Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Control of Free-Floating Humanoid Robots Through Task Prioritization

Luis Sentis and Oussama Khatib
Artificial Intelligence Laboratory
Stanford University
Stanford, CA 94305, USA
{*lsentis, khatib*}*@robotics.stanford.edu*

***Abstract*— The possibility of controlling humanoid robots in free-space opens new fields of application involving free-floating behaviors. Recently, we presented a prioritized task-oriented control framework for the control of multiple motion primitives while complying with physical constraints imposed by the robot's body and environment. We adapt here this framework to the control of free-floating robots.**

***Index Terms*— Free-floating dynamics, prioritized control.**

I. Introduction

An important characteristic of humanoid robots is their free-floating dynamics. For efficient control, we need to describe the interactions between the free-floating base and the robot's motion. We present here dynamic models describing these interactions and use them for efficient control of free-floating humanoids.

The control of multiple prioritized task primitives has been addressed since the early 1980s at the inverse kinematic level [4], [6], [14], [15], [21]. The dynamic interaction between the robot's operational space motions and forces was first addressed in the *Operational Space Formulation* [10], [11]. More recently, we extended this formulation to allow controlling multiple prioritized task primitives [13], [20].

The control of free-floating robots and/or robots with passive degrees of freedom was first addressed in the late 1980s. Arai and Tachi [1] proposed a cartesian space approach for the control of underactuated robots with the aim of reducing the number of actuators. For space applications such as the ETS-VII JAXA (Japanese Aerospace Exploration Agency) mission [16], Umetami and Yoshida introduced an extended end-effector Jacobian named *Generalized Jacobian* [7], [22] – a.k.a. *Free-Floating System's Jacobian* [3], [17] – which combines the dynamic interactions between the joint velocities and the velocities of the free-floating base.

In this paper we extend the theory and control strategies described in [13] and [20] to free-floating robots. First, we introduce an *Extended Generalized Jacobian* that integrates the free-floating dynamics into the task prioritization framework. Second, we analyze task feasibility and propose robust controllers that can operate under task singularity. We will complement the theory with several control examples implemented in a physics-based humanoid robot simulator.

II. Operational space Control in Free-Floating Robots

We consider the humanoid robot shown in fig. 1, where we have assigned a reference frame to the base link represented by the 6×1 vector x_{base}, containing three translational and three orientational components. The free-floating system formed by the robot joints and the free-floating degrees of freedom of the base link, can be characterized by the dynamic equation

$$A_{sys} \begin{pmatrix} \ddot{x}_{base} \\ \ddot{q} \end{pmatrix} + b_{sys} + g_{sys} = \begin{pmatrix} 0 \\ \Gamma \end{pmatrix}, \qquad (1)$$

where A_{sys}, b_{sys} and g_{sys} are the mass matrix, the Coriolis/centrifugal component, and the gravity component respectively, $\ddot{x}_{base}$ is the 6×1 vector of base accelerations, $\ddot{q}$ is the $n \times 1$ vector of joint accelerations (n is the number of joints of the robot), and Γ is the $n \times 1$ vector of joint torques .

Fig. 1. **Free-Floating Humanoid Robot:** A free-floating base frame is assigned to the hip link.

We derive the joint space dynamics, by defining a *joint selection matrix* $S_q = \begin{bmatrix} 0 & I \end{bmatrix}$ characterized by the equality

$$q = S_q \begin{pmatrix} x_{base} \\ q \end{pmatrix}, \quad (2)$$

where I is the $n \times n$ identity matrix. Let us consider the *dynamically consistent generalized inverse* of S_q [10] described by the equality

$$\overline{S}_q = A_{sys}^{-1} S_q^T A, \quad (3)$$

where $A = \left(S_q A_{sys}^{-1} S_q^T\right)^{-1}$ is the joint space mass/inertia. We use this operator to transform eq. (1) into the joint space dynamics:

$$\overline{S}_q^T \left(A_{sys} \begin{pmatrix} \ddot{x}_{base} \\ \ddot{q} \end{pmatrix} + b_{sys} + g_{sys} = \begin{pmatrix} 0 \\ \Gamma \end{pmatrix} \right) \Longrightarrow$$
$$\Longrightarrow A\ddot{q} + b + g = \Gamma. \quad (4)$$

