International Journal of Humanoid Robotics

SYNTHESIS OF WHOLE-BODY BEHAVIORS THROUGH HIERARCHICAL CONTROL OF BEHAVIORAL PRIMITIVES

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

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Communicated by (xxxxxxxxxx)

To synthesize whole-body behaviors interactively, multiple behavioral primitives need to be simultaneously controlled, including those that guarantee that the constraints imposed by the robot's structure and the external environment are satisfied. Behavioral primitives are entities for the control of various movement criteria, e.g. primitives describing the behavior of the center of gravity, the behaviors of the hands, legs, and head, the body attitude and posture, the constrained body parts such as joint-limits and contacts, etc. By aggregating multiple primitives, we synthesize whole-body behaviors. For safety and for efficient control, we establish a control hierarchy among behavioral primitives, which is exploited to establish control priorities among the different control categories, i.e. constraints, operational tasks, and postures. Constraints should always be guaranteed, while operational tasks should be accomplished without violating the acting constraints, and the posture should control the residual movement redundancy. In this paper we will present a multi-level hierarchical control structure that allows the establishment of general priorities among behavioral primitives, and we will describe compliant control strategies for efficient control under contact interactions.

Keywords: Behavioral primitives, control hierarchy, whole-body behaviors.

1. Introduction

Emerging applications of humanoids demand higher and higher degrees of autonomy for efficient interactions in human-populated environments. Controlling humanoids in these environments requires us to synthesize and change complex whole-body behaviors on-demand in the presence of high uncertainty. To synthesize whole-body behaviors on-demand we have developed a behavior-oriented methodology where multiple behavioral primitives are controlled simultaneously. New behaviors are created by adding or removing individual, or collections of, pre-designed behavioral primitives, without the need to interrupt the movement. To guarantee the safety of the robot and its environment we have designed a control hierarchy among primitives, where the control of the most critical ones (i.e. constraints) is always guaranteed while non-safety related primitives (i.e. operational tasks and postures) are controlled without violating higher priority controls. In this context we distinguish three priority levels in the hierarchy: constraints (such as contacts, near-body objects, joint-limits, self-collisions), operational tasks (i.e. manipulation and locomotion), and postures (i.e. the residual motion), which should be controlled with different priority assignments.

In this paper, we will describe in detail this hierarchy based on projecting the control of lower priority primitives into the motion null-space of higher priority primitives. This ordering is selected to reflect the relative importance among the controlling primitives. Constraints, operational tasks, and postures are treated as independent control entities. The hierarchies imposed among these categories allows us to study movement feasibility in realtime and stop or change the global behavior if needed. In this context, infeasible movements result from the presence of constraining objects or from inconsistent or conflicting control primitives.

Control methodologies based on null-space projections[10,11,12,13] have traditionally treated constraints as secondary motion criteria, being unable to fulfill the acting constraints in the case of conflicting tasks. In contrast, our methodology integrates constraints in the control formulation as primary controls and projects the operational tasks and the posture primitives into the constraint motion null-space, thus eliminating the motion components that could cause constraint violations. Additionally, this formulation introduces null-space projections directly at the kinematic level, allowing us to implement operational space compliant controllers while complying with the constraints and other higher priority tasks.

This paper is organized as follows. In Section 2 we describe previous related work, and also lay the mathematical foundations for this research based on our previous work.[9] In Section 3 we introduce a formulation that integrates constraints directly into the control formulation and establishes a control hierarchy among behavioral categories. Section 4 presents a multi-level prioritized framework that allows us to establish multiple priority levels among categories. We briefly explore the experimental setup in Section 5 and demonstrate the framework's capability by evaluating an example scenario. Finally concluding remarks appear in Section 6.

2. Related work

Task-space control was first studied at the inverse kinematic level.[1,3,14] It provided the ability to control specific parts of the robot's body in local task space. In 1987, the *Operational Space Formulation* [6,7] was introduced to address the dynamic interaction between the robot's task-space motion and force. To characterize the additional task redundancy, the operational space formulation defines a dynamically consistent task null-space. Multiple operational tasks can be controlled if they are combined into a single task vector and additional criteria can be controlled within the task-consistent null-space. We previously presented[9] a broader extension of the operational space formulation that efficiently characterizes and controls additional criteria projected into the task null-space.

We first review the fundamental mathematics that will be used in Section 4 to build a multi-level control hierarchy for the synthesis of whole-body behaviors. We begin by describing the robot's joint space dynamics in terms of its joint coordinates q,

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

where Γ 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.

