbf{J}}_C^T\right)\mathbf{S}^T$ were invertible, it could be used to compute inverse dynamics torques. However, in general, this matrix is rank deficient and therefore not invertible.

In [9] we attempt to resolve this issue by treating λ as a vector of external forces. Again projecting (2) into actuated joint space, but replacing λ with $\mathbf{F}_{ext}$, we have

$$\left(\mathbf{S}\mathbf{M}^{-1}\mathbf{S}^T\right)^{-1}\ddot{\mathbf{q}}_r + \bar{\mathbf{S}}^T\left(\mathbf{h} - \mathbf{J}_C^T\mathbf{F}_{ext}\right) = \tau \tag{8}$$

Inverse dynamics torques can then be computed, if $\mathbf{F}_{ext}$ can be measured. However this approach is undesirable since force sensors must exist at all contact points, and are typically noisy and delayed if filtered. Alternatively we could estimate $\mathbf{F}_{ext}$ using (6). In this case, we need to use τ from the previous control time step, which will not produce an analytically correct solution, but may be acceptable with a fast enough control loop. Nevertheless, this approach is just

an approximate solution to floating base inverse dynamics control, and can go unstable in adverse situations.

B. Inverse Dynamics via Orthogonal Decomposition

Because of the problems that are created by working with contact forces directly, it would be advantageous to develop an inverse dynamics approach that is independent of them. Note that when our system moves within the null space of the constraints (i.e. (3) and (4) hold), the total dimensionality of our system is reduced by k. Therefore if we can represent our dynamics in this reduced dimensional space, we can eliminate k equations from the full rigid-body dynamics. We use an orthogonal decomposition approach in order to project the full system into this $n+6-k$ dimensional space. Orthogonal decomposition was first proposed for modal synthesis in [7] as a method of eliminating Lagrange multipliers for linear dynamical systems with linear constraints. In [10], a general linear projection operator is used to demonstrate how joint forces and constraint forces can be orthogonally decomposed (in the fully actuated case), and constraint forces can be eliminated (in the under-actuated case). In this work, we show how using an orthogonal decomposition provides a simple solution to the ill-posed problem of floating base inverse dynamics control, and can cope with the inherent problems of under-actuation, constraint switching, etc.

1) The QR decomposition: If we assume $Rank(\mathbf{J}_C) = k$ (i.e. there are k linearly independent constraints and $\mathbf{J}_C$ is full row rank), we can then compute the $\mathbf{QR}$ decomposition of $\mathbf{J}_C^T$:

$$\mathbf{J}_C^T = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{9}$$

where $\mathbf{Q}$ is orthogonal ($\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$), and $\mathbf{R}$ is an upper triangle matrix of rank k. Additionally, if $\mathbf{R}$ is restricted to have all positive diagonal elements, then $\mathbf{Q}$ and $\mathbf{R}$ are unique. Multiplying (2) by $\mathbf{Q}^T$ allows us to decompose the rigid-body dynamics into two independent equations:

$$\mathbf{S}_c\mathbf{Q}^T\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}\right) = \mathbf{S}_c\mathbf{Q}^T\mathbf{S}^T\tau + \mathbf{R}\lambda \tag{10}$$

$$\mathbf{S}_u\mathbf{Q}^T\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}\right) = \mathbf{S}_u\mathbf{Q}^T\mathbf{S}^T\tau \tag{11}$$

where

$$\mathbf{S}_c = \begin{bmatrix} \mathbf{I}_{k\times k} & \mathbf{0}_{k\times(n+6-k)} \end{bmatrix} \tag{12}$$

$$\mathbf{S}_u = \begin{bmatrix} \mathbf{0}_{(n+6-k)\times k} & \mathbf{I}_{(n+6-k)\times(n+6-k)} \end{bmatrix}, \tag{13}$$

are used to select the top and lower portions of the full equation. In Appendix A we show that if the motion remains within the nullspace of the constraints (i.e. (3) and (4) hold) then equations (10) and (11) will be independent, and we can represent the full dynamics of the robot with either equation. Note that (11) contains no dependance on constraint forces, yet still describes the full dynamics of the system.