Here $b = \overline{S}_q^T b_{sys}$ and $g = \overline{S}_q^T g_{sys}$ are the joint Coriolis/centrifugal and gravity torque components. Let us break down the free-floating system's dynamic quantities into block components:

$$A_{sys} = \begin{pmatrix} A_{bb} & A_{br} \\ A_{rb} & A_{rr} \end{pmatrix} \quad b_{sys} = \begin{pmatrix} b_b \\ b_r \end{pmatrix} \quad g_{sys} = \begin{pmatrix} g_b \\ g_r \end{pmatrix}, \quad (5)$$

where the subscripts b and r represent the *base* and *robot* components respectively. Further development of eq. (4) renders the equalities,

$$A = A_{rr} - A_{rb} A_{bb}^{-1} A_{br} \quad (6)$$
$$b = b_r - A_{rb} A_{bb}^{-1} b_b \quad (7)$$
$$g = g_r - A_{rb} A_{bb}^{-1} g_b, \quad (8)$$

where A is the *Schur Complement* [8] of A_{bb}, and is also referred as the *Generalized Inertial Tensor* [23].

We consider the following kinematic equation of an *operational space point* x:

$$\dot{x} = J_{sys} \begin{pmatrix} \dot{x}_{base} \\ \dot{q} \end{pmatrix} \quad (9)$$

$$J_{sys} = \begin{pmatrix} \partial x / \partial x_{base} & \partial x / \partial q \end{pmatrix} = \begin{pmatrix} J_b & J_r \end{pmatrix}. \quad (10)$$

where J_b and J_r are the components corresponding to the base and the robot joints respectively.

Due to the dynamic coupling between the robot's joint motion and the motion of its free-floating base, the operational space kinematics can be obtained from the joint velocities alone: $\dot{x} = J\,\dot{q}$. Here J is called the *Generalized Jacobian* [22] and combines the system kinematic properties with the inertial quantities due to free-floating dynamics. To derive J we need to consider the conservation of angular and linear momentum. Let us study the upper half of the free-floating system's Lagrangian [7]:

$$\frac{d\left(A_{bb}\dot{x}_{base} + A_{br}\dot{q}\right)}{dt} - \nabla_{x_{base}} K_{sys} = 0. \quad (11)$$

The equality $\nabla_{x_{base}} K_{sys} = 0$ holds, since A_{sys} and consequently the system's kinetic energy K_{sys} are independent of the base motion.

By integrating eq. (11) we reveal the conservation of the system's momentum:

$$A_{bb}\dot{x}_b + A_{br}\dot{q} = constant. \quad (12)$$

where the *constant* term represents the spatial momentum at the starting time. For the time being let us consider this term equal to zero. From eqs. (9), (10) and (12) we can decompose J into the equality

$$J = J_r - J_b A_{bb}^{-1} A_{br}, \quad (13)$$

which reveals the dependency of the Generalized Jacobian on kinematic and dynamic quantities. Alternatively, we can express J as

$$J = J_{sys}\overline{S}_q, \quad (14)$$

where $\overline{S}_q = \begin{pmatrix} -A_{bb}^{-1} A_{br} \\ I \end{pmatrix}$.

A. Task Control

We use the joint dynamics and the Generalized Jacobian described in eqs. (4) and (14) to obtain the operational space dynamics of the free-floating system through the following transformation,

$$\overline{J}^T \left(A\ddot{q} + b + g = \Gamma_{task} \right) \Longrightarrow \Lambda\ddot{x} + \mu + p = F, \quad (15)$$

where Γ_{task} is the torque input that controls the operational space task, $\overline{J} = A^{-1} J^T \Lambda$ is the dynamically consistent generalized inverse of J, $\Lambda = (JA^{-1}J^T)^{-1}$ is the operational space mass/inertia, μ and p are the operational space Coriolis/centrifugal and gravity terms, and $F = \overline{J}^T \Gamma_{task}$ is the vector of operational space forces. Dynamic decoupling is achieved by using the control torque,

$$\Gamma_{task} = J^T \left(\Lambda \ddot{x}_{ref} + \mu + p \right), \quad (16)$$

where $\ddot{x}_{ref}$ is the control reference at the acceleration level. When we apply this control, we obtain the decoupled behavior $\ddot{x} = \ddot{x}_{ref}$.

