# TASK-ORIENTED CONTROL OF HUMANOID ROBOTS THROUGH PRIORITIZATION

LUIS SENTIS AND OUSSAMA KHATIB
*Artificial Intelligence Laboratory*
*Stanford University*
*Stanford, CA 94305, USA*
{*lsentis, khatib*}*@robotics.stanford.edu*

In humanoids, a whole-body task involves the control of multiple subtasks while complying with physical constraints imposed by the robot's structure and the external environment. To interact in realtime with the robot, we must guarantee that these constraints are fulfilled and that essential subtasks are first accomplished. In this paper, we present a hierarchical framework that focuses on two aspects of the control: (1) the definition of an extended task space that complies with the constraints and with higher priority tasks, and (2) the dynamic characterization and control of tasks within this space. We will analyze several control examples demonstrating the effectiveness of this framework.

*Keywords*: Prioritization, task-oriented control, Operational Space Formulation.

## 1. Introduction

A new generation of humanoids that operate interactively [11] is emerging. The control of these robots is challenging because we must control simultaneously multiple criteria. A prioritized hierarchy is needed to accomplish a global task without violating structural and environmental constraints, in addition to fulfilling critical subtasks.

Task-oriented control was first studied at the inverse kinematic level [2,4,13,14,18]. In 1987, the *Operational Space Formulation* [7,8] was introduced to address the dynamic interaction between the robot's end-effector motion and force. In the Operational Space Formulation, an operational task is controlled and a dynamically consistent null-space describes the additional self-motion. Multiple operational tasks can be controlled if they are concatenated into a single task vector. Although this non-prioritized structure is straightforward, if a control conflict occurs, a tracking error will affect all the conflicting tasks.

We solve this problem by controlling secondary tasks (a.k.a. postures) using the robot's self-motion. The *Extended Operation Space Formulation* [15] addresses dynamic control of the posture in joint-space. We presented previously [12] a broader extension to the Operational Space Formulation that characterizes and controls general subtasks in the posture space.

As robot movement becomes more elaborate, we need to accomplish concurrently

a growing number of subtasks. To ensure that constraints and critical subtasks are fulfilled, we must establish a prioritized hierarchy between these two and other less critical subtasks. In addition, we must identify the task space within this hierarchy, as well as a methodology to analyze the feasibility of the tasks. In this paper, we describe precisely such a controller and characterize the dynamic behavior of the subtasks within their priority level.

As an illustration of the potential and feasibility of this controller, we present several examples of interactive whole-body control: (1) an example of interactive tracking of the position of the robot's hand under joint limit constraints, and (2) two examples of compliant tasks.

## 2. Task and Posture Decomposition

We begin by describing the robot's dynamics in terms of the joint coordinates $q$,

$$A(q)\ddot{q} + b(q, \dot{q}) + g(q) = \Gamma. \tag{1}$$

For this system of $n$ equations, $\Gamma$ is the set of joint torques, $A(q)$ is the joint inertia matrix, $b(q, \dot{q})$ is the Coriolis and centrifugal torque vector, and $g(q)$ is the gravity torque vector.

The Operational Space Formulation provides a decomposition that describes the dynamics of a primary task and a posture that operates in the task-consistent null-space according to the torque equation

$$\Gamma = \Gamma_{task} + \Gamma_{posture}. \tag{2}$$

For an $m$ dimensional operational task $x_t(q)$ with Jacobian $J_t(q) = \partial x_t(q)/\partial q$, the projection of the joint dynamics into the task space will render the $m \times m$ dynamic equation given by

$$\Lambda_t \ddot{x}_t + \mu_t + p_t = F_t, \tag{3}$$

where $F_t$ is a force control vector in the task space, $\Lambda_t = \left(J_t A^{-1} J_t^T\right)^{-1}$ is the task inertia matrix, $\mu_t$ is the Coriolis and centrifugal force vector, and $p_t$ is the gravity force vector. The control input

$$\Gamma_{task} = J_t^T F_t, \tag{4}$$

$$F_t = \Lambda_t \ddot{x}_{ref(t)} + \mu_t + p_t, \tag{5}$$

provides the decoupled behavior $\ddot{x}_t = \ddot{x}_{ref(t)}$, where $\ddot{x}_{ref(t)}$ is a reference input at the acceleration level. The task-consistent null-space [6] is defined by

$$N_t(q) = I - \overline{J}_t J_t, \tag{6}$$

where $\overline{J}_t = A^{-1} J_t^T \Lambda_t$ is the dynamically-consistent generalized inverse of $J_t$.

A posture is defined as a linear combination of the columns of $N_t^T$. Therefore the task and posture decomposition is given by the equation

$$\Gamma = \Gamma_{task} + N_t^T \Gamma_{subtask}. \tag{7}$$

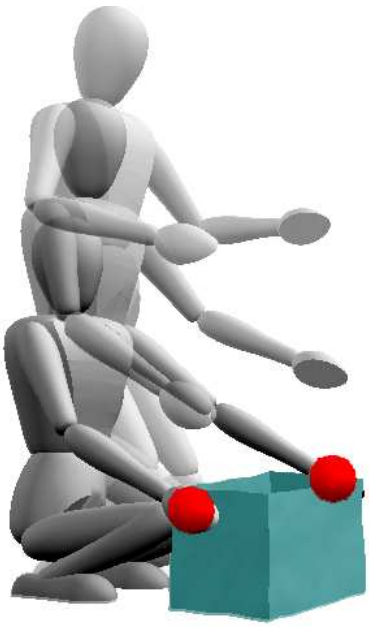

Fig. 1. **Task and Posture Decomposition:** In this sequence, the robot's task is to reach the sides of the box while maintaining body self-balance and upright posture. The spheres represent the target positions of the robot's hands.