2) Resolving under-actuation: Although the full rigid-body equation (2) may be under-actuated, the reduced $n+6-k$ dimensional space may not be. In [11] we show that if we segment the constraint Jacobian into its separate parts relating to base and joint motion:

$$\mathbf{J}_C = \begin{bmatrix} \partial\mathbf{x}_C/\partial\mathbf{q}_r & \partial\mathbf{x}_C/\partial\mathbf{x}_b \end{bmatrix}, \tag{14}$$

and that $Rank(\partial\mathbf{x}_C/\partial\mathbf{x}_b = 6)$, we can then represent the full configuration velocity vector $\dot{\mathbf{q}}$ using only $\dot{\mathbf{q}}_r$. This typically the case for most modern, high degree of freedom humanoid robots with one or two feet flat on the ground, or even with 2 edge or 3 point contacts, provided they are not collinear. As a consequence of this condition, the reduced dimensional space will not be under-actuated, and the matrix $\mathbf{S}_u\mathbf{Q}^T\mathbf{S}^T\tau \in \mathbb{R}^{(n+6-k)\times n}$ will be full row rank.

3) Control equation: We can now compute control torques using (11) and a pseudoinverse:

$$\tau = \left(\mathbf{S}_u\mathbf{Q}^T\mathbf{S}^T\right)^+ \mathbf{S}_u\mathbf{Q}^T\left[\mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{h}\right]. \tag{15}$$

It is straightforward to show that applying (15) to (2) will result in $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d$, provided (3) and (4) are not violated. Additionally, using the Moore-Penrose pseudoinverse ($A^+ = A^T\left(AA^T\right)^{-1}$), will compute the minimum norm torque vector that will achieve the desired motion. Note that here we assume that all robot actuators are rotary joints. If the robot contains a mixture of linear and rotary actuators, then the units of τ will be inhomogeneous, and a norm will not be well defined in the Euclidean metric space [12]. In this case, we can first project (2) into a metric space where units are homogeneous (using a symmetric positive definite matrix), and then take the orthogonal decomposition in this homogeneous metric space. Also note that the inhomogeneity inherent in $\mathbf{x}_b \in SE(3)$ (from (1)), does not affect (15) due to the use of $\mathbf{S}$.

4) Redistribution of control torques: Because a right pseudoinverse is used in (15), when $k > 6$ there are an infinite number of possible torques that can realize $\ddot{\mathbf{q}}_d$. By using a weighted psuedoinverse in (15), i.e.

$$A^{\#} = W^{-1}A^T\left(AW^{-1}A^T\right)^{-1} \tag{16}$$

where W is a positive definite matrix, we may be able to reduce the torque loads on specific joints, at the cost of increasing the total torque generated by the system (as well as the total contact force).

5) Computing contact forces: Also, we can still compute the contact forces, using (10):

$$\lambda = \mathbf{R}^{-1}\mathbf{S}_c\mathbf{Q}^T\left[\mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{h} - \mathbf{S}^T\tau\right]. \tag{17}$$

Note that compared with (6) the modeling error-prone inertia matrix is used only once, in a non-inverted form, and the time derivative of the constraint Jacobian ($\dot{\mathbf{J}}_C$) is not required.

6) Practical issues: Note that because $\mathbf{Q}$ is numerically determined (typically with Housholder transformations), it is possible that $\mathbf{Q}$ can exhibit discontinuities during tracking: an arbitrarily small change in $\mathbf{J}_C$ may result in a large change in $\mathbf{Q}$. However, as (15) is actually using $\mathbf{Q}$ twice, first to project into the unconstrained space, and then again to transform back into the original configuration space, a discontinuity in $\mathbf{Q}$ will not create a discontinuity in the expression $\left(\mathbf{S}_u\mathbf{Q}^T\mathbf{S}^T\right)^+\mathbf{S}_u\mathbf{Q}^T$ and corresponding control torque.

