2008 8th IEEE-RAS International Conference on Humanoid Robots
December 1 ~ 3, 2008 / Daejeon, Korea

Inverse Kinematics with Floating Base and Constraints for Full Body Humanoid Robot Control

Michael Mistry[1,2], Jun Nakanishi[1,4], Gordon Cheng, and Stefan Schaal[1,2,3]
[1]*ATR Computational Neuroscience Laboratories, Kyoto 619-0228, Japan*
[2]*Computer Science &* [3]*Neuroscience, University of Southern California, Los Angeles, CA 90089, USA*
[4]*ICORP, Japan Science and Technology Agency, Saitama 332-0012, Japan*
mmistry@atr.jp, jun@atr.jp, gordon.cheng@ieee.org, sschaal@usc.edu

***Abstract*— This paper explores inverse kinematics for full body, floating base, task space control on a real humanoid robot. We discuss how constraints can be used to address the issue of under-actuation due to floating base, and list the sufficient conditions for maintaining task space control of arbitrary robot tasks. We suggest a controller based on a task priority framework and demonstrate the feasibility of the approach on the SARCOS/ATR CBi humanoid robot. We implement examples of position control via constrained floating base inverse kinematics as first approach to full body model-based inverse dynamics control. The examples demonstrate center of gravity position tracking as well as hand figure-8 tracking while simultaneously balancing.**

I. Introduction

Because most humanoid robots are not physically attached to the world, as are fixed manipulators, the floating base framework provides a more general representation for humanoid robot control. We attach a base reference frame to the robot, and assume that the base frame can freely move relative to a fixed inertial frame. We can then completely describe the configuration of a rigid body robot, with respect to the inertial frame, 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 $\mathbf{x}_b \in \mathbb{R}^6$ is the position and orientation of the coordinate system attached to the robot base, and measured with respect to the fixed inertial frame. Figure 1 illustrates this representation by showing the 6 virtual degrees of freedom attached from inertial frame to the robot base frame.

Floating-base kinematics and dynamics have been extensively analyzed for spacecraft control [1]-[4], and in recent years for the control of humanoid robots [5]-[8] under the framework of Khatib's operational space control [9]. The advantage of Khatib's framework is that it can compute the control forces that decouple task and null-space dynamics. Additionally, as it is a force control method, it does not use joint position references, and thus does not require explicit inverse kinematics computation. Because these methods do not explicitly control for joint positions, they can maintain high level of compliance and thus are suitable for use in environments with human interaction.[1]

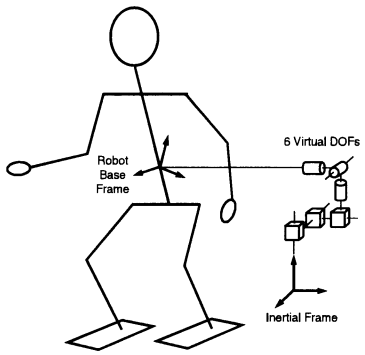

Fig. 1. The base frame attached to the robot is connected to the inertial frame via 6 unactuated virtual DOFs

In [10] we conducted an extensive theoretical and empirical evaluation of task space controllers for compliant control of fixed-based redundant manipulators. The types of controllers examined included velocity, acceleration and force based methods, where velocity and acceleration based methods utilize explicit inverse kinematics to compute joint references before feeding into a inverse dynamics computation. We found that the force control methods were highly sensitive to rigid-body dynamics model inaccuracies, in particular the inertia matrix which must be inverted. In this particular evaluation, we found that a joint acceleration-based controller coupled with inverse dynamics, had slightly better performance in light of modeling errors.

The goal of this paper is to compliment the force control work of [5]-[8], by focusing on the specific issues related to inverse kinematics for floating base systems in order to provide alternate control solutions. In particular, if we have a methodology for inverse kinematics computation for floating base humanoids, we can then easily utilize velocity or acceleration based task-space control coupled with floating base inverse dynamics [20], as well as traditional

[1]There is also evidence that humans do not explicitly control their joint positions in arm reaching tasks, see [11].

978-1-4244-2822-9/08/$25.00 ©2008 IEEE

joint position control. These methods, which maintain joint position references, can potentially provide a more robust solution in light of dynamic model inaccuracies. Figure 2 is a high level overview of a potential control framework. If we have an accurate inverse dynamics model, we can operate the feedback controller at near zero gain in order to keep compliance high. However in practice, dynamics models are rarely accurate, so we still use a moderate feedback gain to maintain robustness at the cost of higher stiffness. Additionally, pure position based feedback control can be very useful for learning system dynamics, since we can execute trajectories without any prior dynamics knowledge, in order to collect data for building dynamics models [12].