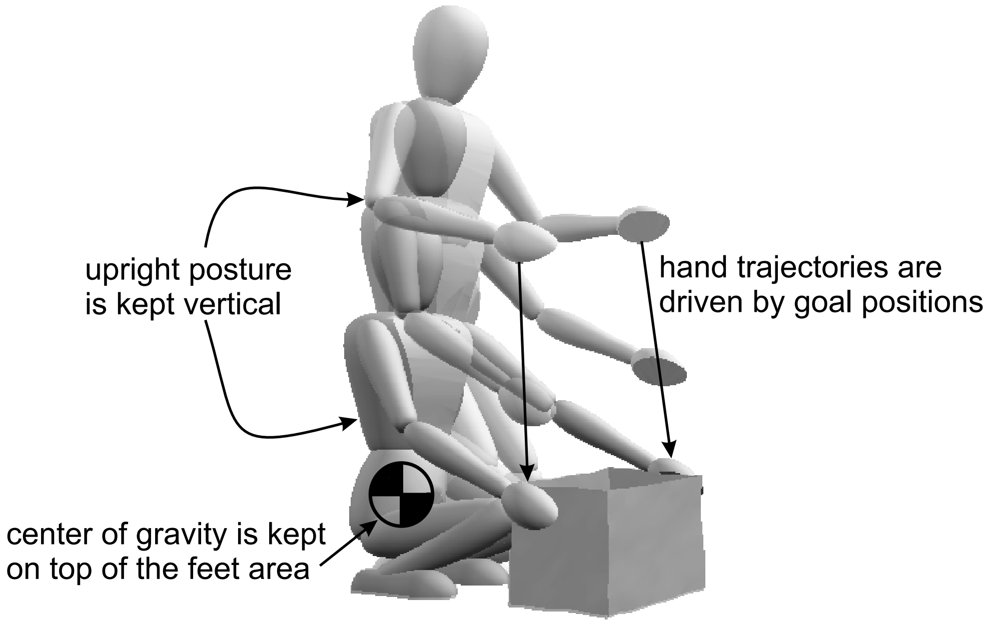

Fig. 1. **Task and posture decomposition:** In this sequence, we control the robot's hands to grab a box while maintaining body self-balance (based on the control of the global center of gravity) and control of the torso's upright posture.

The operational space formulation describes the torque level decomposition of an operational task and a posture (i.e. a secondary control criteria projected into the task-consistent null-space) according to the torque equation

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

For an operational task with coordinates $x_t(q)$ (in general, this is any arbitrary function of q) and Jacobian $J_t(q) = \partial x_t(q)/\partial q$, the control

$$\Gamma_{\text{task}} = J_t^T F_t, \tag{3}$$

$$F_t = \Lambda_t \, \ddot{x}_{t(\text{ref})} + \mu_t + p_t, \tag{4}$$

provides the decoupled behavior $\ddot{x}_t = \ddot{x}_{t(\text{ref})}$, where $\ddot{x}_{t(\text{ref})}$ is a reference input at the acceleration level, and Λ_t, μ_t, and p_t are the mass matrix, Coriolis/centrifugal forces, and gravity forces respectively. A task-consistent null-space[5] is defined as $N_t(q) = I - \overline{J}_t J_t$, where $\overline{J}_t = A^{-1} J_t^T \Lambda_t$ is the dynamically-consistent generalized inverse of J_t.

The control input, Γ_{posture}, is used to control a posture criteria with coordinates $x_p(q)$ and Jacobian $J_p = \partial x_p / \partial q$. For example, an upright posture would involve the control of the vertical orientation of the robot's torso reference frame. For compliant control of the posture[8] and to establish a hierarchy where the posture is projected within the task null-space (so that constraints are always satisfied), we define the control forces

$$\Gamma_{\text{posture}} = J_{p|t}^T F_{p|t}, \tag{5}$$

$$F_{p|t} = \Lambda_{p|t} \, \ddot{x}_{p(\text{ref})} + \mu_{p|t} + p_{p|t}. \tag{6}$$

Here $J_{p|t} = J_p N_t$ is a projection of the posture Jacobian into the task null-space,[9] $\ddot{x}_{p(\text{ref})}$ is a control reference (a desired trajectory in free space, a desired force, or a combination of both) for the posture, and $\Lambda_{p|t} = (J_{p|t} A^{-1} J_{p|t})^{-1}$, $\mu_{p|t}$, and $p_{p|t}$ are dynamic quantities. Notice that $\Lambda_{p|t}$ defines the inertial properties of the posture provided that no coupling is induced into the task. This control can then be used to provide compliant control at the posture level.

Under these conditions, Equation (2) becomes

$$\Gamma = \left(J_t^T F_t\right) + \left(J_{p|t}^T F_{p|t}\right), \tag{7}$$

revealing the implementation of two operational-space control strategies for the task and posture levels, while the projection of the posture into the task null-space is integrated into $J_{p|t}$.

If $J_{p|t}$ is full rank, the posture is feasible (within the hierarchy) and this controller will yield the decoupled behavior $\ddot{x}_p = \ddot{x}_{p(\text{ref})}$. Otherwise, the posture is singular or conflicts with the task and therefore the posture trajectories should be modified or its movement should be halted. Figure 1 illustrates the control of an upright posture with simultaneous control of the position of the hands and the robot's global center of gravity, where these last two controls represent operational tasks.