It is also important to note that (15) can only achieve $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d$ when (3) and (4) are true (i.e. our motion exists in

the null-space of the constraints, and the contact condition assumed by $\mathbf{J}_C$ is accurate). However, in the case that either of these conditions are violated (e.g. slippage at contact points or we incorrectly believe a foot is on the ground), joint tracking will degrade, but the system does not become singular, numerically unstable, etc. Thus this approach is feasible for real world situations with environmental uncertainties.

Finally, accurate tracking highly depends on the accuracy of your model (particularly the $\mathbf{M}$ and $\mathbf{h}$ terms in (15)). We have developed techniques for estimating inertial parameters from within the reduced dimensional space independent of constraint forces [13]. Doing so allows us to use data to better predict the actuation torques required for some desired motion, without requiring noisy (or unavailable) contact force sensing. However, it is important to note that the projection in (15) only relies on kinematic parameters, which are generally easier to determine than inertial parameters. The methods of [2]-[4] rely on inertia weighted projections, which can potentially compound modeling errors[1].

IV. Evaluations

In this section we show the feasibility of the inverse dynamics control approach on a simulated biped model as well as a real quadruped robot that walks on rough terrain. In both these floating base examples, we try to demonstrate compliance, dynamically changing contact conditions, and robustness to unforeseen disturbances.

A. Bipedal Humanoid Platform

We use the SL simulated bipedal robot, modeled after the lower half of the Sarcos Humanoid robot (figure (2))[14]. The simulated robot has 2 $\times$ 7 DOF legs and a 1 $\times$ 2 DOF torso, for a total of 16 DOF. Each robot foot is represented by 4 point contacts, and contact with the floor is simulated using a spring-damper contact model. In the simulator, the integration frequency is at 1000Hz, while the feedback control loop runs at a 500Hz cycle.

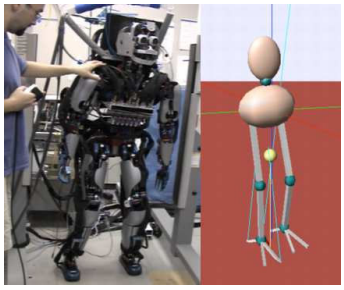

Fig. 2. Sarcos CBi robot and SL biped simulator

B. Joint Tracking with Feedforward Control

In our first example, we demonstrate how floating base inverse dynamics can improve simple joint tracking performance of the biped robot. We implement a *feedforward controller* [15]:

$$\tau = InvDyn(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_d) + \mathbf{K}_P\mathbf{S}\left(\mathbf{q}_d - \mathbf{q}\right) + \mathbf{K}_D\mathbf{S}\left(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}\right) \tag{18}$$

where $\mathbf{K}_P$ and $\mathbf{K}_D$ are position and velocity feedback gain matrices and $InvDyn()$ is computed via (15). Also, desired motion is planned to be in the nullspace of the constraints.

We plan a combination squatting/bowing motion in the sagittal plane via joint angles of combined sinusoids. In order to demonstrate robustness to a joint limit we also gradually extend the knee joint, until full extension. Figure 3 displays some snapshots of the resulting motion. Figure 4 shows tracking performance of our method (labeled "PD+FF"), compared to only high-gain PD control, for a relatively slow execution of the desired trajectories. Tracking is significantly improved using feedforward control, achieving RMS error values of 0.001, 0.011, and 0.005 for the hip, knee, and ankle flexion extension joints respectively. In comparison, PD only control had 0.008, 0.076, and 0.021 RMS error over the same desired motion.

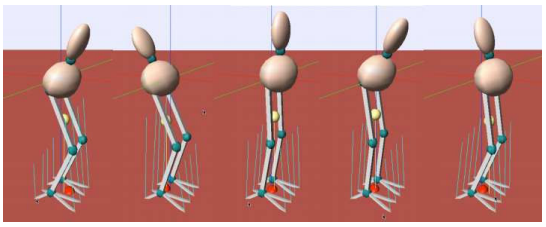

Fig. 3. Some configurations of the biped simulator during the squatting/bowing motion.