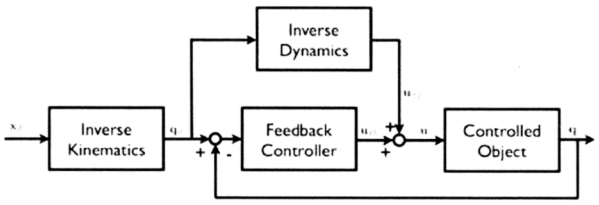

Fig. 2. Inverse Kinematics/Inverse Dynamics control diagram. $\mathbf{x}_d$ is a desired task, $\mathbf{u}_{ff}$ is feedforward torque and $\mathbf{u}_{fb}$ is feedback torque

In this paper we first introduce the floating base Jacobians and other notation we will use. We then provide and prove an inverse kinematics equation, and the sufficient conditions, that allow for the control of arbitrary task variables, regardless of under-actuation of floating base. We then suggest a controller which controls for static balance, as well as other tasks, while maintaining environmental constraints. Finally we demonstrate the feasibility of the controller on an actual humanoid robot platform.

II. Floating Body Kinematics

In this section we will define the necessary notation and the Jacobians required for floating base inverse kinematics. We will also discuss the role of constraints in the control of unactuated degrees of freedom.

A. Floating Base Jacobians

If $\mathbf{x} = \mathbf{f}(\mathbf{q})$ is the position and orientation of any frame located on the robot, and represented in the fixed (inertial) world coordinate system, the velocity of $\mathbf{x}$ is given by the equation $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{q}_r} & \frac{\partial \mathbf{x}}{\partial \mathbf{x}_b} \end{bmatrix} = \begin{bmatrix} \mathbf{J}^{\partial \mathbf{q}_r} & \mathbf{J}^{\partial \mathbf{x}_b} \end{bmatrix}, \tag{2}$$

and $\mathbf{J}^{\partial \mathbf{q}_r}$ and $\mathbf{J}^{\partial \mathbf{x}_b}$ are the components of $\mathbf{J}$ relating to robot joint and base motion, respectively. Each component of the Jacobian can be computed geometrically [22]. For example, end-effector motion relative to base motion is

$$\mathbf{J}^{\partial \mathbf{x}_b} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}_b} = \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{R}_b \times \left(\mathbf{x}^P - \mathbf{x}_b^P\right) \\ \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \end{bmatrix} \tag{3}$$

where $\mathbf{x}^P \in \mathbb{R}^3$ are the three positional components of $\mathbf{x}$ and $\mathbf{R}_b \times$ is a column-wise cross product of $\mathbf{R}_b$, the rotation matrix representing base orientation relative to the fixed world frame. Other important free-floating base Jacobians to consider are the Jacobian describing motion of the base itself:

$$\mathbf{J}_b = \begin{bmatrix} \mathbf{0}_{6\times n} & \mathbf{I}_{6\times 6} \end{bmatrix} \tag{4}$$

as well as the center of gravity of the robot [21]:

$$\mathbf{J}_{\text{cog}} = \begin{bmatrix} \frac{\partial \mathbf{x}_{\text{cog}}^P}{\partial \mathbf{q}_r} & \mathbf{I}_{3\times 3} & \mathbf{R}_b \times \left(\mathbf{x}_{\text{cog}}^P - \mathbf{x}_b^P\right) \end{bmatrix} \tag{5}$$

where $\mathbf{x}_{\text{cog}}^P$ is the position of the robot's center of gravity, and

$$\frac{\partial \mathbf{x}_{\text{cog}}^P}{\partial \mathbf{q}_r} = \frac{\sum_i m_i \mathbf{J}_{i,\text{cog}}}{\sum_i m_i} \tag{6}$$

is the mass weighted average of the Jacobians of COGs of individual links.