To synthesize whole-body movements, we must control multiple operational tasks and postures while satisfying all acting constraints on the robot's body. Furthermore, to integrate these constraints and establish further priorities between the different control categories, we need to extend the hierarchy expressed in Equations (3) and (5) to multiple levels of priorities.

3. Integration of constraints

The control of robots under constraints has been investigated since the mid 1970s. In this context, redundancy has received much attention, with most algorithms being based on instantaneous kinematic solutions with constraint-handling criteria projected into the task null-space, i.e.

$$dq = J^{\#}\, dx + \left(I - J^{\#}\, J\right) dq_{\text{null}}. \tag{8}$$

Here, J is the Jacobian of an operational task, x is a desired task-space trajectory, $(I - J^{\#}\, J)$ is the kinematically-consistent null-space, and q_{null} is a vector used to control secondary motion criteria to avoid constraints. The problem with these methodologies is that the task is allowed to violate the constraints if no additional redundancy is available to avoid them. Additionally, because these methodologies are based on inverse kinematics, they do not provide support for impedance regulation.

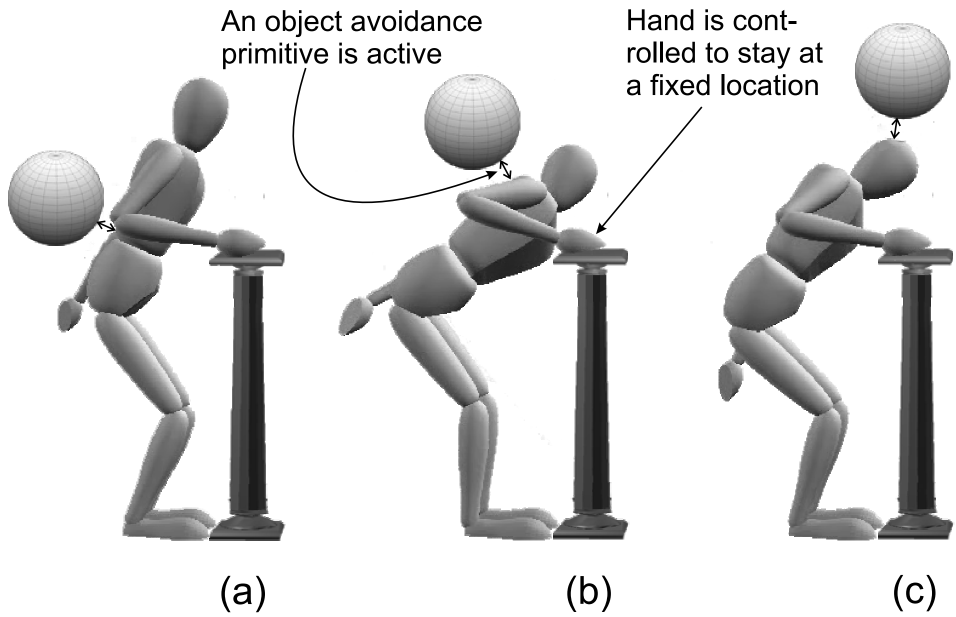

Fig. 2. **Collision avoidance and control of multiple task primitives:** This sequence depicts a robot avoiding an obstacle that is moved interactively towards several points near the robot's body. While the right hand is commanded to stay at a fixed location, the center of gravity is controlled for balance, and the additional posture is controlled to minimize distance with respect to a symmetrical posture.

Based on the operational space formulation for redundant robots, further represented by the torque decomposition

$$\Gamma = J^T F + N^T \Gamma_{\text{null}}, \tag{9}$$

where F are forces that control an operational task, Γ_{null} is an additional control component projected in the task null-space, and $N = (I - \overline{J}J)$ is the task null-space, we present a new approach where constraints are accounted for as high priority motion controls and the operational task is projected into the constraint null-space. This decomposition takes the mathematical form

$$\Gamma = J^T_{\text{constraint}} F_{\text{constraint}} + N^T_{\text{constraint}} \Gamma_{\text{task}}, \tag{10}$$

where $J_{\text{constraint}}$ is the Jacobian associated with the constrained points, $F_{\text{constraint}}$ is a vector of forces to control the distance to, or the forces onto the constrained body parts, and $N_{\text{constraint}} = (I - \overline{J}_{\text{constraint}} J_{\text{constraint}})$ is the dynamically-consistent null-space associated with the constrained space. This projection ensures that the operational task does not introduce acceleration components into the constrained directions. Therefore, in the case of a motion conflict, the task would be unable to operate within this projection.

To synthesize complex whole-body behaviors, humanoids need to control multiple body parts at once while satisfying all acting constraints. In Figure 2 we depict a sequence where the robot's right hand is controlled to remain at a fixed location while an object is moved towards the robot's body, constraining the global movement. To accomplish the task and handle the constraint efficiently, we apply the control described in Equation (10).