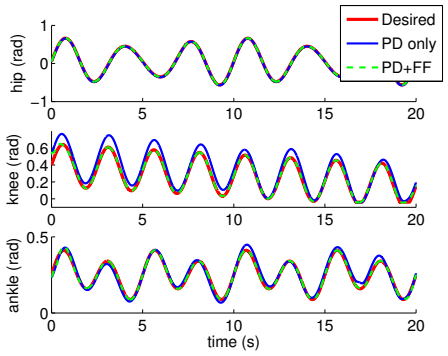

Fig. 4. Tracking performance of the feedforward controller and PD only controller during a slow squatting/bowing motion. Shown are the joint angles of the left hip, knee, and ankle flexion/extension joints.

We repeated the trajectories, 5 times faster (Fig. 5). PD only control was not able to keep the robot upright after 1 second. Note that in this example there is no explicit balance controller, however, compensating for the robot's inertia via floating base inverse dynamics significantly improved stability. The method was also able to maintain stability during full knee extension (starting roughly at the 4 second mark), although this resulted in a degradation of tracking performance due to hitting the joint range limits. The feedforward controller achieved tracking RMS errors

[1]However, using inertia weighted projections does have the benefit of unit and Gauge invariance [12]

of 0.005, 0.109, and 0.052 for the hip, knee, and ankle flexion/extension joints respectively.

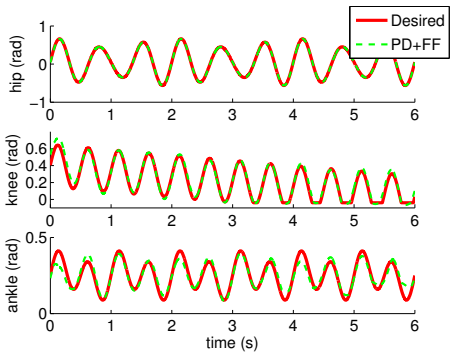

Fig. 5. Tracking performance of the feedforward controller during fast squatting/bowing motion. The PD only controller was not able to maintain stability after 1 second. Notice that the knee joint reaches full extension (after roughly 4 seconds).

C. Weighted Distribution of Forces

Next we show how we can use a weighting term to redistribute control forces. Imagine that the robot's left knee joint becomes damaged, and we wish to minimize the torque output required by this actuator. From (16), we set the weight matrix W to identity, except at the entry corresponding to the left knee joint, where we use 10^{-5}. We execute a squatting motion (2Hz sinusoids at hip, knee, and ankle flexion/extension joints), and measure the RMS torque over 6 seconds. As shown in Fig. 6, by using the weighting factor, we were able to achieve a 3.0 times reduction in left knee RMS torque, at the cost of a 1.9, 2.4, and 1.7 times increase in left ankle, right hip, and right knee (respectively).

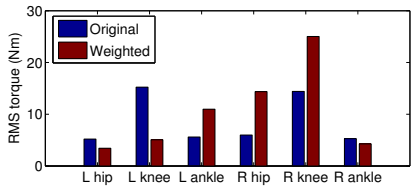

Fig. 6. RMS torque values for a squatting task when using the original ($W = \mathbf{I}$) formulation, and a weighed formulation intended to decrease the torque required by the left knee joint.

This redistribution of torque did not greatly impact tracking performance. Fig. 7 shows relatively small increases in tracking error when compared to PD only control.

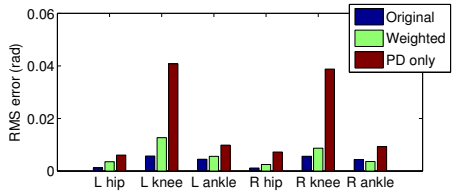

Fig. 7. RMS tracking errors for a squatting task. The Weighted formulation reduces the torque required at the left knee joint, without significantly effecting tracking performance.

D. Superposition of joint space and operational space control

Next we show how floating base inverse dynamics can be used to modify joint tracking control in order to achieve an operational space objective. In this case, we execute a periodic squatting motion in joint space, while maintaing stability using a feedforward balance controller in operational space. We also introduce a stepping pattern which demonstrates the balance controller's robustness to constraint switching, due to making and breaking contacts with the feet. Additionally, we perturb the balancing system with external forces.