B. Inverse Kinematics with Floating Base

Task space control of robots using floating base Jacobians is complicated due to the addition of unactuated degrees of freedom. Computing joint velocities via a traditional Jacobian pseudo-inverse approach will create velocities for the base position and orientation which may not be realizable due to under-actuation. However, if we sufficiently constrain our system, we will be able fully realize endeffector velocities, provided certain conditions are met. Calling $\mathbf{x}_C \in \mathbb{R}^k$ the vector of k constraints, and $\mathbf{J}_C$ the Jacobian of constraints ($\dot{\mathbf{x}}_C = \mathbf{J}_C\dot{\mathbf{q}}$), using the following equation, we can compute the required actuated joint velocities ($\dot{\mathbf{q}}_r$) that will realize an arbitrary desired end effector velocity ($\dot{\mathbf{x}}_d$):

$$\dot{\mathbf{q}}_r = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times 6} \end{bmatrix} \begin{bmatrix} \mathbf{J}_C \\ \mathbf{J} \end{bmatrix}^+ \begin{bmatrix} \mathbf{0}_{k\times 1} \\ \dot{\mathbf{x}}_d \end{bmatrix}, \tag{7}$$

where $A^+ = A^T\left(AA^T\right)^{-1}$ is the right pseudoinverse, provided the following sufficient conditions are satisfied:

1) $\dot{\mathbf{x}}_C = \mathbf{0}$ is maintained (there is sufficient friction and stiffness at the constraint locations to prevent motion relative to the inertial frame).
2) The system is not over-constrained: The matrix $\begin{bmatrix} \mathbf{J}_C^T & \mathbf{J}^T \end{bmatrix}^T$ should remain full row rank.
3) The system is not under-constrained: Recalling the notation introduced in (2) ($\mathbf{J}_C = \begin{bmatrix} \mathbf{J}_C^{\partial \mathbf{q}_r} & \mathbf{J}_C^{\partial \mathbf{x}_b} \end{bmatrix}$), the constraint Jacobian related to base motion ($\mathbf{J}_C^{\partial \mathbf{x}_b}$), must have a rank equal to 6.

Proof: If $\begin{bmatrix} \mathbf{J}_C^T & \mathbf{J}^T \end{bmatrix}^T$ is full row rank (condition 2), its right pseudoinverse exists, and we can compute $\dot{\mathbf{q}}$ as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_C \\ \mathbf{J} \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{J}_C \\ \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{J}_C \\ \mathbf{J} \end{bmatrix}^T \right)^{-1} \begin{bmatrix} \mathbf{0}_{k\times 1} \\ \dot{\mathbf{x}}_d \end{bmatrix}. \tag{8}$$

Assuming that this $\dot{\mathbf{q}}$ can be fully realized by some controller, and contact points do not slip (condition 1), it is trivial to verify that $\dot{\mathbf{x}} = \dot{\mathbf{x}}_d$ and $\dot{\mathbf{x}}_C = \mathbf{0}$ (the task is successfully completed while constraints are maintained). Additionally, because of (8), $\dot{\mathbf{q}}$ will be in the null space of the constraints, and we can write:

$$\mathbf{J}_C\dot{\mathbf{q}} = \mathbf{J}_C^{\partial \mathbf{q}_r}\dot{\mathbf{q}}_r + \mathbf{J}_C^{\partial \mathbf{x}_b}\dot{\mathbf{x}}_b = \mathbf{0}. \tag{9}$$

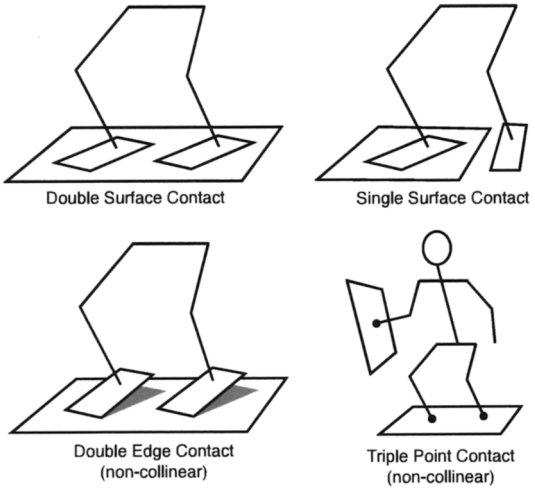

Fig. 3. Examples of systems that are adequately constrained: $Rank\left(\mathbf{J}_C^{\partial \mathbf{x}_b}\right) = 6$

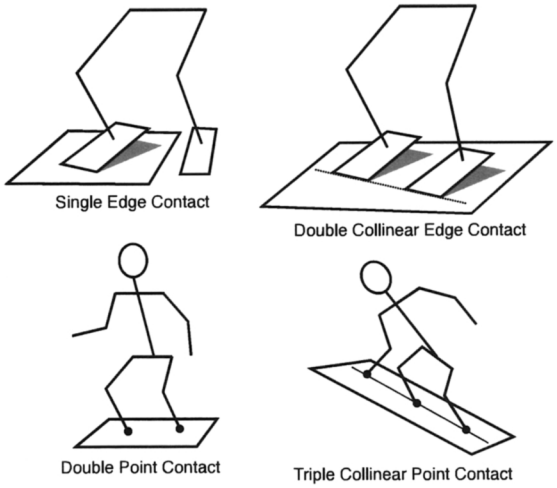

Fig. 4. Examples of systems that are under-constrained: $Rank\left(\mathbf{J}_C^{\partial \mathbf{x}_b}\right) < 6$