In the next section we extend this framework by introducing a recursive hierarchical structure that not only serves as a platform to integrate multiple constraints but also can be used to create additional hierarchies between the desired operational tasks and posture primitives.

4. Multi-level hierarchy

We propose a multi-level control hierarchy that extends the task and posture decomposition previously described. We create this hierarchy to integrate constraints and organize additional tasks according to desired priorities, while optimizing the execution of the global task. This framework ensures that all constraints affecting the robot simultaneously are never violated, and allows us to impose hierarchies between tasks that may conflict while in motion. The criteria to organize these priorities will be discussed in a future paper.

A humanoid robot must accomplish a collection of operational tasks while satisfying several constraints acting on the robot's body. At the same time the additional redundancy (a.k.a the posture) must also be controlled. Let us suppose there are N behavioral primitives (constraints, operational tasks, and postures) controlling the robot's behavior at a given time. Let us also assume that constraints are to take the top-most priorities in the hierarchy and that operational tasks are to be projected into the lower-most levels, ensuring that constraints will be fulfilled first. For the time being let us assume that the set of controls to handle the acting constraints is combined into a single non-prioritized torque control reference, $\Gamma_{\text{constraints}}$. On the other hand, for a set of N task primitives defining the robot's whole-body task, let us associate a set of task coordinate vectors $\{x_k(q) \,|\, k = 1, 2, \ldots, N\}$ and task Jacobians $J_k(q) = \partial x_k(q)/\partial q$, where the numbering represents also the desired ordering of priorities. This ordering is selected to reflect the relative importance of the controlled tasks. For example, we may want to give higher priority to the control of the robot's center of gravity over the control of its hands. The following torque equation embodies a multi-level control hierarchy integrating both constraints and tasks into a single torque control reference:

$$\Gamma = \Gamma_{\text{constraints}} + N^T_{\text{constraints}}(\Gamma_{\text{task}(1)} + N^T_{\text{task}(1)}(\Gamma_{\text{task}(2)} + N_{\text{task}(2)} \tag{11}$$

$$\times (\Gamma_{\text{task}(3)} + \cdots N^T_{\text{task}(N-1)}\Gamma_{\text{task}(N)}))) \tag{12}$$

Here, $N_{\text{constraints}}$ is a null-space associated with all constrained points in the robot's body and will be discussed in the follow-up paper, the $\Gamma_{\text{task}(k)}$' s are the controls for the individual operational tasks, and the $N_{\text{task}(k)}$' s are the associated dynamically-consistent null-spaces. This nested topology can be simplified by defining an extended null-space matrix containing the null-spaces of all preceding constraints and tasks:

$$N_{\text{prec}(k)} = N_{\text{task}(k-1)} N_{\text{task}(k-2)} \cdots N_{\text{task}(1)} N_{\text{constraints}}, \tag{13}$$

where $\text{prec}(k) = \{\text{task}(k-1), \cdots, \text{task}(1), \text{constraints}\}$. With this notation, Equation (12) becomes

$$\Gamma = \Gamma_{\text{constraints}} + \Gamma_{1|\text{prec}(1)} + \Gamma_{2|\text{prec}(2)} + \cdots + \Gamma_{N|\text{prec}(N)}, \tag{14}$$

where $\Gamma_{k|\text{prec}(k)} = N^T_{\text{prec}(k)}\Gamma_{\text{task}(k)}$ are the prioritized controls, and the subscript $k|\text{prec}(k)$ indicates that the k^{th} task is projected into the null-space of all preceding tasks and constraints. To provide compliant control solutions within the hierarchy we project the task Jacobians into the null-spaces $N_{\text{prec}(k)}$, i.e.

$$J_{k|\text{prec}(k)} \triangleq J_k N_{\text{prec}(k)}, \tag{15}$$

and we associate an extended inertia matrix:

$$\Lambda_{k|\text{prec}(k)} = \left(J_{k|\text{prec}(k)} A^{-1} J^T_{k|\text{prec}(k)}\right)^{-1}. \tag{16}$$

The dynamic behavior in task space can be obtained by projecting the robot's joint dynamics into the associated task space, i.e.

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

$$\Lambda_{k|\text{prec}(k)}\ddot{x}_{k|\text{prec}(k)} + \mu_{k|\text{prec}(k)} + p_{k|\text{prec}(k)} = F_{k|\text{prec}(k)}, \tag{17}$$

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

4.1. *Recursive null-spaces*

The null-space of task k in the hierarchy represents the space of motion with no acceleration effects on any of the preceding levels, or equivalently

$$\forall\, i \in \text{prec}(k) \quad J_i A^{-1} N^T_{\text{prec}(k)} = 0. \tag{18}$$

This mathematical constraint leads to the following unique solution

$$N_{\text{prec}(k)} = I - \sum_{i=1}^{k-1} \overline{J}_{i|\text{prec}(i)} J_{i|\text{prec}(i)}, \tag{19}$$

where $\overline{J}_{i|\text{prec}(i)} = A^{-1} J^T_{i|\text{prec}(i)} \Lambda_{i|\text{prec}(i)}$, are the individual dynamically-consistent inverses of the prioritized Jacobians.