The control input, $\Gamma_{subtask}$, is used to control a secondary task $x_p(q)$ with Jacobian $J_p = \partial x_p / \partial q$. In a previous paper, [12] we introduced an extended kinematic space defined by the Jacobian $J_{p|t} = J_p N_t$, resulting from projecting the joint rates into the task consistent null-space. An $l$ dimensional posture subtask is characterized by the $l \times l$ inertia matrix $\Lambda_{p|t} = \left(J_{p|t} A^{-1} J_{p|t}^T\right)^{-1}$, and its dynamic behavior is described by the equation

$$\Lambda_{p|t} \ddot{x}_{p|t} + \mu_{p|t} + p_{p|t} = F_{p|t}, \tag{8}$$

where $F_{p|t}$ is the control force vector, and $\mu_{p|t}$ and $p_{p|t}$ are, respectively, the Coriolis/centrifugal, and gravity force vectors of the posture. To accomplish dynamically-consistent control of posture tasks [9] we choose the control input

$$\Gamma_{posture} = J_{p|t}^T F_{p|t}, \tag{9}$$

$$F_{p|t} = \Lambda_{p|t}\left(\ddot{x}_{ref(p)} - \ddot{x}_{p|bias}\right) + \mu_{p|t} + p_{p|t}, \tag{10}$$

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 task.

If $J_{p|t}$ is full rank, the posture task is feasible and this controller will yield the decoupled behavior $\ddot{x}_p = \ddot{x}_{ref(p)}$. Figure 1 illustrates the control of an upright posture with simultaneous control of the position of the hands and the robot's self-balance.

A posture is unfeasible when the Jacobian $J_{p|t}$ drops rank. The inertial properties of the posture approach infinity in the uncontrollable directions. Under these circumstances, the inertia matrix has the following eigen-decomposition

$$\Lambda_{p|t}^{-1} = J_{p|t}A^{-1}J_{p|t}^T = \begin{bmatrix} U_r & U_n \end{bmatrix} \begin{bmatrix} \Sigma_r & \\ & 0_{(l-k)\times(l-k)} \end{bmatrix} \begin{bmatrix} U_r^T \\ U_n^T \end{bmatrix}, \tag{11}$$

where $k$ represents the number of uncontrollable directions, $\Sigma_r$ is a $k \times k$ diagonal matrix of non-zero eigenvalues ($r$ stands for rank), $U_r$ is an $l \times k$ matrix with columns corresponding to the non-zero eigenvectors, and $U_n$ is an $l \times (l-k)$ matrix corresponding to the zero eigenvectors ($n$ stands for null). Because some eigenvalues are equal to zero, it is not possible to fully decouple $\ddot{x}_p$, however, if the control input is chosen to be

$$F_{p|t} = \left(U_r\Sigma_r^{-1}U_r^T\right)\left(\ddot{x}_{ref(p)} - \ddot{x}_{p|bias}\right) + \mu_{p|t} + p_{p|t}, \tag{12}$$

we will accomplish dynamic decoupling in the controllable directions $U_r$ according to $U_r^T\left(\ddot{x}_p = \ddot{x}_{ref(p)}\right)$.

To control whole-body movements effectively, we must comply with multiple criteria. In the following section we present an extension that allows us to introduce a multi-level prioritized hierarchy.

## 3. Prioritized Tasks

In this section, we propose a multi-level control hierarchy that extends the task and posture decomposition previously described. We impose this hierarchy to ensure that constraints and critical tasks are first accomplished, while optimizing the execution of the global task.

To prevent a task from violating a constraint, we express the constraint as a high priority subtask, and we control the task within the null-space of the constraint $N_c$ according to the torque decomposition

$$\Gamma = \Gamma_{constraint} + N_c^T\Gamma_{task}. \tag{13}$$

In general, the robot must control simultaneously a collection of $N$ subtasks (including constraints), $\{x_k(q) \,|\, k = 1, 2, \ldots, N\}$ with Jacobians $J_k(q) = \partial x_k(q)/\partial q$, where the numbering represents the ordering in the control hierarchy. The following torque equation represents the multi-level control hierarchy:

$$\Gamma = \Gamma_{task(1)} + N_{task(1)}^T\left(\Gamma_{task(2)} + N_{task(2)}^T\left(\Gamma_{task(3)} + \cdots\right)\right). \tag{14}$$

Here, $\Gamma_{task(k)}$ is the control input of a subtask $k$, 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 tasks:

$$N_{prev(k)} = N_{task(k-1)} \cdot N_{task(k-2)} \cdots N_{task(1)}, \tag{15}$$

where $prev(k) = \{1, \cdots, k-1\}$ represents the set of *previous* tasks. Equation (14) becomes

$$\Gamma = \Gamma_1 + \Gamma_{2|prev(2} + \Gamma_{3|prev(3)} + \cdots, \tag{16}$$

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

Similarly to the task and posture decomposition discussed in Section 2, we define an extended task-consistent Jacobian resulting from the projection of the joint velocities into the null-space $N_{prev(k)}$:

$$J_{k|prev(k)} \triangleq J_k N_{prev(k)}, \tag{17}$$

and we associate an extended inertia matrix:

$$\Lambda_{k|prev(k)} = \left(J_{k|prev(k)} A^{-1} J^T_{k|prev(k)}\right)^{-1}. \tag{18}$$

The dynamic behavior of a subtask is obtained by projecting the robot's joint dynamics into the subtask space:

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

$$\Lambda_{k|prev(k)}\ddot{x}_{k|prev(k)} + \mu_{k|prev(k)} + p_{k|prev(k)} = F_{k|prev(k)}, \tag{19}$$

where $\overline{J}_{k|prev(k)}$ is the dynamically-consistent generalized inverse of the extended Jacobian, and $\mu_{k|prev(k)}$, $p_{k|prev(k)}$, and $F_{k|prev(k)}$ are the Coriolis/centrifugal, gravity, and force vectors of the subtask, respectively.