Because humanoid robots are redundant with respect to the operational space task, we associate a dynamically consistent task null space [10] defined by

$$N = I - \overline{J}J. \quad (17)$$

B. Posture Control

The control of the null space motion (a.k.a. self-motion) is called *posture control*. We can control task and posture simultaneously by considering the compound torque

$$\Gamma = \Gamma_{task} + \Gamma_{posture}. \quad (18)$$

Because the posture is dynamically consistent with the task, it can be further expressed as a linear combination of the columns of N^T:

$$\Gamma = \Gamma_{task} + N^T \Gamma_{subtask}, \quad (19)$$

where $\Gamma_{subtask}$ represents the control input of a secondary task (a.k.a. posture task). Let us consider controlling a posture task defined by the vector x_p, and with Jacobian $J_{sys(p)} = (\partial x_p/\partial x_{base}\ ,\ \partial x_p/\partial q)$. Similarly to eq. (14), a Generalized Jacobian can be associated with this task:

$$J_p = J_{sys(p)}\overline{S}_q. \tag{20}$$

In a previous paper, [13] we introduced an *extended posture Jacobian*, $J_{p|t}$, that was dynamically consistent with the task. Using eq. (20) we extend this Jacobian to free-floating robots:

$$J_{p|t} = J_{sys(p)}\overline{S}_q N, \tag{21}$$

$$\Lambda_{p|t} = \left(J_{p|t}A^{-1}J_{p|t}^T\right)^{-1}. \tag{22}$$

Here $\Lambda_{p|t}$ is the mass matrix associated with this extended posture space. The extended posture dynamics can be obtained from the projection,

$$\overline{J}_{p|t}^T\left(A\ddot{q} + b + g = \Gamma_{posture}\right) \Longrightarrow \Lambda_{p|t}\ddot{x}_{p|t} + \mu_{p|t} + p_{p|t} = F_{p|t}, \tag{23}$$

where $\overline{J}_{p|t} = A^{-1}J_{p|t}^T\Lambda_{p|t}$ is the dynamically consistent generalized inverse of $J_{p|t}$, $F_{p|t}$ is the vector of control forces, and $\mu_{p|t}$ and $p_{p|t}$ are the Coriolis/centrifugal, and gravity force vectors of the extended posture space. To accomplish dynamically-consistent control of the posture subtask we choose the control input

$$\Gamma_{posture} = J_{p|t}^T F_{p|t}, \tag{24}$$

$$F_{p|t} = \Lambda_{p|t}\left(\ddot{x}_{ref(p)} - \ddot{x}_{p|bias}\right) + \mu_{p|t} + p_{p|t}, \tag{25}$$

where $\ddot{x}_{ref(p)}$ is the control reference. Here, $\ddot{x}_{p|bias}$ is a bias acceleration induced by the coupling of the primary task into the posture subtask. If $J_{p|t}$ is full rank, the subtask is feasible and this controller will yield the decoupled behavior $\ddot{x}_p = \ddot{x}_{ref(p)}$.