Proof by Induction of Equation (19):

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

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

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

4.2. *Control of behavioral primitives within the hierarchy*

We accomplish efficient control of tasks $x_k(q)$ within the hierarchy (i.e. fulfilling all preceding controls) by choosing the operational-space control torque

$$\Gamma_{k|\text{prec}(k)} = J^T_{k|\text{prec}(k)} F_{k|\text{prec}(k)}, \tag{20}$$

$$F_{k|\text{prec}(k)} = \Lambda_{k|\text{prec}(k)} \ddot{x}_{k(\text{ref})} + \mu_{k|\text{prec}(k)} + p_{k|\text{prec}(k)}. \tag{21}$$

If $J_{k|\text{prec}(k)}$ is full rank, this controller will yield the decoupled behavior $\ddot{x}_k = \ddot{x}_{k(\text{ref})}$, where $\ddot{x}_{k(\text{ref})}$ is the control reference at the acceleration level.

4.3. *Movement feasibility*

An operational task k is singular within the hierarchy if the extended Jacobian $J_{k|\text{prec}(k)}$ drops rank. In this case, the inverted prioritized inertia matrix has the following eigen-decomposition

$$\Lambda^{-1}_{k|\text{prec}(k)} = J_{k|\text{prec}(k)} A^{-1} J^T_{k|\text{prec}(k)} = \begin{pmatrix} U_{r(k)} \; U_{n(k)} \end{pmatrix} \begin{pmatrix} \Sigma_{r(k)} & \\ & 0 \end{pmatrix} \begin{pmatrix} U^T_{r(k)} \\ U^T_{n(k)} \end{pmatrix}, \tag{22}$$

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 fully control $\ddot{x}_k$. However, by choosing the control input

$$F_{k|\text{prec}(k)} = \Big(U_{r(k)} \Sigma^{-1}_{r(k)} U^T_{r(k)}\Big) \ddot{x}_{k(\text{ref})} + \mu_{k|\text{prec}(k)} + p_{k|\text{prec}(k)}, \tag{23}$$

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

The singular values of $J_{k|\text{prec}(k)}$, or the eigenvalues of $\Lambda_{k|\text{prec}(k)}$ as well, give us a mechanism to study the feasibility of individual tasks. In the case that an active task becomes ill-conditioned we say that the robot's movement is infeasible. We can then modify the task trajectory or remove its control while the control of other higher priority tasks such as balancing or control of the contact points is maintained. These issues will be discussed in detail in the follow-up paper.

In the next section we put this control framework into practice by applying it to the study of a complex behavior formed by controlling multiple behavioral primitives under joint-limit constraints.

5. Example: hand position control under joint-limit constraints

We explore an example of a whole-body behavior where the robot's right hand position is interactively operated by a user while the controller handles joint-limit constraints (see Fig. 3. This complex behavior is complemented with additional tasks to control the robot's center of gravity and to maintain a body symmetry posture. For this simulated experiment, we use a humanoid robot model consisting of 24 degrees of freedom: 2×6 for the legs, 2×4 for the arms, 2 for the torso, and 2 for head. The robot's height is 1.65 m and its weight is 71 Kg. The robot's center of gravity (a.k.a. COG) task is defined by the global center of gravity coordinates and the associated Jacobian, which can be expressed as

$$x_{cog} = \frac{1}{M} \sum_{i=1}^{n} m_i \cdot x_{com(i)} \,, \qquad J_{cog} = \frac{1}{M} \sum_{i=1}^{n} m_i \cdot J_{com(i)}, \tag{24}$$

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 Cartesian position x_{hand}. On the other hand, the symmetry posture is designed to maximize joint range. To control all these behavioral primitives simultaneously we first define the following goal criteria in the form of artificial potential fields:[4]

$$V_{JLC} = \| \, q_{violating} - q_{limit} \, \|^2 \qquad V_{HAND} = \| \, x_{hand} - x_{target} \, \|^2, \tag{25}$$

$$V_{COG} = \| \, x_{cog} - x_{feet} \, \|^2 \qquad V_{SYM} = \| \, W\big(q - q_{mid}\big) \, \|^2 \,. \tag{26}$$

The abbreviations JLC, COG, and SYM stand for joint-limit constraints, center of gravity, and body symmetry control respectively. In addition, $q_{violating}$ is the vector of robot joints that, at a given time, have penetrated predefined joint-limit activation zones, q_{limit} is the vector of desired safety positions (away from the hard limits), x_{target} is a hand target position (commanded externally from a 3D tracking device), x_{feet} is the horizontal position at the center of the feet's convex hull, $q_{mid} = (q^-_{JL} + q^+_{JL})/2$ 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 limit specifications. In this experiment, every task primitive k is controlled through a