We can then solve for $\dot{\mathbf{x}}_b$:

$$\dot{\mathbf{x}}_b = -\left[\mathbf{J}_C^{\partial \mathbf{x}_b}\right]^+ \mathbf{J}_C^{\partial \mathbf{q}_r} \dot{\mathbf{q}}_r. \tag{10}$$

Only when (10) has a unique solution will we be able to represent $\dot{\mathbf{q}}$ unambiguously using only $\dot{\mathbf{q}}_r$. Otherwise, for a fixed $\dot{\mathbf{q}}_r$, $\mathbf{x}_b$ may be able to move independently. Note that $\mathbf{J}_C^{\partial \mathbf{x}_b} \in \mathbb{R}^{k \times 6}$. If $k < 6$, (10) has potentially an infinite number of solutions. If $k \geq 6$ and $Rank\left(\mathbf{J}_C^{\partial \mathbf{x}_b}\right) = 6$ a unique solution is guaranteed [24][2], and given by the left pseudo-inverse ($A^+ = \left(A^T A\right)^{-1} A^T$). Finally, when $Rank\left(\mathbf{J}_C^{\partial \mathbf{x}_b}\right) = 6$, we can represent $\dot{\mathbf{q}}$ unambiguously as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{I}_{n \times n} & -\left[\mathbf{J}_C^{\partial \mathbf{x}_b T} \mathbf{J}_C^{\partial \mathbf{x}_b}\right]^{-1} \mathbf{J}_C^{\partial \mathbf{x}_b T} \mathbf{J}_C^{\partial \mathbf{q}_r} \end{bmatrix} \dot{\mathbf{q}}_r. \tag{11}$$

■

Because of condition 3), the structure of the $\mathbf{J}_C^{\partial \mathbf{x}_b}$ determines whether or not our system is controllable with respect to an arbitrary task point on the robot. The minimum number of constraints required is 6, however, their relative positions are also critical. For example, two edge contacts are adequate provided they are not collinear. Figures 3 and 4 illustrate some examples of adequately constrained and under-constrained configurations, respectively.

It is important to note that when constraints change (for example when feet make and break contact with the ground during locomotion), $\mathbf{J}_C$ will abruptly change structure (possibly size as well, if new constraints are added or removed), creating undesirable discontinuities in $\dot{\mathbf{q}}_r$ at the moment of constraint switch. Methods for reducing these discontinuities for locomotion are addressed in [19].

[2]Assuming at least one solution exists, which we are able to do, since we also assume (9) is true.

III. Controller Formulation

The high number of DOFs of humanoids allow for a variety of tasks to be completed simultaneously, for example locomotion while manipulating a tool and orienting the head to look at an object. However, in order to maintain stability and controllability of the robot, it is critical to manage the different tasks in an appropriate hierarchical manner. Therefore, as similarly done in [7], we categorize our robot's tasks into one of three prioritization groups:

1) *Critical tasks*: essential tasks which much succeed. All other tasks depend on the success of these tasks.
2) *Functional tasks*: tasks we want to accomplish with a high level of accuracy, but their success or failure generally does not effect other tasks. This group could potentially have additional sub-levels of hierarchy within it.
3) *Extraneous tasks*: These tasks are used for resolving any remaining redundancy (such that the robot does not drift in null-space). Success of these tasks is generally irrelevant.

Once categorized, these tasks will be controlled via a task prioritization framework.

As emphasized in the previous section, in order to be able to use (7) to control arbitrary tasks, it is critical that $\dot{\mathbf{x}}_C = 0$ at all times. If a contact point ever has a nonzero velocity, then the robot's base will undergo motion independent of the joints, and as a consequence we lose the controllability of all tasks. Thus, when controlling the robot, the most critical task is to maintain zero velocity at all contact points. Secondly, balance is obviously critical, since a fall can be disastrous. Thus we will always place emphasis on static stability (maintaining the COG over the support polygon). For the purpose of this paper, we will always assume that the robot is in double surface contact with the environment