### 3.1. *Recursive Null-space*

The extended null-space of a subtask $k$, is the space of motion with no acceleration effects on any of the higher priority tasks, or equivalently

$$\forall\, i \in prev(k) \quad J_i A^{-1} N^T_{prev(k)} = 0. \tag{20}$$

This constraint leads to the following unique solution

$$N_{prev(k)} = I - \sum_{i=1}^{k-1} \overline{J}_{i|prev(i)} J_{i|prev(i)}, \tag{21}$$

where $\overline{J}_{i|prev(i)} = A^{-1} J^T_{i|prev(i)} \Lambda_{i|prev(i)}$.

*Proof by Induction of Equation (21):*

(1) For $k = 2$, $J_1 A^{-1} N^T_{prev(2)} = J_1 A^{-1} - J_1 A^{-1} J_1^T \Lambda_1 J_1 A^{-1} = 0$.
(2) For any $k$ and $\forall i \in prev(k)$ let us assume $J_i A^{-1} N^T_{prev(k)} = 0$.
(3) For $k+1$, $\forall i \in prev(k+1)$, and using (2),

$$J_i A^{-1} N^T_{prev(k+1)} = J_i A^{-1} N^T_{prev(k)} \left(I - J^T_{k|prev(k)} \overline{J}^T_{k|prev(k)}\right) = 0. \qquad \square$$

Here, we have used the properties: $\forall k \ \left(N_{prev(k)}\right)^n = N_{prev(k)}$, and $J_k A^{-1} N^T_{prev(k)} = 0$, where $n$ represents any integer.

### 3.2. *Control of Prioritized Tasks*

We accomplish efficient control of a task $x_k(q)$ by choosing the control torque

$$\Gamma_{k|prev(k)} = J^T_{k|prev(k)} F_{k|prev(k)}, \tag{22}$$

$$F_{k|prev(k)} = \Lambda_{k|prev(k)} \big( \ddot{x}_{ref(k)} - \ddot{x}_{k|bias} \big) + \mu_{k|prev(k)} + p_{k|prev(k)}. \tag{23}$$

Here $\ddot{x}_{k|bias}$ is a bias acceleration induced by the coupling of higher priority tasks. 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.

### 3.3. *Task Feasibility*

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

$$\Lambda^{-1}_{k|prev(k)} = J_{k|prev(k)} A^{-1} J^T_{k|prev(k)} = \begin{bmatrix} U_{r(k)} \; U_{n(k)} \end{bmatrix} \begin{bmatrix} \Sigma_{r(k)} & \\ & 0 \end{bmatrix} \begin{bmatrix} U^T_{r(k)} \\ U^T_{n(k)} \end{bmatrix}, \tag{24}$$

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 zero, it is not possible to control $\ddot{x}_k$ fully. However, by choosing the control input

$$F_{k|prev(k)} = \big( U_{r(k)} \Sigma^{-1}_{r(k)} U^T_{r(k)} \big) \big( \ddot{x}_{ref(k)} - \ddot{x}_{k|bias} \big) + \mu_{k|prev(k)} + p_{k|prev(k)}, \tag{25}$$

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.

## 4. Simulation Environment

To verify the proposed controller, we have developed a realtime humanoid robotic simulation environment, SAI [10]. SAI is a unique virtual environment that integrates multi-body dynamics [1], 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 [16]. Figure 2 displays a sequence of snapshots from simulated falling of our humanoid.

## 5. Example: Interactive Hand Position Control under Joint Limit Constraints

To study the proposed controller, we first explore an example of interactive hand position control with joint limit constraints. Our humanoid robot model consists of $n = 24$ DOF: $2 \times 6$ for the legs, $2 \times 4$ for the arms, 2 for the torso, and 2 for head. The robot's height is 1.65 $m$; its weight is 71 $Kg$. In this example, the robot

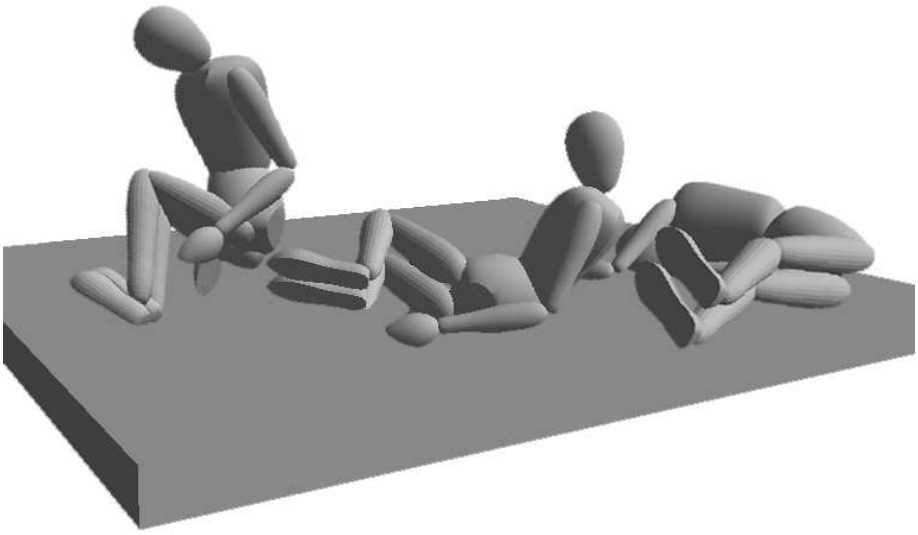

Fig. 2. **Robot Simulation:** This SAI simulation shows a robot falling due to gravity. Efficient multi-body dynamics and multi-contact algorithms resolve the simulation in real-time.

