Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

A Whole-Body Control Framework for Humanoids Operating in Human Environments

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

***Abstract*— Tomorrow's humanoids will operate in human environments, where efficient manipulation and locomotion skills, and safe contact interactions will be critical design factors. We report here our recent efforts into these issues, materialized into a whole-body control framework. This framework integrates task-oriented dynamic control and control prioritization [14] allowing to control multiple task primitives while complying with physical and movement-related constraints. Prioritization establishes a hierarchy between control spaces, assigning top priority to constraint-handling tasks, while projecting operational tasks in the null space of the constraints, and controlling the posture within the residual redundancy. This hierarchy is directly integrated at the kinematic level, allowing the program to monitor behavior feasibility at runtime. In addition, prioritization allows us to characterize the dynamic behavior of the individual control primitives subject to the constraints, and to synthesize operational space controllers at multiple levels. To complete this framework, we have developed free-floating models of the humanoid and incorporate the associated dynamics and the effects of the resulting support contacts into the control hierarchy. As part of a long term collaboration with Honda, we are currently implementing this framework into the humanoid robot Asimo.**

***Index Terms*— Hierarchical control, whole-body behaviors, prioritization, free-floating dynamics, supporting contacts.**

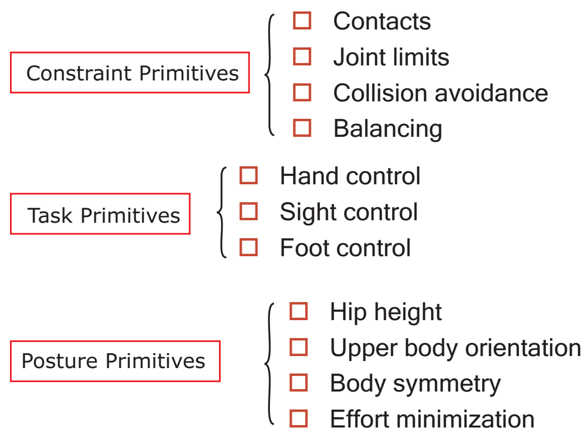

Fig. 1. **Control primitives:** Complex behaviors are formed by combining multiple control primitives. These primitives are divided into three distinct categories: (1) constraints, representing physical and movement-related restrictions, (2) operational tasks, corresponding to precision behaviors such as manipulation, vision, and locomotion, and (3) postures to control the residual redundancy.

I. Introduction

In recent years, a large number of humanoid robots have emerged from private corporations and academic institutions [2]. Despite their popularity, these mechanical marvels have not achieved their full potential in terms of responsive behaviors and physical interactions with their environment. A major limitation comes from employing inverse kinematic methods, which are unable to control task impedances, a key element for whole-body contact interactions. Another limitation comes from the lack of techniques that can monitor behavior feasibility and solve scenarios where the global behavior is infeasible under the acting constraints. In response to these limitations, we recently proposed a prioritized control approach [14] that allows us to project operational tasks into the constraint null-space and establish further priorities among the tasks themselves. This hierarchical approach prevents lower priority tasks from interfering with higher priority tasks, and provides the means to monitor behavior feasibility at runtime.

This hierarchy is used to prevent operational and postural tasks from violating the acting constraints in the robot's body. From a feasibility perspective, we can view prioritization as a kinematic projection of lower-priority Jacobians into the null-space of higher priority tasks. If any operational task or posture is infeasible under the acting constraints, these constrained Jacobians become singular. This a unique feature of our approach, that allows the program to monitor behavior feasibility at runtime and provides the support for applying singular avoidance strategies.

In contrast to our control approach, previous methods have addressed constraints as secondary control objectives [10], [12], and therefore lack the ability to prevent violations of the acting constraints in conflicting scenarios. In addition, most of the previous methods that deal with constraints are based

0-7803-9505-0/06/$20.00 ©2006 IEEE

on inverse kinematic resolution algorithms, which conceal the interaction between motion and force of the task points and postural structures being controlled.

For safety, and to allow whole-body contact interactions, we have developed extensions to the operational space formulation [4], that provide decoupled dynamic behaviors, and force-level control for all control primitives.

This paper is aimed to describe all the components of a behavior-oriented whole-body control framework, based on task prioritization. To do that in Section II we complete our previous work [14] on prioritization by establishing three distinct control categories: constraints, operational tasks, and postures. We also introduce a behavior-oriented approach to movement, where behaviors are constructed as aggregations of independent control primitives (see Fig. 1). In addition we explain how to use force control strategies for online interactions with physical agents. In Section III we describe how to deal with physical constraints and what are the control transitions involved. Finally, in Section IV we introduce extensions for dealing with free-floating DOF and supporting contacts.

Based on this framework we have built a task-oriented behavior architecture which is now being implemented into the robot Asimo. For modularity reasons, control primitives are represented as independent control abstractions, providing kinematic and dynamic information, as well as the parameters needed for control. This architecture, allows a supervisory system to synthesize new behaviors on-demand, while complying autonomously with environmental and body constraints. It also provides task and posture impedance control, allowing to perform accurate manipulation and locomotion tasks involving contact interactions, while making the posture compliant (soft).