We use the same feedforward controller as in (18), however for the desired joint acceleration (in the inverse dynamics computation) we use:

$$\ddot{\mathbf{q}}_d = \begin{bmatrix} \mathbf{J}_C \\ \mathbf{J}_{b,xy} \end{bmatrix}^+ \left(\begin{bmatrix} \mathbf{0} \\ \ddot{\mathbf{x}}_{b_d,xy} \end{bmatrix} - \begin{bmatrix} \dot{\mathbf{J}}_C \\ \dot{\mathbf{J}}_{b,xy} \end{bmatrix} \dot{\mathbf{q}} \right), \quad (19)$$

where $\mathbf{J}_{b,xy}$ is the Jacobian of the x and y components of the robot's base (assuming z is the direction of gravity), and $\ddot{\mathbf{x}}_{b_d,xy}$ is a desired task-space tracking trajectory for the x and y components of the robot's base. In this case, we simply define:

$$\ddot{\mathbf{x}}_{b_d,xy} = \mathbf{K}_P \left(\mathbf{x}_{b_d,xy} - \mathbf{x}_{b,xy} \right) - \mathbf{K}_D \dot{\mathbf{x}}_{b,xy} \quad (20)$$

Note that we augment the task Jacobian with the constraint Jacobian $\mathbf{J}_C$, and include a null vector, $\mathbf{0}$, in order to ensure that $\ddot{\mathbf{q}}_d$ remains consistent with the constraints. For a detailed discussion of this, and other issues related to floating base inverse kinematics, please see [11]. Additionally, we use the base Jacobian as an approximation to the COG Jacobian, which works well for our top-heavy biped system. Also, we intentionally do not include the z component of the base in (19). This way, we can control for the robot's balance, and a stretched knee posture does not introduce a singularity.

The desired base position $\mathbf{x}_{b_d}$ is set to alternate between the two feet, such that the COG lies above the support polygon of a single foot, and the opposing foot can be lifted. To lift the feet, we instantaneously change the desired joint offsets of the hip, knee, and ankle of a single leg. Also at this time, we provide for a constraint switch by instantly changing the constraint Jacobian (from double-support, to right foot, to double-support, to left foot, etc.). Doing so creates discontinuities in the torque output of (15), but does not adversely affect balance.

We test this controller with a 2Hz periodic squatting motion, superimposed with a 3 second period stepping pattern (alternating right and left feet). The robot bobs up and down while hopping from one foot to the other. Additionally every 5 seconds, we perturb the system with an external 200 N force at the base for 150ms. Resulting motion during disturbance and stepping is shown in Fig. 8, as well as in the accompanying video. Tracking performance of the base as well as leg joints is shown in Fig. 9. The simulated robot is able to maintain stability during disturbances and constraint switches.

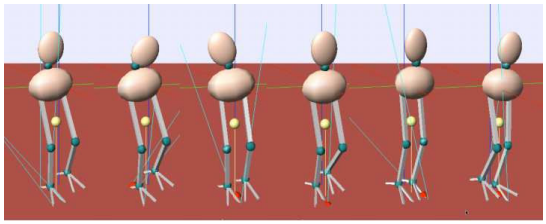

Fig. 8. Reaction of the biped during a 200 N external disturbance, and a constraint switch from the right to left foot. Time progresses from left frame to the right.

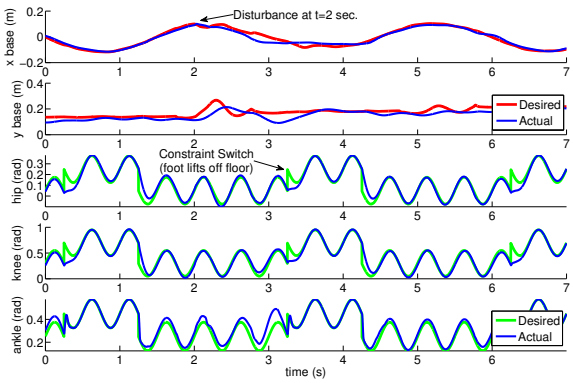

Fig. 9. Tracking results of the feedforward balance controller during the stepping task. The base (top two graphs) is controlled to sway side-to-side, to allow for the sequential lifting of alternate feet. Discontinuities in desired joint trajectories are due to an instant offset change to lift the feet. A t=5 a 200 N external disturbance pushes the robot for 150ms.