is commanded to reach a target position with its left hand. To achieve this goal, the robot must take into account four subtasks: joint limit constraints, self-balance, hand position control, and whole-body posture. Self-balance is based on the control of the global center of mass:

$$x_{com} = \frac{1}{M}\sum_{i=1}^{n} m_i \cdot x_{com(i)} \qquad J_{com} = \frac{1}{M}\sum_{i=1}^{n} m_i \cdot J_{com(i)}, \tag{26}$$

where $x_{com(i)}$ represents the center of mass of link $i$ and $M$ is the robot's total mass. The control of the hand is based on the latter's Cartesian position $x_{hand}$. The posture maximizes joint manipulability. To accomplish these subtasks we first define the following *Potential Fields* [5]:

$$V_{JLC} = \| \, q_{violating} - q_{limit} \, \|^2 \qquad V_{HAND} = \| \, x_{hand} - x_{target} \, \|^2, \tag{27}$$

$$V_{BLN} = \| \, x_{com} - x_{foot} \, \|^2 \qquad V_{MNP} = \| \, W\big(q - q_{mid}\big) \, \|^2 \, . \tag{28}$$

The abbreviations $JLC$, $BLN$, and $MNP$ stand for joint limit constraints, self-balance, and joint manipulability respectively. In addition, $q_{violating}$ is the vector of robot joints that, at a given time, violate joint limits, $q_{limit}$ is the vector of joint limit values, $x_{target}$ is an interactive hand position target, $x_{foot}$ is the position of the right foot, $q_{mid} = (q^-_{JL} + q^+_{JL})/2$ comprises the joint mid-range positions, $W = diag(\, q^+_{JL} - q^-_{JL} \,)$ is a normalizing matrix, and $q^+_{JL}$ and $q^-_{JL}$ are the upper and lower joint limits. In this experiment, for every task $k$, we use a simple PD control

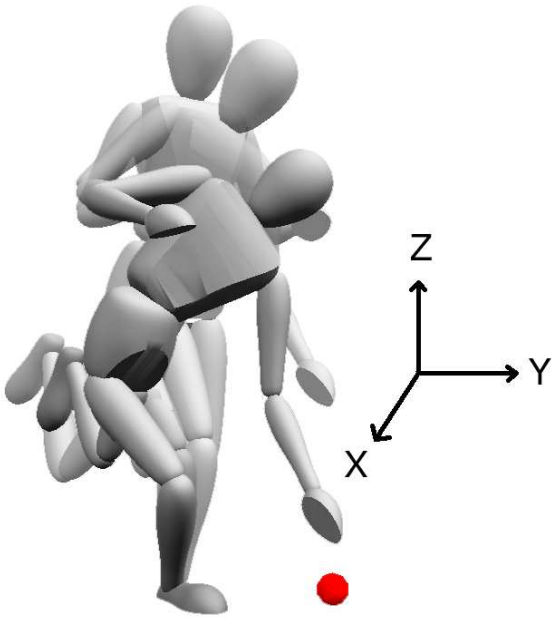


Fig. 3. **Hand Position Control under Joint Limit Constraints:** In this sequence, the robot is commanded to reach a target position with its left hand while maintaining self-balance and while complying with joint limit constraints. The robot is not able to reach its goal due to joint limits in the left elbow, right-leg, and upper body joints.

reference with velocity saturation:

$$\ddot{x}_{ref(k)} = -k_v\big(\dot{x}_k - \nu\dot{x}_{des(k)}\big), \tag{29}$$

$$\dot{x}_{des(k)} = \frac{k_p}{k_v}\nabla V_k \qquad \nu = min\bigg(1, \frac{v_{max(k)}}{||\dot{x}_{des(k)}||}\bigg), \tag{30}$$

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

To illustrate the advantages of prioritizing the tasks, we first study a controller that sets equal priorities for the self-balance and hand position subtasks. This condition is expressed by the following torque equation

$$\Gamma = \Gamma_{JLC(1)} + \Gamma_{BLN(2)} + \Gamma_{HAND(2)} + \Gamma_{MNP(3)}, \tag{31}$$

where the priorities are displayed in parentheses. We control the self-balance and hand position subtasks by combining them into a joined subtask with Jacobian matrix and control input defined by

$$J_{combined} = \begin{bmatrix} J_{BLN} \\ J_{HAND} \end{bmatrix} \qquad \ddot{x}_{ref(combined)} = \begin{bmatrix} \ddot{x}_{ref(BLN)} \\ \ddot{x}_{ref(HAND)} \end{bmatrix}. \tag{32}$$