(both feet are flat on the floor), and we define our critical task Jacobian as:

$$\mathbf{J}_A = \begin{bmatrix} \mathbf{J}_{\text{rightfoot}} \\ \mathbf{J}_{\text{leftfoot}} \\ \mathbf{J}_{\text{cog,xy}} \end{bmatrix}, \tag{12}$$

where $\mathbf{J}_{\text{rightfoot}}$ and $\mathbf{J}_{\text{leftfoot}}$ are the Jacobians for the position and orientations of the 2 feet, and $\mathbf{J}_{\text{cog,xy}}$ is the Jacobian for the projection of the COG onto the XY plane (assuming the gravity vector is in the Z direction). Since we plan to use a matrix psuedoinverse for control, as with (7), it is critical for robot safety that we avoid any singularities of $\mathbf{J}_A$. Singularities will occur when two or more tasks are in conflict, i.e when the task of controlling the XY position of the COG conflicts with the task of maintaining the position of either foot. This particular situation would only occur when the robot is fully extended across the floor (i.e the robot has already fallen), or has its legs in a full split position. Thus in practice, as long as the robot is upright, we have no real concerns of $\mathbf{J}_A$ reaching a singularity. Note that fully extended knees are not a singular configuration in this case, and balance can still be maintained (provided the ankle joints have sufficient torque).

For a functional task we may want to control hand motion, for example, and thus we define the functional task Jacobian as:

$$\mathbf{J}_B = [\mathbf{J}_{\text{righthand}}], \tag{13}$$

where $\mathbf{J}_{\text{righthand}}$ is the Jacobian for the right hand.

Finally we define our lowest priority (extraneous) tasks as:

$$\mathbf{J}_E = \begin{bmatrix} \mathbf{J}_{\text{rpy}} \\ \mathbf{J}_{\text{cog,z}} \end{bmatrix} \tag{14}$$

where $\mathbf{J}_{\text{rpy}}$ dictates base orientation (in roll, pitch, yaw angles), and $\mathbf{J}_{\text{cog,z}}$ for the height of the COG above ground. Note that it is necessary to control for these variables in order to roughly maintain some desired posture, and prevent the robot from drifting in null-space.

Next we will define the vectors of desired end effector velocities, noting that $\dot{\mathbf{x}}_{\text{rightfoot}} = \dot{\mathbf{x}}_{\text{leftfoot}} = \mathbf{0}$ in order to maintain constraints:

$$\dot{\mathbf{x}}_A = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{x}}_{\text{cog,xy}} \end{bmatrix}, \; \dot{\mathbf{x}}_B = [\dot{\mathbf{x}}_{\text{righthand}}], \; \dot{\mathbf{x}}_E = \begin{bmatrix} \dot{\mathbf{x}}_{\text{rpy}} \\ \dot{\mathbf{x}}_{\text{cog,z}} \end{bmatrix}. \tag{15}$$

Finally, we will compute the desired joint velocities as:

$$\dot{\mathbf{q}}_r = \begin{bmatrix} \mathbf{I}_{n\times n} \\ \mathbf{0}_{6\times n} \end{bmatrix}^T \left(\mathbf{J}_A^+ \dot{\mathbf{x}}_A + \mathbf{N}_A \mathbf{J}_B^+ \dot{\mathbf{x}}_B + \mathbf{N}_{AB} \mathbf{J}_E^+ \dot{\mathbf{x}}_E\right), \tag{16}$$

where $\mathbf{N}_A = \mathbf{I} - \mathbf{J}_A^+ \mathbf{J}_A$ is the null space projection matrix of $\mathbf{J}_A$ and

$$\mathbf{N}_{AB} = \mathbf{I} - \begin{bmatrix} \mathbf{J}_A \\ \mathbf{J}_B \end{bmatrix}^+ \begin{bmatrix} \mathbf{J}_A \\ \mathbf{J}_B \end{bmatrix} \tag{17}$$

projects into the intersection of the $\mathbf{J}_A$ and $\mathbf{J}_B$ null spaces. Note that we conduct task prioritization via null space projections, which as a consequence does not track secondary tasks as accurately as other task prioritization frameworks [18]. However (16) uses the only task prioritization framework that will guarantee the success of $\dot{\mathbf{x}}_A$ (our critical tasks) for all time (except at rare singularities of $\mathbf{J}_A$, as discussed previously) and does not suffer from the problem of algorithmic singularities as the other approaches do [15],[16], or require a less accurate damped pseudoinverse [17]. For more discussion of these issues in relation to floating base kinematics, please refer to [19].[3]