E. Quadruped locomotion

We also evaluated our floating base inverse dynamics controller on the LittleDog robot, a roughly 0.3 meter long and 0.2 meter tall robot dog that walks over rough terrain (Fig. 10). Please see [16] for a full description of this robot and its locomotion controller. To locomote successfully on rough terrain, this robot needs to accurately track its planned joint-space trajectories for maintaining balance and proper foot placement. However, a high level of compliance is also required to cope with the uncertainty of the environment. In order to have a compliant yet accurate trajectory controller, we use the feedforward controller of (18), with relatively low feedback gains.

We test the robustness of the controller towards unperceived obstacles of 2-4 cm height and a non-perceived rock board (Fig 10). With such unknown obstacles and terrain, a high gain PD controller alone is much too stiff to be able to reject the unplanned disturbances. However, by lowering gains, and adding feedforward inverse dynamics, we are able to maintain tracking accuracy while robustly giving into unseen obstacles. Please see the video attachment for a full demonstration.

Figure 11 (top) shows a typical example of joint tracking quality while the robot is traversing flat terrain. Figure 11 (bottom) shows the contributions of feedback and feedforward terms to the total control torque. The feedforward term is the main contributor for this motion, and only relatively small amounts of feedback are required for error correction.

Fig. 10. The LittleDog robot and various setups to test the robustness of the controller towards non-perceived obstacles and terrains. Please see the video supplement.

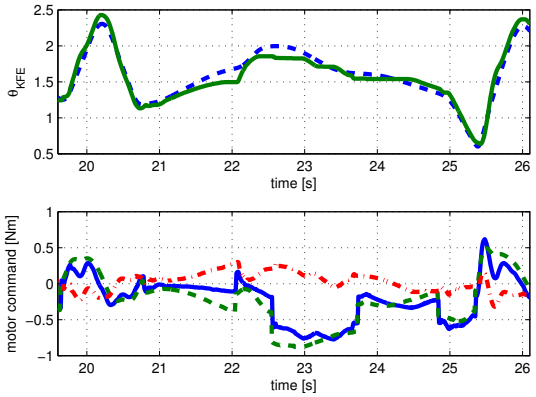

Fig. 11. Top: Tracking of the right hind knee of littledog in a steady walk (solid green is actual). Bottom: total control torque τ (solid blue) is the sum of feedback (red) and feedforward (green) torques. Feedforward torque is the main contributor most of the time. The effect of a constraint switch is clearly visible due to the discontinuity.

V. Conclusion

In this paper, we address the issue of compliant dexterous control of floating base robotic systems using model-based control. Because floating base systems have the added complexities of changing constraints when the robot interacts with the environment, potential closed-loop kinematics, and potentially unknown location and force of contacts, a general inverse dynamics control approach is needed that can flexibly accommodate constraints without the need to derive new analytical models for every new constraint situation. For this purpose, we have used orthogonal decomposition to reduce the dimensionality of our robotic system to only unconstrained degrees of freedom, that do not depend on constraint forces. In this way, we can find a solution to an ill-posed inverse dynamics problem. We are able to compute the analytically correct inverse dynamics torques for a sufficiently constrained, floating base rigid-body system, such as a humanoid or quadruped robot, without the need to know the constraint forces. This is a major advantage, since otherwise the contact forces must either be estimated, or measured with noisy sensors. The resulting formulation is surprisingly simple, and does not require complex formulations with inverted inertia matrices, which are prone to magnify modeling errors. Additionally, we can use this formulation to naturally combine operational space control with joint tracking control. For example, we can combine a central pattern generator for locomotion with a COG controller for balance. We have demonstrated the feasibility

of our approach with an evaluation on a simulated bipedal robot, and a real quadruped robot. We have shown how our approach improves the performance of tracking joint trajectories significantly and, thus, can help to keep systems compliant with low feedback gains. We have also shown that our approach can robustly handle constraint switching, balancing under disturbances, and the uncertainty of rough terrain. Our next step is the implementation on a real high dimensional force-controlled humanoid robot, to perform locomotion and manipulation tasks.