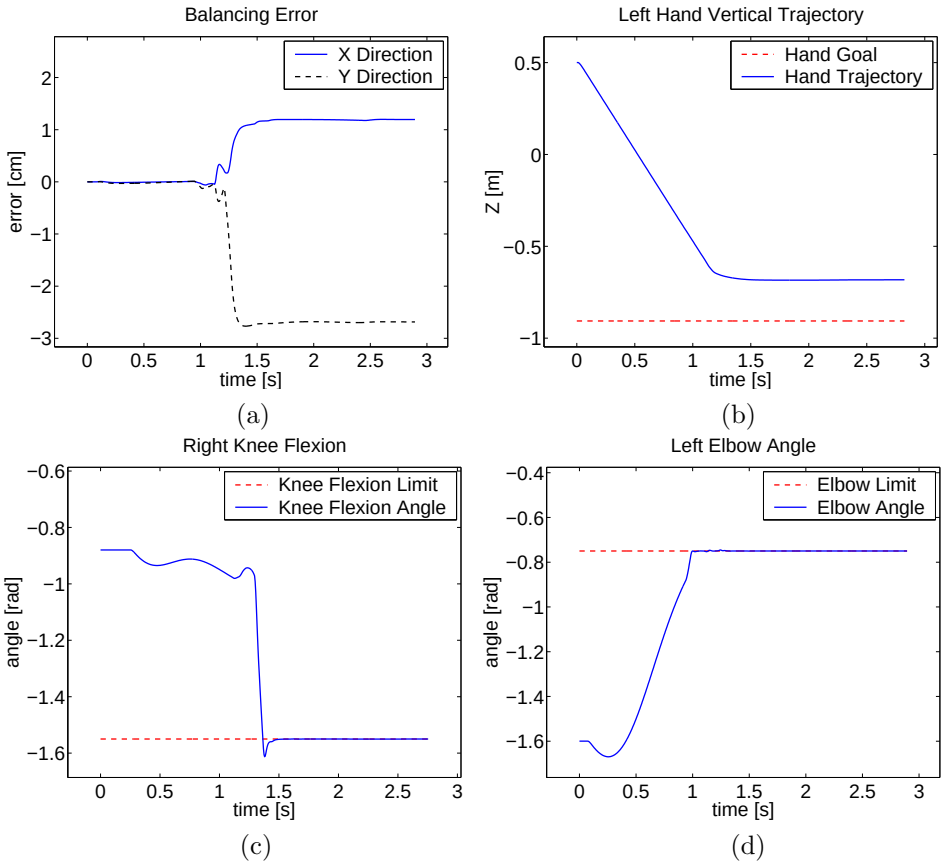


Fig. 4. **Data Recorded When the Self-Balance and Hand Position Tasks Share Priority:** When the knee flexion and left elbow-joint limits are reached (c and d), the self-balance error increases substantially (a). The hand's vertical position is not permitted to reach its goal, due to joint limits (d).

When we apply the controller defined in Equation (23) we obtain the results shown in Figure 4. The error in self-balance is zero while the hand moves down with steady speed. When the hip, elbow, and knee flexion joint limits are reached at $t = 0.9\ s$, $1\ s$, and $1.2\ s$ respectively, self-balance and hand position control cannot be accomplished simultaneously. Because these two subtasks have equal priority, an error will appear according to their control gains. The steady-state error associated with self-balance is 1 $cm$ in the $X$ direction and 3 $cm$ in the $Y$ direction, while the hand stops approximately 22 $cm$ away from its target due to joint limits.

Next, we assign higher priority to the self-balance subtask:

$$\Gamma = \Gamma_{JLC(1)} + \Gamma_{BLN(2)} + \Gamma_{HAND(3)} + \Gamma_{MNP(4)}. \tag{33}$$