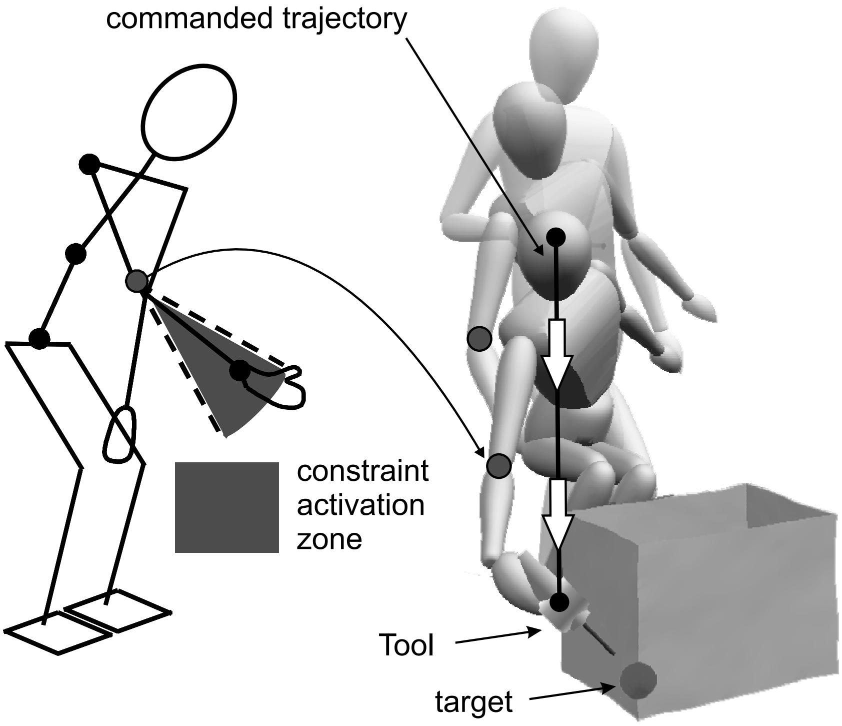

Fig. 3. **Hand position control under joint limit constraints:** In this sequence, the robot is commanded to reach a target position with its right hand while controlling the center of gravity and while complying with joint limit constraints. When several joint-limits are reached, including the right elbow, right-leg, and upper body joints, the hand is unable to move further away. Due to the projection of the hand task into the constraint null-space, joint limits are never violated during motion.

simple PD controller that includes velocity saturation, i.e.

$$\ddot{x}_{k(\mathrm{ref})} = -k_v\big(\dot{x}_k - \nu\,\dot{x}_{k(\mathrm{des})}\big), \tag{27}$$

$$\dot{x}_{k(\mathrm{des})} = \frac{k_p}{k_v}\nabla V_k\,, \qquad \nu = min\bigg(1, \frac{v_{max(k)}}{||\dot{x}_{k(\mathrm{des})}||}\bigg), \tag{28}$$

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

The control of joint-limit constraints is integrated by using the top-most priority level as specified in Equation (12), while the operational tasks are projected into the constraint null-space. Furthermore, we organize the tasks themselves into a hierarchy to guarantee that, in case of a conflict, the center of gravity control (a more critical task) will prevail over the control of the hand task. But first, to evaluate the performance and determine the optimal ordering we examine a scenario where the center of gravity control shares control priority with the hand position control,