ACKNOWLEDGMENT

This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, the DARPA program on Learning Locomotion, a NASA grant AC98-516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

VI. APPENDIX A

In this appendix we show the independence of (10) and (11). We begin by multiplying (2) by $\mathbf{Q}^T$:

$$\mathbf{Q}^T\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Q}^T\mathbf{h} = \mathbf{Q}^T\mathbf{S}^T\tau + \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\lambda. \tag{21}$$

and define a new coordinate system $\mathbf{p}$ such that:

$$\dot{\mathbf{q}} = \mathbf{Q}\dot{\mathbf{p}}, \qquad \ddot{\mathbf{q}} = \mathbf{Q}\ddot{\mathbf{p}} + \dot{\mathbf{Q}}\dot{\mathbf{p}}. \tag{22}$$

We then rewrite (21) as:

$$\mathbf{Q}^T\mathbf{M}\mathbf{Q}\ddot{\mathbf{p}} + \mathbf{Q}^T\mathbf{M}\dot{\mathbf{Q}}\dot{\mathbf{p}} + \mathbf{Q}^T\mathbf{h} = \mathbf{Q}^T\mathbf{S}^T\tau + \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\lambda, \tag{23}$$

or in more compact notation:

$$\hat{\mathbf{M}}\ddot{\mathbf{p}} + \hat{\mathbf{M}}'\dot{\mathbf{p}} + \hat{\mathbf{h}} = \mathbf{D}^T\tau + \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\lambda, \tag{24}$$

where we define:

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{Q}^T\mathbf{M}\mathbf{Q}, \quad \hat{\mathbf{M}}' = \mathbf{Q}^T\mathbf{M}\dot{\mathbf{Q}}, \\ \hat{\mathbf{h}} &= \mathbf{Q}^T\mathbf{h}, \qquad \mathbf{D} = \mathbf{S}\mathbf{Q}. \end{aligned} \tag{25}$$

Next, we decompose $\mathbf{p}$ as follows:

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_c \\ \mathbf{p}_u \end{bmatrix}, \tag{26}$$

where $\mathbf{p}_c \in \mathbb{R}^k$ and $\mathbf{p}_u \in \mathbb{R}^{n+6-k}$, and expand (24) as:

$$\begin{bmatrix} \hat{\mathbf{M}}_{cc} & \hat{\mathbf{M}}_{cu} \\ \hat{\mathbf{M}}_{uc} & \hat{\mathbf{M}}_{uu} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{p}}_c \\ \ddot{\mathbf{p}}_u \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{M}}'_{cc} & \hat{\mathbf{M}}'_{cu} \\ \hat{\mathbf{M}}'_{uc} & \hat{\mathbf{M}}'_{uu} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{p}}_c \\ \dot{\mathbf{p}}_u \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{h}}_c \\ \hat{\mathbf{h}}_u \end{bmatrix} = \begin{bmatrix} \mathbf{D}_c^T \\ \mathbf{D}_u^T \end{bmatrix}\tau + \begin{bmatrix} \mathbf{R}\lambda \\ \mathbf{0} \end{bmatrix}. \tag{27}$$

If $\dot{\mathbf{q}}$ is in the null space of $\mathbf{J}_C$:

$$\mathbf{J}_C\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \end{bmatrix}\mathbf{Q}^T\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{p}}_c \\ \dot{\mathbf{p}}_u \end{bmatrix} = \mathbf{0}, \tag{28}$$

implying that $\dot{\mathbf{p}}_c = \mathbf{0}$. If we differentiate (28):

$$\begin{bmatrix} \mathbf{R}^T & \mathbf{0} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{p}}_c \\ \ddot{\mathbf{p}}_u \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{R}}^T & \mathbf{0} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{p}}_c \\ \dot{\mathbf{p}}_u \end{bmatrix} = \mathbf{0}. \tag{29}$$