Extending (16) to the 2nd order acceleration case, we have:

$$\ddot{\mathbf{q}} = \mathbf{J}_A^+ \left(\ddot{\mathbf{x}}_A - \dot{\mathbf{J}}_A \dot{\mathbf{q}}\right) + \mathbf{N}_A \mathbf{J}_B^+ \left(\ddot{\mathbf{x}}_B - \dot{\mathbf{J}}_B \dot{\mathbf{q}}\right) + \mathbf{N}_{AB} \mathbf{J}_E^+ \left(\ddot{\mathbf{x}}_E - \dot{\mathbf{J}}_E \dot{\mathbf{q}}\right) \tag{18}$$

and $\ddot{\mathbf{q}}_r = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times 6} \end{bmatrix} \ddot{\mathbf{q}}$. Ideally, we would like to use (18) in conjunction with floating base inverse dynamics in order to compute joint torques [20] .

After computing $\dot{\mathbf{q}}_r$ from (16), $\mathbf{q}_r$ can be computed via numerical integration:

$$\mathbf{q}_r^t = \mathbf{q}_r^{t-1} + \Delta T \dot{\mathbf{q}}_r \tag{19}$$

where t is the current time step, and ΔT is the control cycle period. Similarly, if equation (18) is used:

$$\ddot{\mathbf{q}}_r^t = \begin{bmatrix} \mathbf{I}_{n\times n} & \mathbf{0}_{n\times 6} \end{bmatrix} \ddot{\mathbf{q}} \tag{20}$$

$$\dot{\mathbf{q}}_r^t = \dot{\mathbf{q}}_r^{t-1} + \Delta T \ddot{\mathbf{q}}_r^t \tag{21}$$

$$\mathbf{q}_r^t = \mathbf{q}_r^{t-1} + \Delta T \dot{\mathbf{q}}_r \tag{22}$$

Note that (18) requires knowledge of $\dot{\mathbf{q}}$, which is often obtained via noisy sensors. In order to ensure that $\dot{\mathbf{q}}$ is consistent with the constraints, we either need to use $\dot{\mathbf{q}}^{t-1}$ computed from the previous time step via integration, or use current measured or estimated values filtered through the null space of the constraints ($\mathbf{N}_A$).

IV. Evaluations

A. Platform

We evaluate our controller on the CBi humanoid robot (figure (5)), developed by SARCOS Corporation and ATR Computational Neuroscience Laboratories [23]. CBi is a full-body, 51 degree of freedom humanoid (2 × 2 DOF eyes, 1 × 3 DOF neck, 2 × 7 DOF arms, 2 × 7 DOF legs, 1 × 3 DOF torso, 2 × 6 DOF fingers, 1 × 1 DOF jaw). CBi is approximately 157.5 cm in height and 87 kg in weight. All DOFs except eyes, fingers, and jaw, are hydraulically actuated. The eyes and jaw use electric motors and the fingers use pneumatics. Each joint has local position, velocity, and torque control available at a 5 kHz rate. Position, velocity, and torque set points, are computed on an off-board computer, and communicated via Ethernet at a 500 Hz bandwidth.

[3] Also, because technically $\mathbf{J}_A$ and $\mathbf{J}_B$ could conflict, we should replace $\mathbf{N}_{AB}$ with $\mathbf{N}_A \mathbf{N}_{AB}^{\lambda}$ where $\mathbf{N}_{AB}^{\lambda}$ uses a damping for the pseudo-inverse, see [19] for more detail.

Fig. 5. SARCOS/ATR CBi Humanoid Robot

B. COG Tracking

In the first experiment we attempt COG tracking control using (16) along with (19) and high gain joint position control on the robot. Although a support harness exists for safety, it does not bare any load, and the robot is fully standing and balancing by its own power. The projection of the COG on to the floor is given a side-to-side sinusoidal trajectory to track (shown in green) in figure 6. The front-back COG position (lower graph) is commanded to remain in a fixed position. Actual COG positions are shown in blue. The video associated with this paper demonstrates the performance.

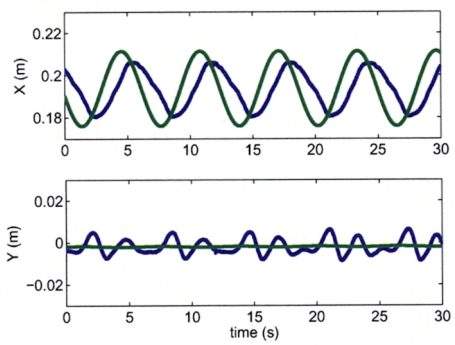

Fig. 6. Tracking of the COG position in the horizontal plane. The top graph shows the left-right sinusoidal target (green) and actual position (blue). The bottom graph shows the front-back tracking performance.