III. Prioritized Control in Free-Floating Robots

In this section we adapt the framework for the control of multiple prioritized task primitives [20] to free-floating robots. Prioritization was designed to ensure that constraints and critical task objectives are first accomplished, while optimizing the execution of the robot's global task. In general, a humanoid robot must control simultaneously a collection of N task objectives (including constraints), $\{x_k(q)\,|\,k = 1, 2, \ldots, N\}$, where the numbering represents the priority level in the control hierarchy. The following torque equation represents a prioritized control hierarchy:

$$\Gamma = \Gamma_{task(1)} + N_{task(1)}^T\Bigg(\Gamma_{task(2)} + N_{task(2)}^T \cdot \Big(\Gamma_{task(3)} + \cdots\Big)\Bigg). \tag{26}$$

Here, $\Gamma_{task(k)}$ is the control input of the k^{th} t task objective, and $N_{task(k)}$ is its null-space. This nested topology can be simplified by defining an extended null-space matrix containing the null-spaces of all previous task objectives:

$$N_{prev(k)} = N_{task(k-1)} \cdot N_{task(k-2)} \cdots N_{task(1)}, \tag{27}$$

where $prev(k) = \{1, \cdots, k-1\}$ represents the set of *previous* objectives. Equation (26) becomes

$$\Gamma = \Gamma_1 + \Gamma_{2|prev(2)} + \Gamma_{3|prev(3)} + \cdots, \tag{28}$$

where $\Gamma_{k|prev(k)} = N_{prev(k)}^T\Gamma_{task(k)}$, and the subscript $k|prev(k)$ indicates that the k^{th} task objective is controlled in the null-space of all previous ones.

A Generalized Jacobian can be associated to every task objective according to

$$J_k = J_{sys(k)}\overline{S}_q, \tag{29}$$

where $J_{sys(k)} = (\partial x_k/\partial x_{base}\ ,\ \partial x_k/\partial q)$. We define an Extended Generalized Jacobian resulting from the projection of the joint velocities into the null-space $N_{prev(k)}$:

$$J_{k|prev(k)} = J_k N_{prev(k)} = J_{sys(k)}\overline{S}_q N_{prev(k)}, \tag{30}$$

and we associate an *extended generalized inertia matrix*:

$$\Lambda_{k|prev(k)} = \left(J_{k|prev(k)}A^{-1}J_{k|prev(k)}^T\right)^{-1}. \tag{31}$$

where the subscript $k|prev(k)$ indicates the k^{th} task objective is operated within the null-space $N_{prev(k)}$. To establish prioritization at the acceleration level, $\Gamma_{k|prev(k)}$ should render null accelerations in all preceding levels $prev(k)$, or equivalently

$$\forall\, i \in prev(k) \quad J_i A^{-1} N_{prev(k)}^T = 0, \tag{32}$$

where J_i represents the Generalized Jacobian defined in eq. (29). In [20] we found that the null space that fulfills this constraint has the following unique solution,

$$N_{prev(k)} = I - \sum_{i=1}^{k-1}\overline{J}_{i|P(i)}J_{i|P(i)}, \tag{33}$$

where $\overline{J}_{i|P(i)} = A^{-1}J_{i|P(i)}^T\Lambda_{i|P(i)}$ is the dynamically consistent generalized inverse of $J_{i|P(i)}$. We obtain the dynamic behavior of the k^{th} task objective from the projection

$$\overline{J}_{k|prev(k)}^T\left(A\ddot{q} + b + g = \Gamma_{k|prev(k)}\right) \Longrightarrow \tag{34}$$

$$\Lambda_{k|prev(k)}\ddot{x}_{k|prev(k)} + \mu_{k|prev(k)} + p_{k|prev(k)} = F_{k|prev(k)},$$

where $\mu_{k|prev(k)}$ and $p_{k|prev(k)}$ are the prioritized Coriolis/centrifugal and gravity forces, and

$$F_{k|prev(k)} = \overline{J}_{k|prev(k)}^T\Gamma_{k|prev(k)} \tag{35}$$

are the virtual forces associated with the k^th task objective. We accomplish efficient control of these objectives by choosing the control input

$$\Gamma_{k|prev(k))} = J_{k|prev(k)}^T\Bigg(\Lambda_{k|prev(k)}\left(\ddot{x}_{ref(k)} - \ddot{x}_{k|bias}\right) + \mu_{k|prev(k)} + p_{k|prev(k)}\Bigg). \tag{36}$$

Here $\ddot{x}_{k|bias}$ is a bias acceleration induced by the coupling of higher priority task objectives (refer to [20] for further details). If $J_{k|prev(k)}$ is full rank, this controller will yield the decoupled behavior $\ddot{x}_k = \ddot{x}_{ref(k)}$, where $\ddot{x}_{ref(k)}$ is a control reference.