II. A Whole-Body Control Framework

A. Control Primitives

Our control approach, extends the operational space formulation to allow the robot to simultaneously accomplish multiple low-level tasks (a.k.a control primitives) as part of a whole-body behavior. Each control primitive is responsible for controlling the low-level behavior of a different body part or movement criteria. To build a prioritized control hierarchy, we categorize control primitives into three distinct types: constraint-handling tasks, operational tasks, and postures.

Constraint-handling tasks are primitives designed to deal with physical and movement constraints that could endanger the robot or the robot's physical environment. Operational tasks are low-level behaviors designed to provide manipulation and locomotion skills involving compliant contact interactions. Operational tasks usually involve the control of the robot's hands, head, and feet, but are also used to operate other parts of the robot's body such as the global center of gravity, the hip, or any other point in the robot's body.

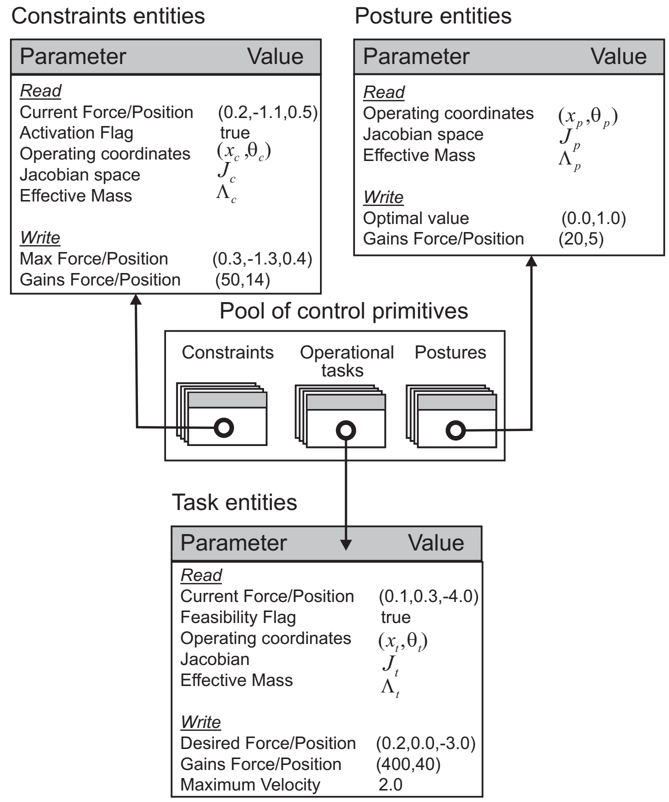

Fig. 2. **Control entities:** A behavior-based architectures has been implemented, where control primitives are represented as independent abstractions. They provide access to the kinematics and dynamics of the primitives, and the control parameters that define the feedback control laws that govern their movement.

Postures are primitives designed to control the additional redundancy and are used for many different purposes such as to mimic human-like postures or to minimize torque effort [6]. We combine together several control primitives to form coherent whole-body movement behaviors. For example, for a behavior involving drilling a hole on an object, we provide the control of the drilling tool, head orientation, and feet position, while controlling automatically body balance, feet moments, body constraints, support contacts, hip height, and torso orientation. For modularity reasons, the associated control primitives are designed as independent goal-based entities (see Fig. 2), and linked directly to the relevant sensory inputs, while receiving goal parameters and trajectories from the program.

B. Whole-Body Kinematics

To represent low-level behaviors at the whole-body level, our control framework uses a free-floating model of the robot, where 6 virtual unactuated DOF describe the dynamics of the free-floating base (the hip link usually). In this context, task and posture kinematics are defined with respect to the origin of the free-floating model:

$$x_{\text{task}} = T(x_{\text{base}}, q), \qquad \theta_{\text{posture}} = T(x_{\text{base}}, q). \quad (1)$$

Here x_{base} is a 6×1 vector of virtual unactuated joints, and q is the vector of actuated (robot) joints. These kinematic descriptions allow the robot to virtually involve any combination of joints (depending on the controlled tasks) to achieve the commanded behaviors. In contrast, traditional control approaches use local Jacobians defined with respect to the reference frame at the origin of the limbs. Although these traditional approaches prevent conflicting scenarios between the low-level behaviors, they are unable to coordinate joint resources at the whole-body level.

C. Control Hierarchy

We review the control hierarchy that we proposed in [14] and we introduce three distinct control categories: constraint-handling tasks, operational tasks, and postures. A control hierarchy between these categories is used to monitor and deal with conflicting scenarios between the acting constraints and the commanded behaviors. Previous control methods [10], [12] have dealt with constraints as secondary tasks, being unable to prevent constraint violations at all times.