C. Hand Tracking with Simultaneous Balance

Next we attempt to track a hand trajectory, while simultaneously maintaining static balance (also using joint position control). The COG is controlled to remain at a fixed position within the support polygon. The right hand position is given a figure-8 trajectory to track in the perpendicular plane. Tracking performance in a time series graph is shown in figure 7, and in the plane in figure 8. Note that the controller tracks the desired hand trajectory in the null space of the COG controller, and therefore is less accurate. However, the arm motion theoretically does not interfere with the static balance of the robot. COG tracking during this experiment is shown in figure 9. Please see the video associated with this paper for the full motion.

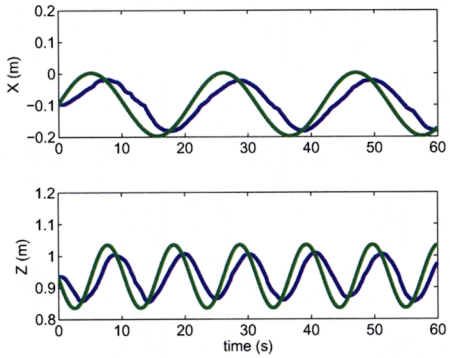

Fig. 7. Time series graph of right hand tracking of a figure 8 in the perpendicular plane. The top graph shows the left-right position, and the bottom is the vertical. Green is desired and blue is actual.

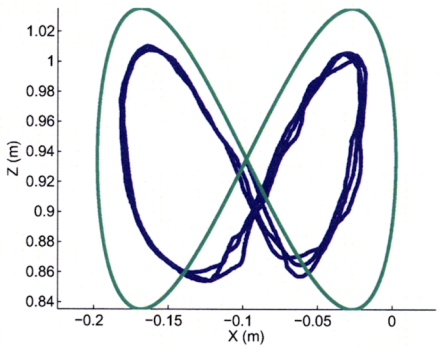

Fig. 8. Right hand tracking of a figure 8 in the perpendicular plane. Green is desired position and blue is actual. Note that performance is degraded because the hand controller operates in the null-space of the balance/constraint controller

V. Discussion

Ideally we would like to control the robot via floating base inverse dynamics [20]. This would allow us to achieve accurate tracking performance while maintaining compliance, since high gain position controllers would not be required. However, the accurate estimation of inertial parameters of a floating base inverse dynamics model is challenging, and work continues to be on going. The position based control explored in this paper has the advantage of model-freedom, and demonstrates some nice results on simple tasks. However, for critical tasks such as balancing, we would prefer more accurate control methods. For example, moderate to low task space gains contribute to the tracking lags observed in the COG tracking task (figure 6). Raising task space gains, may result in better tracking performance with a higher risk of instability. Limitations of the position control approach become obvious as we attempt more complicated or faster

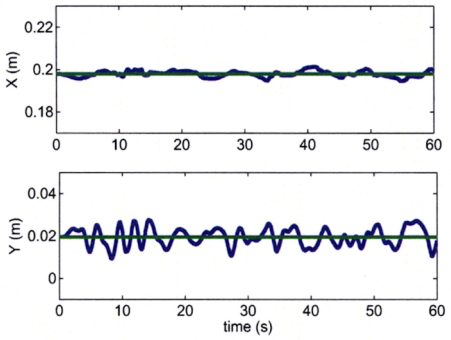

Fig. 9. COG tracking in the horizontal plane during the hand figure-8 task. The COG is commanded to remain in a fixed position.

motion, which may generate more inertia. Unfortunately, the model-free techniques cannot cope with the disturbances due to momentum, and the need for dynamic balance [13], [14] becomes apparent. The hand tracking experiment demonstrates the feasibility of whole body motions, while maintaining static balance. Again, faster motion would require model-based approaches. In addition, we would like to attempt whole body motion during constraint switching, for more complex tasks including locomotion.

VI. Conclusion

In this paper, we have discussed some relevant issues concerning inverse kinematics algorithms for floating base structures. In particular we have addressed how to deal with under-actuation by sufficiently constraining the robot. We also have taken the first steps of implementing full body, floating base, task space control on a real humanoid robot. While the performance of the demonstrated position based controllers is limiting, they show promise and demonstrate the feasibility of the approach. Work continues on model-based floating base inverse dynamics, and the development of robust controllers for more complex tasks including constraint switching and locomotion.