A. *Task Feasibility*

A task objective k is unfeasible if the Jacobian $J_{k|prev(k)}$ drops rank. In this circumstances, the extended generalized inertia matrix has the following eigen-decomposition

$$\Lambda^{-1}_{k|prev(k)} = J_{k|prev(k)} A^{-1} J^T_{k|prev(k)} = \begin{pmatrix} U_{r(k)} & U_{n(k)} \end{pmatrix} \begin{pmatrix} \Sigma_{r(k)} & \\ & 0 \end{pmatrix} \begin{pmatrix} U^T_{r(k)} \\ U^T_{n(k)} \end{pmatrix}, \quad (37)$$

where $\Sigma_{r(k)}$ is a diagonal matrix of non-zero eigenvalues, and $U_{r(k)}$ and $U_{n(k)}$ are matrices corresponding to non-zero and zero eigenvectors, respectively. Because some eigenvalues are equal to zero, it is not possible to fully control $\ddot{x}_k$. However, by choosing the control input

$$\Gamma_{k|prev(k)} = J^T_{k|prev(k)} \Bigg(\Big(U_{r(k)} \, \Sigma^{-1}_{r(k)} \, U^T_{r(k)} \Big) \cdot \big(\ddot{x}_{ref(k)} - \ddot{x}_{k|bias} \big) + \mu_{k|prev(k)} + p_{k|prev(k)} \Bigg). \quad (38)$$

we accomplish dynamic decoupling in the controllable directions according to $U^T_{r(k)} \big(\ddot{x}_k = \ddot{x}_{ref(k)} \big)$, where $U_{r(k)}$ defines these directions.

IV. Simulation Environment

To verify the proposed controller, we have developed a realtime humanoid robotic simulation environment, SAI [12]. SAI is a unique virtual environment that integrates multi-body dynamics [2], robot control, multi-contact simulation, and haptic interaction. It incorporates a dynamics engine that resolves forward and inverse dynamics of an n degrees-of-freedom (DOF) branching multi-body system with linear complexity, $O(n)$. Moreover, we can resolve p collisions with a complexity of $O(np + p^3)$ using operational space models [18].

V. Example: Interactive Hand Position Control under Joint Limit Constraints

We first study an example of interactive hand position control of a free-floating humanoid robot while complying with joint limit constraints. The humanoid model has $n = 24$ joint DOFs (degrees of freedom): 2×6 for the legs, 2×4 for the arms, 2 for the torso, and 2 for the head. The robot's height is $1.65m$; its weight is $71kg$. In this example, the robot is commanded to reach a target position with its left hand. To achieve this goal, the robot must accomplish simultaneously two task objectives: comply with joint limit constraints and control the left hand position. Therefore the following controller is applied:

$$\Gamma = \Gamma_{JLC}(1) + \Gamma_{HAND}(2) + \Gamma_{FRIC}(3), \quad (39)$$

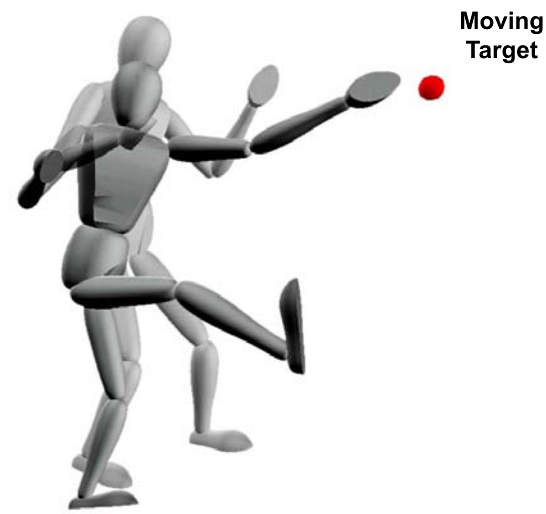
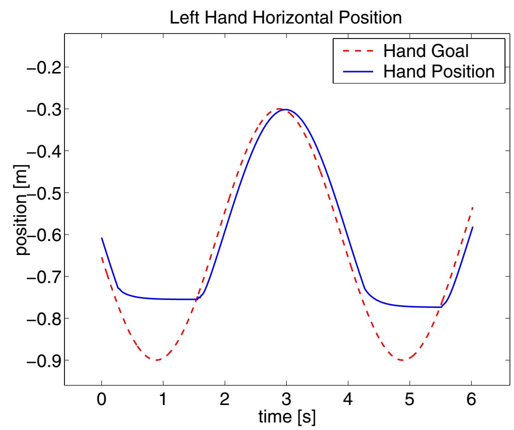
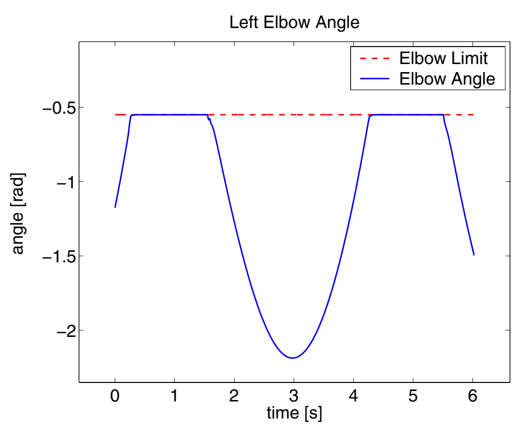


Fig. 2. **Hand Position Control under Joint Limit Constraints:** In this sequence, the hand is commanded to track a target. When the arm is fully stretched reaching the elbow limit, the task becomes unfeasible and the target cannot be reached.



(a)



(b)

Fig. 3. **Data Recorded During the Experiment Shown in Fig. 2:** The left hand tracks precisely the trajectory (a), except when the left elbow limit is reached (b).

where the subscripts JLC, $HAND$ and $FRIC$ stand for joint limit constraints, hand position control, and joint friction respectively, and the numbers in parenthesis represent their priority order. Because this hierarchy is imposed, if the hand is commanded to move beyond the robot's reach, the extended generalized mass matrix will display singular behavior according to eq. (37). The joint friction component is applied to damp the remaining self-motion.

The task objectives for this example are based on the following *Potential Fields* [9]:

$$V_{JLC} = \| q_{viol} - q_{limit} \|^2 \quad (40)$$

$$V_{HAND} = \| x_{hand} - x_{target} \|^2, \quad (41)$$

where $q_{violating}$ is the vector of joints that, at a given time, violate joint limits, q_{limit} is the vector of joint limit values, and x_{target} is an interactive hand position target. Because joint limits are defined in joint space, we associate a matrix $S_{violating}$ that selects the violating joints according to

$$q_{violating} = S_{violating} \begin{pmatrix} x_{base} \\ q \end{pmatrix}. \quad (42)$$

Furthermore, the Jacobian associated with the position of the left hand has the following decomposition:

$$J_{hand} = \begin{pmatrix} \partial x_{hand}/\partial x_{base} & \partial x_{hand}/\partial q \end{pmatrix}. \quad (43)$$

In this experiment, for every task objective k, we use a simple PD control reference with velocity saturation:

$$\ddot{x}_{ref(k)} = -k_v \left(\dot{x}_k - \nu \dot{x}_{des(k)} \right), \quad (44)$$

$$\dot{x}_{des(k)} = \frac{k_p}{k_v} \nabla V_k \qquad \nu = min\left(1, \frac{v_{max(k)}}{||\dot{x}_{des(k)}||}\right), \quad (45)$$

where $\dot{x}_{des(k)}$ is a desired velocity and $v_{max(k)}$ is a saturation value.