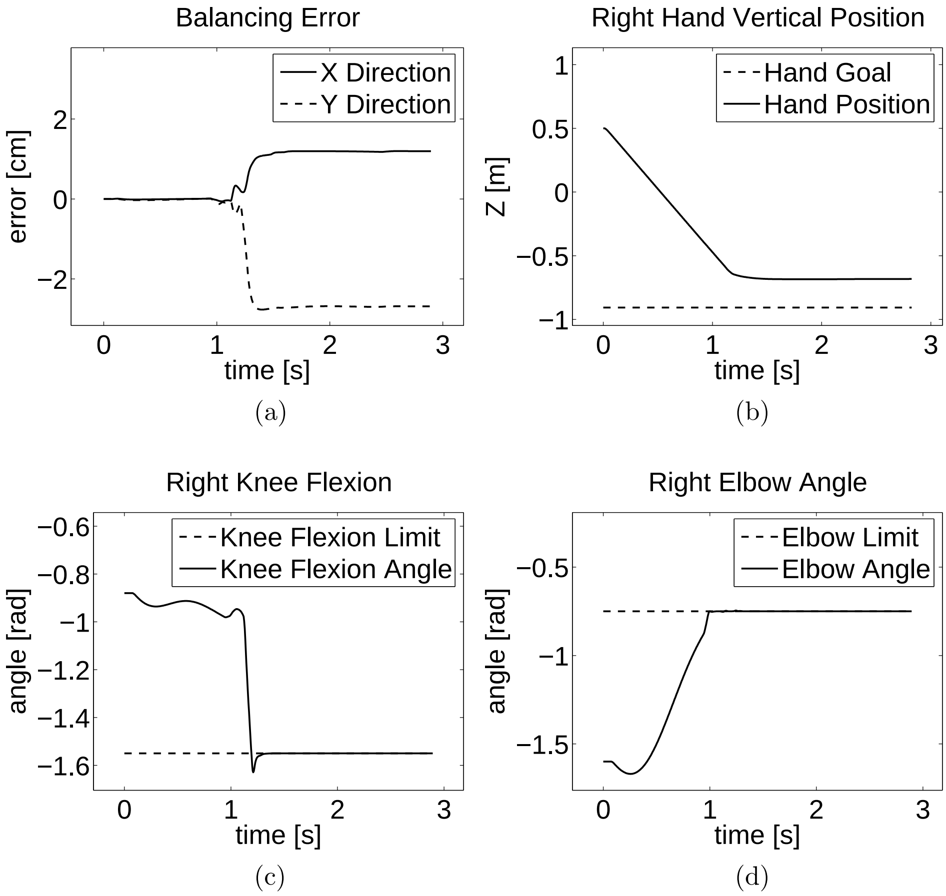

Fig. 4. **Data recorded when the center of gravity and the hand position tasks share priority:** When the knees flexion and right elbow joint-limits are reached (c and d), the right hand stops its motion (d), remaining at a fixed distance from its goal. However, the center of gravity horizontal position cannot be maintained (a), because its control is directly affected by the hand control.

i.e.

$$\Gamma = \Gamma_{JLC} + N_{JLC}^T \bigg(\Big(\Gamma_{COG} + \Gamma_{HAND} \Big) + N_{\{COG,\,HAND\}}^T \Gamma_{SYM} \bigg). \tag{29}$$

Here, N_{JLC} is the constraint null-space and $N_{\{COG,\,HAND\}}$ is the combined center of gravity and hand position null-space. As described earlier, constraints need to be always placed at the top of the chain, while tasks are controlled with lower priority to avoid constraint violations. Finally, the posture is projected into both the constraint and task null-spaces to access the residual redundancy. Since the center of gravity task and the hand task share priority, we can combine them into a joined operational task with a combined Jacobian and control input defined by

$$J = \begin{pmatrix} J_{COG} \\ J_{HAND} \end{pmatrix}, \qquad \ddot{x}_{\mathrm{ref}} = \begin{pmatrix} \ddot{x}_{COG(\mathrm{ref})} \\ \ddot{x}_{HAND(\mathrm{ref})} \end{pmatrix}. \tag{30}$$