We observe that even though the hip, elbow, and knee flexion joint limits are

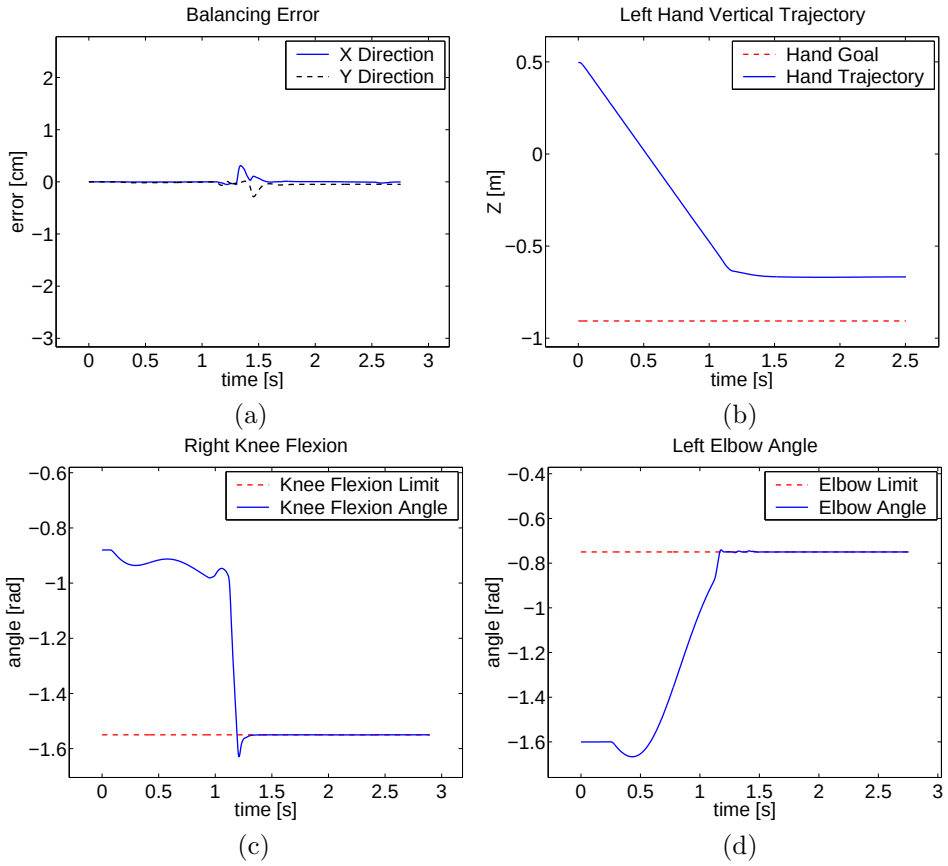


Fig. 5. **Data Recorded When the Self-Balance Task Precedes the Hand Position Task:** The self-balance error (a) stays small when the knee flexion and left elbow joint limits are reached (c and d). Because the hierarchy assigns it a higher priority, self-balance (a) remains undisturbed by hand position control (b).

reached, the maximum error in self-balance is only 2 $mm$ in the $X$ and $Y$ directions. The hand is still unable to reach its goal, also due to joint limits. Instead, it reaches the closest possible position 24 $cm$ away from its target, 2 $cm$ farther away than it reached the previous example. This difference is expected, because self-balance is now fully accomplished.

In conclusion, critical subtasks need to be assigned the highest priority to avoid conflicts in which these subtasks may be compromised by less important ones.

## 6. Example: Compliant Tasks

One of the advantages of a controller that provides decoupled closed loop dynamics at all levels is that the control gains can be greatly reduced without the tracking error being degraded. We can use this ability to add open-loop compliance to

desired subtasks. We shall study an example in which we combine four subtasks,

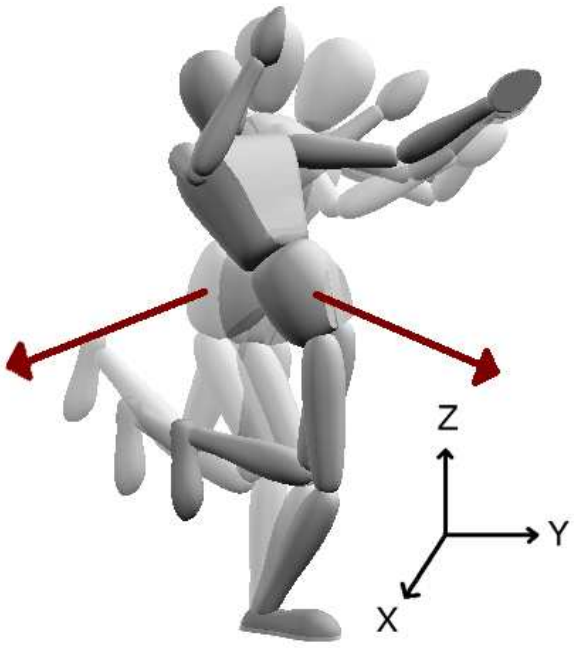


Fig. 6. **Compliant Upright Posture:** This sequence of snapshots shows an external force applied to the robot while the latter maintains self-balance, left-hand position control, and upright posture. The upright-posture gains are low to provide compliance. Notice that self-balance and left-hand position are unaffected.

self-balance, left-hand position control, upright posture, and joint manipulability, according to the torque equation

$$\Gamma = \Gamma_{BLN(1)} + \Gamma_{HAND(2)} + \Gamma_{UPR(3)} + \Gamma_{MNP(4)}, \tag{34}$$

where the new subscript $UPR$ stands for upright posture, and the number in parenthesis defines the arrangement in the control hierarchy. The self-balance, hand position, and joint-manipulability subtasks are controlled in the same manner as were in Section 5. The upright posture consists of aligning the chest's $Z$ axis (vertical direction) with a desired upright orientation. The orientation of the chest is represented by the quaternion $\lambda_{r(chest)}$; the desired upright orientation is represented by the quaternion $\lambda_{r(upright)}$. We also consider the $3 \times n$ angular Jacobian of the chest $J_{r(chest)}$. The Jacobian and the control reference of the upright posture is expressed as

$$J_{UPR} = S_z J_{r(chest)} \qquad \ddot{x}_{ref(UPR)} = -S_z(k_p \triangle\phi_{chest} + k_v \dot{\phi}_{chest}), \tag{35}$$

where $S_z$ is a selection matrix that chooses the chest vertical direction, $\triangle\phi_{chest} = E_r(\lambda_{r(chest)} - \lambda_{r(upright)})$ is the posture angular error, $E_r$ is a representation transformation, and $\dot{\phi}_{chest} = J_{r(chest)}\dot{q}$ is the instantaneous angular velocity of the chest.