Since $\dot{\mathbf{p}}_c = \mathbf{0}$, we must also have $\ddot{\mathbf{p}}_c = \mathbf{0}$. Finally we can conclude that:

$$\dot{\mathbf{p}}_c = \ddot{\mathbf{p}}_c = \mathbf{0}, \qquad \mathbf{p}_c = \mathbf{k} \tag{30}$$

where $\mathbf{k}$ is some constant vector (it is unknown, but irrelevant). Because of (30), the upper and lower equations of (27) are decoupled, and we can write the complete dynamics of our system without any constraint forces as:

$$\hat{\mathbf{M}}_{uu}\ddot{\mathbf{p}}_u + \hat{\mathbf{M}}'_{uu}\dot{\mathbf{p}}_u + \hat{\mathbf{h}}_u = \mathbf{D}_u^T\tau \tag{31}$$

Subsequently, we can write our control equation as:

$$\tau = \left(\mathbf{D}_u^T\right)^+\left[\hat{\mathbf{M}}_{uu}\ddot{\mathbf{p}}_u + \hat{\mathbf{M}}'_{uu}\dot{\mathbf{p}}_u + \hat{\mathbf{h}}_u\right] \tag{32}$$

where $(.)^+$ is the right pseudoinverse ($A^+ = A^T\left(AA^T\right)^{-1}$).

Since (32) requires a $\dot{\mathbf{Q}}$ term, which may be difficult to determine analytically, we can substitute (22) back into (32), and use (13) to write the control equation as:

$$\tau = \left(\mathbf{S}_u\mathbf{Q}^T\mathbf{S}^T\right)^+\mathbf{S}_u\mathbf{Q}^T\left[\mathbf{M}\ddot{\mathbf{q}}_d + \mathbf{h}\right] \tag{33}$$

REFERENCES

[1] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, Dec 2008.

[2] L. Sentis and O. Khatib, "Control of free-floating humanoid robots through task prioritization," *ICRA 2005*, pp. 1718–1723, 2005.

[3] J.-H. Park and O. Khatib, "Contact consistent control framework for humanoid robots," *ICRA 2006*, pp. 1963–1969, 2006.

[4] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[5] S.-H. Hyon, J. Hale, and G. Cheng, "Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[6] S.-H. Hyon, R. Osu, and Y. Otaka, "Integration of multi-level postural balancing on humanoid robots," *ICRA 2009*, pp. 1549–1556, 2009.

[7] H. Flashner, "An orthogonal decomposition approach to modal synthesis," *International journal for numerical methods in engineering*, vol. 23, no. 3, pp. 471–493, Dec 1986.

[8] F. Aghili, "Inverse and direct dynamics of constrained multibody systems based on orthogonal decomposition of generalized force," *ICRA 2003*, pp. 4035 – 4041, 2003.

[9] J. Nakanishi, M. Mistry, and S. Schaal, "Inverse dynamics control with floating base and constraints," *ICRA 2007*, pp. 1942–1947, 2007.

[10] F. Aghili, "A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 834 – 849, Oct 2005.

[11] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," *8th IEEE-RAS International Conference on Humanoid Robots*, pp. 22–27, 2008.

[12] K. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.

[13] M. Mistry, S. Schaal, and K. Yamane, "Inertial parameter estimation of floating base humanoid systems using partial force sensing," *9th IEEE-RAS International Conference on Humanoid Robots*, 2009.

[14] G. Cheng, S.-H. Hyon, J. Morimoto, A. Ude, J. Hale, G. Colvin, W. Scroggin, and S. Jacobsen, "Cb: a humanoid research platform for exploring neuroscience," *Advanced Robotics*, vol. 21, no. 10, pp. 1097–1114, Oct 2007.

[15] A. Liegeois, A. Fournier, and M. Aldon, "Model reference control of high-velocity industrial robots." *Proc. Joint Automatic Control Conf.*, 1980.

[16] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrian," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.