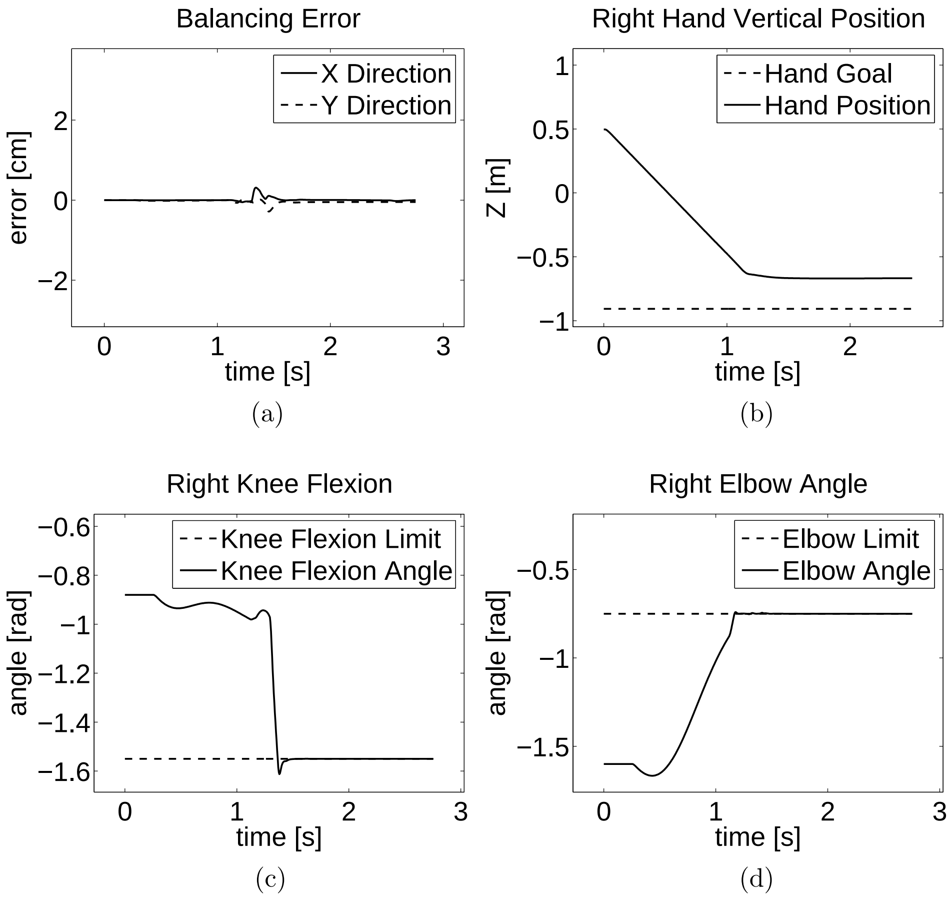

Fig. 5. **Data recorded when the center of gravity task precedes the hand position task:** The center of gravity error stays small (a) when the knee flexion and right elbow joint limits are reached (c and d). Because the hierarchy assigns higher priority to the center of gravity task, it maintains its desired goal position (above the robot's feet) at all times, while the robot's hand is prevented from reaching its goal (b), thus avoiding falling.

The results of this control are shown in Figure 4. A desired target for the robot's right hand position is fixed at the bottom of the image. While in motion, the error in the center of gravity horizontal position is initially zero while the hand moves down at 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, the center of gravity and the hand position conflict in their control (cannot be simultaneously accomplished). As a result, an error appears in both tasks according to their control gains. The steady-state errors for the center of gravity task are 1 *cm* in the X direction and 3 *cm* in the Y direction, while the hand stops approximately 22 *cm* away from its target.

To demonstrate the effectiveness of further prioritization, we reorganize the control by assigning higher priority to the center of gravity task and further projecting the hand position task into the COG null-space:

$$\Gamma = \Gamma_{JLC} + N_{JLC}^T \Big(\Gamma_{COG} + N_{COG}^T \Big(\Gamma_{HAND} + N_{HAND}^T \Gamma_{SYM} \Big) \Big). \qquad (31)$$

For the same experiment, we observe (see Figure 5) that even though the hip, elbow, and knee flexion joint limits are reached, the maximum error in the center of gravity position is only 2 *mm* in the X and Y directions. The hand control is now the only task unable to reach its goal. Instead, it reaches the closest possible position 24 *cm* away from its target. Thus, this second control is much more effective since we not only satisfy joint-limit constraints, but also prevent the robot from falling down.

In conclusion, the integration of constraints at the top-most level of the hierarchy has been demonstrated to be an effective methodology to avoid constraint violations. At the same time, critical tasks need to be assigned second highest priorities to avoid conflicts in which they may be compromised by less important ones.

6. Summary and discussion

To facilitate the application of humanoids to human environments we must be able to control these robots interactively while handling physical and environmental constraints affecting multiple body parts. Although motion planning techniques can give us local trajectories for walking and manipulation tasks while avoiding obstacles, the methodologies presented in this paper address other aspects of the robot's motion where motion planning is not applicable. Our research has addressed a wide set of constraints, such as joint-limits, collision avoidance, and self-collision avoidance, based on reactive techniques at the whole-body level. In this context, operational tasks (the tasks controlling manipulation, locomotion and vision) are projected into the constraint null space to avoid constraint violations. At the same time multiple operational tasks can be combined and further organized into hierarchies. Our major contribution is in presenting a novel and unified framework that is based on robust theoretical results.

Additionally, for safety under high environmental uncertainty and to provide robust contact interactions, this framework provides compliant controllers for all behavioral primitives (i.e. constraints, operational tasks, and posture primitives).

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 primitives, or by changing their governing parameters. This methodology allows us to synthesis new behaviors on-demand while handling all acting constraints.

Acknowledgments

This research has been supported by the Honda Grant SPO-21678.[2,15] We are grateful for the contributions of Mattan Erez and Vincent De Sapio in the editing of this manuscript.

References

1. 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.
2. 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.
3. 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.
4. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–8, 1986.
5. 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.
6. O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics Research*, 3(1):43–53, 1987.
7. 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.
8. O. Khatib. Inertial Properties in Robotics Manipulation: An Object-Level Framework. *International Journal of Robotics Research*, 14(1):19–36, 1995.
9. 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.
10. S.J. Kwon, W.K.Chung, Y.Youm, and M.S.Kim. Self-collision avoidance for n-link redundant manipulators. In *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pages 937–942, Charlottesville, USA, October 1991.
11. A. Liegois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7:868–871, 1977.
12. F. Lingelbach. Path planning for mobile manipulation using probabilistic cell decomposition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2807–2812, Stockholm, Sweden, September-October 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. 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.

Luis Sentis is a Ph.D. candidate in Electrical Engineering at the Stanford Artificial Intelligence Laboratory. His research focuses on the synthesis and control of low-level behaviors in humanoid robots and the wide application of robotics to human environments.

Oussama Khatib is Professor of Computer Science and Mechanical Engineering at the Stanford Artificial Intelligence Laboratory. His research interests include human-centered robotics and haptic systems.