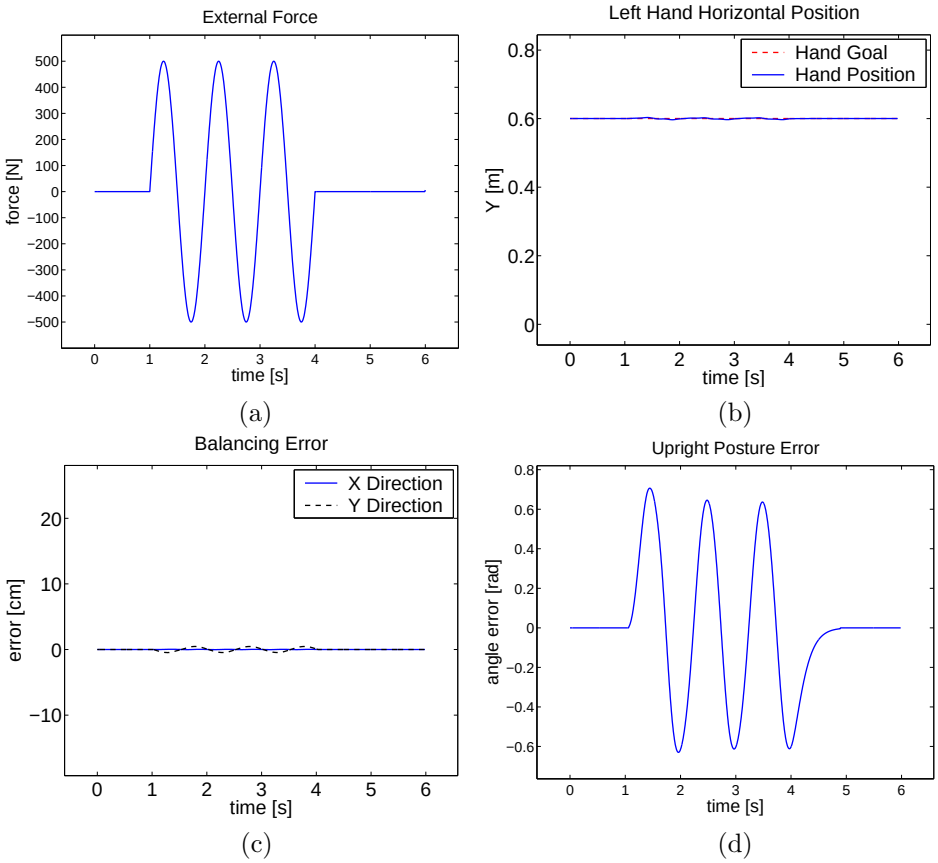


Fig. 7. **Data on Compliant Upright Posture:** When we apply an external force to the robot's hip (a), we induce a large oscillation to the upright posture (d) due to that task's low gain. In contrast, the self-balance (c) and hand's position errors (b) are small due to the high gains of these tasks.

We first study an example in which we allow low upright posture gains while maintaining high self-balance and hand position gains. In particular, we choose the following gains: $k_{p(BLN)} = 5000$, $K_{p(HAND)} = 1000$ and $k_{p(UPR)} = 50$ with a servo rate of 1 $kHz$. As shown in Figure 6, we apply an external force to the hip of the robot in the $Y$ direction. The shape of this force is sinusoidal; its amplitude is 500 $Nm$. The results are displayed in Figure 7.

The upright posture oscillates approximately $\pm 40^\circ$ due to the compliance in-

duced by the low gain. In contrast, the self-balance error oscillates $\pm 0.5$ $cm$, and the hand's position error oscillates $\pm 3.2$ $mm$. Notice, in Figure 6, that the global center of gravity of the robot stays centered above the right foot despite the large forces applied to the hip. The left hand position does not move either, due to the decoupled dynamics and high gain.

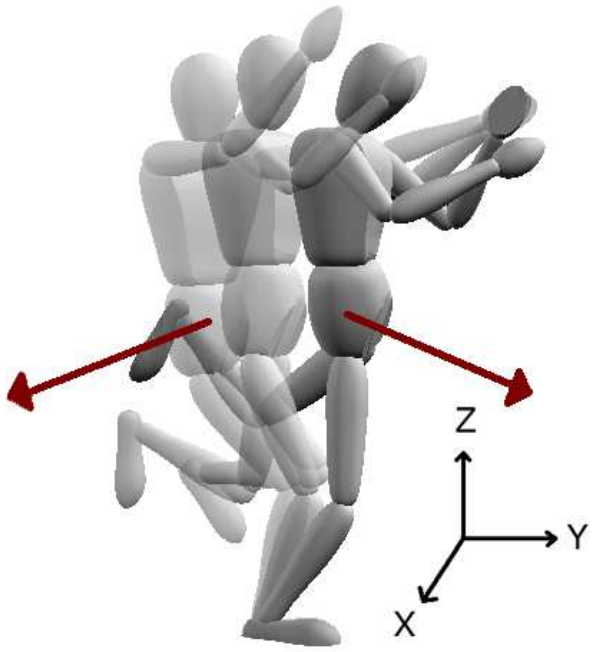


Fig. 8. **Compliant Self-Balance:** The same force as that in Figure 6 is applied while the robot maintains self-balance, left-hand position control, and upright posture. The control gain in the self-balance task is now very low. The upright posture and hand position are undisturbed.

We look at a second example by applying a low self-balance gain while maintaining high gains in the hand position control and upright posture. The gains are now changed to: $k_{p(BLN)} = 120$, $K_{p(HAND)} = 1000$ and $k_{p(UPR)} = 5000$. In Figure 8, we show snapshots of this experiment, with application of external force the same as in the previous example. The results are displayed in Figure 9. Observe that the self-balance task oscillates $\pm 15$ $cm$ approximately, whereas the upright posture oscillates only $\pm 0.6°$. The hand's position error oscillates $\pm$ 3.5 $mm$. Figure 8 shows that the upright posture remains unchanged. Notice that the torso twists to the sides; this movement is a desired effect that results from controlling only the vertical orientation of the chest.

In conclusion, by adjusting the gains we can control the compliance of selected subtasks.