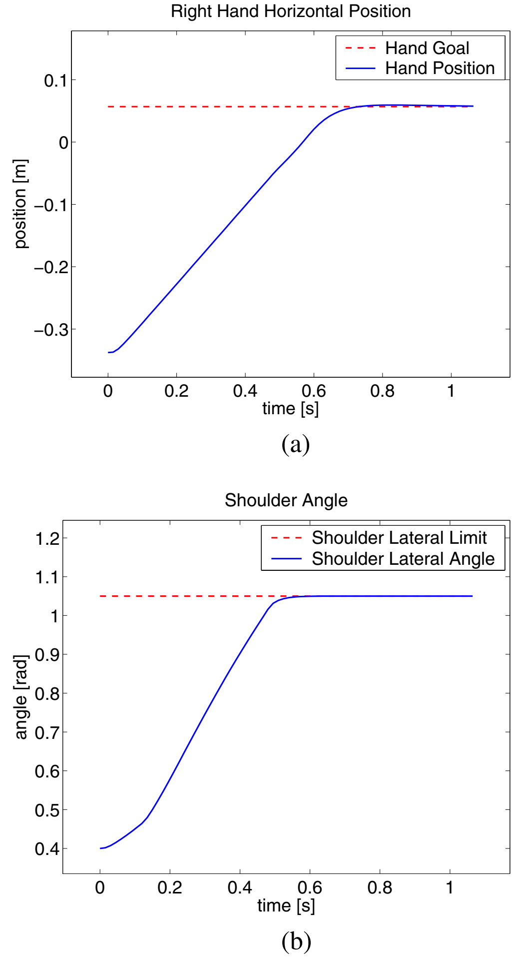


Fig. 5. **Data Recorded During the Experiment Shown in Fig. 4:** The right hand reaches effectively the goal position (a) despite reaching the right shoulder joint limit moments before (b).

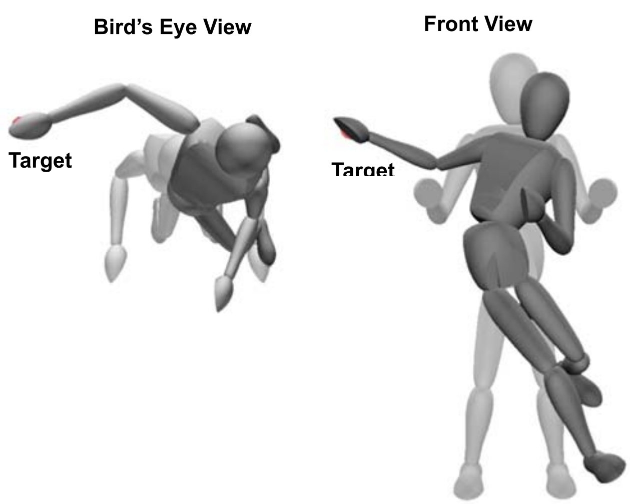


Fig. 4. **Another Example of Hand Position Control under Joint Limit Constraint:** This time, the shoulder lateral joint limit is reached in the middle of the motion. In contrast with the previous example, the task is feasible at all times and therefore it is accomplished effectively. Notice that the legs twist to the right side to compensate for the torso's side motion.

When we apply the controller defined in eq. (39) we obtain the results shown in fig. 3, where the robot's left hand is commanded to track a sinusoidal trajectory with period $T = 4s$ and amplitude $A = 0.6m$. The hand tracks effectively the goal trajectory except when the elbow joint limit is encountered during the intervals $\Delta t_1 = (0.2s, 1.55s)$, and $\Delta t_2 = (4.2s, 5.55s)$. During these intervals, the controller of eq. (38) is applied instead, thus stopping the hand $0.13m$ away from the final target position. Notice that the right leg stretches out to preserve angular momentum and the robot's body moves downward to preserve the center of mass position. Both of these behaviors are due to free-floating dynamics.

Let us study a similar example (see fig. 4) where the robot's right hand is commanded to move to a different target position. This time however, despite reaching the shoulder joint limit, the target is reached effectively because is feasible at all times. In fig. 5, we show the target being reached at $t = 0.7s$, while the right shoulder joint limit is reached at $t = 0.5s$.