VII. Acknowledgments

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, the ATR Computational Neuroscience Laboratories, the Japan Society for the Promotion of Science's Summer Program, and the National Science Foundation's East Asia and Pacific Summer Institutes for U.S. Graduate Students.

References

[1] Y. Umetani and K. Yoshida, Resolved motion rate control of space manipulators with generalized Jacobian matrix, IEEE Transactions on Robotics Automation, 5(3):303-314, 1989.

[2] Z. Vafa and S. Dubowsky, The kinematics and dynamics of space manipulators: the virtual manipulator approach, Int. J. Robotics Res. 9(4):3-2, 1990.

[3] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. IEEE Transactions on Robotics and Automation, 9(5):531-543, 1993.

[4] F. Fabrizio ad B. Siciliano. Kinematic control of redundant free-floating systems. Advanced Robotics, Vol. 15, No. 4, pp. 429-448, 2001.

[5] L. Sentis and O. Khatib, "Control of Free-Floating Humanoid Robots Through Task Prioritization", Proceedings of the 2005 International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 1730-1735.

[6] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots,", Proceedings of the 2006 ICRA IEEE International Conference on Robotics and Automation, Orlando, USA, 2006, pp. 1963-1969.

[7] L. Sentis and O. Khatib,"A whole-body control framework for humanoids operating in human environments,", Proceedings of the 2006 ICRA IEEE International Conference on Robotics and Automation, Orlando, USA, 2006, pp. 2641-2648.

[8] O. Khatib, L. Sentis and J. Park, A Unified Framework for Whole-Body Humanoid Robot Control with Multiple Constraints and Contacts, *Springer Tracts in Advanced Robotics - STAR Series, European Robotics Symposium (EURON)*, Prague, Czech Republic, March 2008.

[9] O. Khatib, A unified approach to motion and force control of robot manipulators: The operational space formulation, Int. J. Robotics Res. 3(1):43-53, 1987.

[10] J. Nakanishi, R. Cory, M. Mistry, J. Peters, S. Schaal, Operational Space Control: A Theoretical and Empirical Comparison, Int. J. Robotics Res. Vol. 27, No. 6, 737-757, 2008.

[11] M. Mistry, P. Mohajerian, S. Schaal., "Arm movement experiments with joint space force fields using an exoskeleton robot" IEEE Ninth International Conference on Rehabilitation Robotics, Chicago, IL, pp.408-413, 2005.

[12] J. Ting, M. Mistry, J. Peters, S. Schaal, J. Nakanishi. "A Bayesian Approach to Nonlinear Parameter Identification for Rigid Body Dynamics", Robotics: Science and Systems, Philadelphia, PA, 2006.

[13] M. Vukobratovic and J. Stepanenko, "On the stability of anthropomorphic systems," Journal of Mathematical Bioscience, vol. 15, pp. 1-37, 1972.

[14] M. Vukobratovic and B. Borovac, "Zero-moment point – Thirty five years of its life," International Journal of Humanoid Robotics, vol. 1, pp. 157-173, 2004.

[15] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based redundancy control of robot manipulators, Int. J. Robotics Res., 6(2):3-15, 1987.

[16] B. Siciliano and J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems, Fifth International Conference on Advance Robotics, Pisa, Italy, pp. 1211-1216, 1991,

[17] S. Chiaverini. "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators", IEEE Transactions on Robotics and Automation, Vol 13, No. 3:398-410, 1997.

[18] P. Baerlocher, and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 323-329, 1998.

[19] M. Mistry, J. Nakanishi, S. Schaal, "Task Space Control with Prioritization for Balance and Locomotion", 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, pp. 331-338, 2007.

[20] J. Nakanishi, M. Mistry, and S. Schaal, "Inverse dynamics control with floating base and constraints", 2007 International Conference on Robotics and Automation, Rome, Italy, pp. 1942-1947, 2007.

[21] T. Sugihara, Mobility Enhancement Control of Humanoid Robot based on Reaction Force Manipulation via Whole Body Motion. PhD thesis, University of Tokyo, 2004.

[22] L. Sciavicco, B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, Inc., New York, 1996.

[23] G. Cheng, S. Hyon, J. Morimoto, A. Ude, J. Hale, G. Colvin, W. Scroggin, and S. C. Jacobsen, "CB: A humanoid research platform for exploring neuroscience", Advanced Robotics, Vol 21, No. 10, 1097-1114, 2007.

[24] G. Strang, *Linear Algebra and its Applications*, 3rd ed., Harcourt Brace Jovanovich College Publishers, Orlando, 1988.