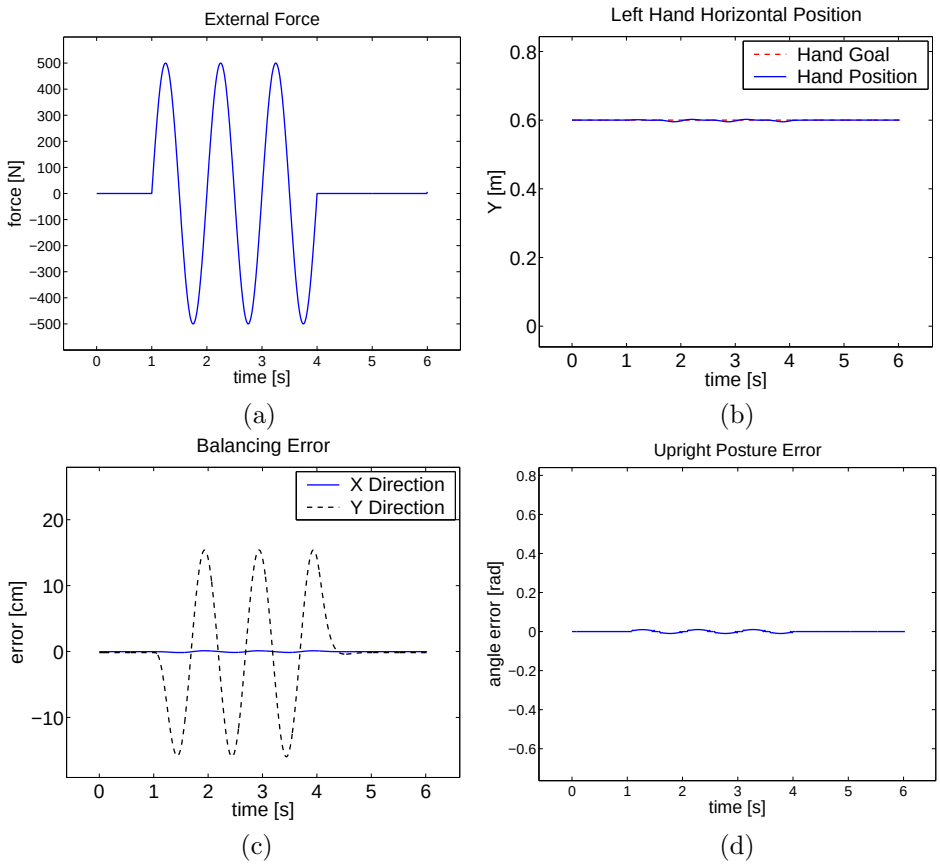


Fig. 9. **Data on Compliant Self-Balance:** An external force is applied to the robot's hip (a). A large oscillation is induced in the self-balance task (c), due to the latter's low gain. The hand's position and upright posture errors (b and d) remain close to zero, due to the associated high gains.

## 7. Conclusion

To become practical, humanoid robots must be controlled interactively while complying with physical constraints and high priority subtasks. However, most control approaches are based on offline optimization or computationally expensive trajectory planning. In contrast, our approach, addresses interactive control of global tasks by explicitly defining the space of motion that complies with the constraints and essential subtasks. In this paper, we proposed to break down whole-body tasks into simpler subtasks, and we presented a methodology to check for task feasibility at runtime. This allows us to change interactively individual task parameters, such as gains and target positions, without the need to check their feasibility in advance.

Our controller characterizes for the first time the mass properties of all subtasks in the hierarchy, allowing us to implement operational-space dynamic controllers

for every subtask.

We envision humanoid robots making contact with multiple body parts at once. Our framework allows us to individually control each contact point, while accomplishing the global task.

While today the interactive control of humanoids is limited to the online selection of a few preplanned motions, with this new controller, we construct complex behaviors at runtime by adding new subtasks, or by changing the individual task parameters.

## Acknowledgments

We are grateful for the contributions of Jae-Heung Park, James Warren, Vincent De Sapio, Tine Lefebvre, François Conti, Irena Pashchenko, Kyong-Sok Chang, Diego Ruspini, and Oliver Brock. Our work was supported by the Honda Humanoid Project [3,17].

## References

1. 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.
2. 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.
3. 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.
4. 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.
5. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–8, 1986.
6. O. Khatib. A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *International Journal of Robotics and Automation*, RA–3(1):43–53, February 1987.
7. 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.
8. 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.
9. O. Khatib. Inertial Properties in Robotics Manipulation: An Object-Level Framework. *International Journal of Robotics Research*, 14(1):19–36, 1995.
10. 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.
11. O. Khatib, O. Brock, K.C. Chang, D. Ruspini, L. Sentis, and S. Viji. Human-centered robotics and interactive haptic simulation. *International Journal of Robotics Research*, 23(2), February 2004.
12. 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.
13. 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.
14. Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based control of robot manipulators. *International Journal of Robotics Research*, 6(2):3–15, 1987.
15. K. C. Park, P. H. Chang, and S. Lee. Analysis and control of redundant manipulator dynamics based on an extended operational space. *Robotica*, 19:649–662, 2001.
16. 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.
17. 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.
18. 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.

**Luis Sentis** is a Research Assistant at the Stanford Artificial Intelligence Laboratory. He earned his B.S. degree from the Technical University of Catalonia and his M.S. from Stanford University, both in Electrical Engineering. His research focuses on interactive robot control for human environments.

**Oussama Khatib** is Professor of Computer Science and Mechanical Engineering at the Stanford Artificial Intelligence Laboratory. His research interests include autonomous robots, human-centered robotics, human-friendly robot design, dynamic simulations, and haptic interactions.