VI. Conclusion

In an effort to develop controllers for new applications of humanoid robots, we have adapted the prioritized task-oriented control framework described in [20] to free-floating robots. This step allows us for the first time to generate on the spot *free-floating behaviors* such as running, jumping, or touching objects in free space.

This controller characterizes for the first time the virtual mass properties of prioritized tasks in free-floating robots, allowing us to control precisely task motions or to implement hybrid position/force controllers for free-floating behaviors.

We envision humanoid robots performing free-floating whole-body behaviors while maintaining high precision control of the task and body posture. The reported controller is suited for these applications, because it decomposes these motions into basic task primitives, and incorporates the free-floating dynamics into the control of each task objective. At the same time it allows to study task feasibility at runtime allowing to modify the control strategy if needed.

Acknowledgments

We are grateful for the contributions of Tine Lefebvre, Jae-Heung Park, James Warren, Vincent De Sapio, François Conti, Irena Pashchenko, Kyong-Sok Chang, Diego Ruspini, and Oliver Brock. Our work was supported by the Honda Humanoid Project [5], [19].

References

[1] H. Arai and S. Tachi. Dynamic control of a manipulator with passive joints in an operational coordinate space. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, USA, April 1991.

[2] K.C. Chang and O. Khatib. Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2000.

[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), October 1993.

[4] H. Hanafusa, T. Yoshikawa, and Y. Nakamura. Analysis and control of articulated robot with redundancy. In *Proceedings of IFAC Symposium on Robot Control*, volume 4, pages 1927–1932, 1981.

[5] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1321–1326, Leuven, Belgium, 1998.

[6] J. M. Hollerbach and K. C. Suh. Redundancy resolution of manipulators through torque optimization. *International Journal of Robotics and Automation*, 3(4):308–316, 1987.

[7] A. Jain and G. Rodriguez. An analysis of the kinematics and dynamics of uner-actuated manipulators. *IEEE Transactions on Robotics and Automation*, 9(4), August 1993.

[8] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[9] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–8, 1986.

[10] O. Khatib. A united approach to motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics Research*, 3(1):43–53, 1987.

[11] O. Khatib. Object manipulation in a multi-effector robot system. In R. Bolles and B. Roth, editors, *Robotics Research 4*, pages 137–144. MIT Press, 1988.

[12] O. Khatib, O. Brock, K.C. Chang, F. Conti, D. Ruspini, and L. Sentis. Robotics and interactive simulation. *Communications of the ACM*, 45(3):46–51, March 2002.

[13] O. Khatib, L. Sentis, J.H. Park, and J. Warren. Whole body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(1):29–43, March 2004.

[14] A.A. Maciejewski and C.A. Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *International Journal of Robotics Research*, 4(3):109–117, 1985.

[15] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based control of robot manipulators. *International Journal of Robotics Research*, 6(2):3–15, 1987.

[16] M. Oda, K. Kibe, and F. Yamagata. ETS-VII space robot in-orbit experiment satellite. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, USA, April 1996.

[17] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for free-floating space manipulators. *IEEE Transactions on Robotics and Automation*, 7(6), December 1991.

[18] D. Ruspini and O. Khatib. Collision/Contact Models for Dynamic Simulation and Haptic Interaction. In *The 9th International Symposium of Robotics Research (ISRR'99)*, pages 185–195, Snowbird, USA, October 1999.

[19] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: System overview and integration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2478–2483, Laussane, Switzerland, October 2002.

[20] L. Sentis and O. Khatib. A prioritized multi-objective dynamic controller for robots in human environments. In *Proceeding or the IEEE/RSJ International Conference on Humanoid Robots*, Los Angeles, USA, November 2004.

[21] B. Siciliano and J. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Proceedings of the IEEE International Conference on Advanced Robotics*, pages 1211–1216, Pisa, Italy, June 1991.

[22] Y. Umetami and K. Yoshida. Resolved motion rate control of space manipulators with generalized Jacobian matrix. *IEEE Transactions on Robotics and Automation*, 5(3), June 1989.

[23] Y. Xu and T. Kanade. *"Space Robotics: Dynamics and Control"*. Prentice Hall, 1992.