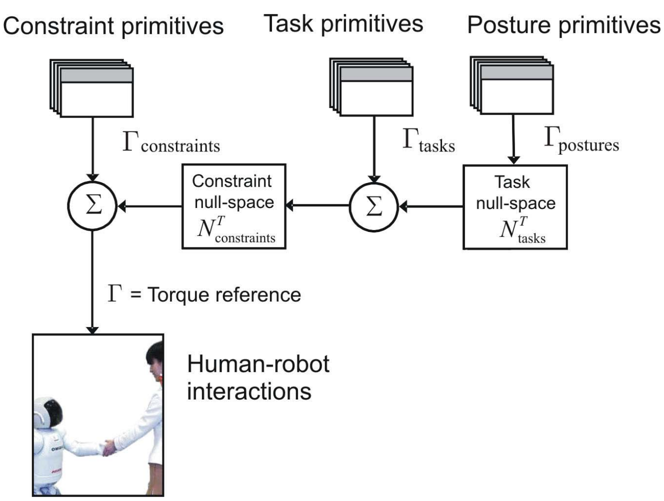

Fig. 3. A control hierarchy is established using null-space projections.

The control hierarchy that we proposed is a departure from these methods for constraints are accounted for as primary tasks (for they can never be violated), while operational tasks are projected into the constraint null-space (see Fig. 3). To use the additional redundancy, postures are projected into both the constrain null-space and the task null-space. This hierarchy is embodied in the following torque-level equation:

$$\Gamma = \Gamma_{\text{constraints}} + N^T_{\text{constraints}}\big(\Gamma_{\text{tasks}} + N^T_{\text{tasks}}\Gamma_{\text{postures}}\big). \quad (2)$$

Here Γ is a vector of joint torques, the subscripted Γ's are control vectors for the different categories in the hierarchy, and the N^T matrices correspond to dynamically-consistent null-spaces [14] of higher priority control levels.

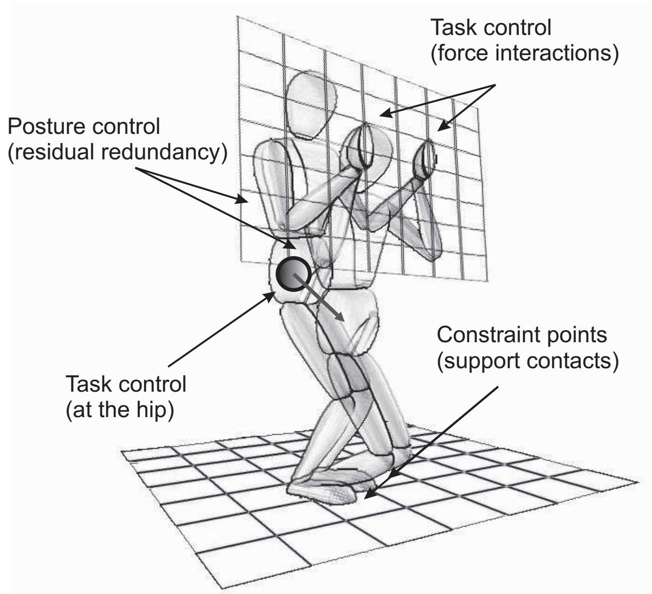

Fig. 4. Illustration of contact constraints

On the control side, we provide operational space control policies for all priority levels in the hierarchy. Operational space control was first designed to provide task-based force control. As such, it projects task space forces F into joint torques Γ through the task forward kinematics J, i.e.

$$\Gamma = J^T F. \quad (3)$$

To adapt operational space control for our control hierarchy we first define a new class of virtual Jacobians formed by projecting the unconstrained task-space Jacobians into the constraint null-space, i.e.

$$\boxed{J_{t|c} = J_{\text{tasks}} N_{\text{constraints}}.} \quad (4)$$

Here $J_{t|c}$ is the prioritized (constrained) task-space Jacobian, and the subscript $t|c$ means that the task is operating within the null-space of the constrained points. As for the posture, we define the following prioritized Jacobian:

$$J_{p|t|c} = J_{\text{postures}} N_{\text{tasks}} N_{\text{constraint}}, \quad (5)$$

where $J_{p|t|c}$ is the posture Jacobian operating in the residual redundancy. The subscript $p|t|c$ means that the posture operates within the null-spaces of both the constrained points and the operational tasks. These null-space matrices can be computed as described in [14]. We note here that $J_{\text{constraints}}$, J_{tasks}, and J_{postures} can be used to combine multiple primitives into a single control category, and therefore controlled with the same priority. For example, for manipulation tasks J_{tasks} would consists of

$$J_{\text{tasks}} = \Big(J_{\text{righHand}}; J_{\text{leftHand}} \Big). \quad (6)$$

Under these extensions, Equation (2) can be further expressed in operational space form as

$$\Gamma = \left(J_{\text{constraints}}^{T} F_{\text{constraints}}\right) + \left(J_{t|c}^{T} F_{t|c}\right) + \left(J_{p|t|c}^{T} F_{p|t|c}\right).$$

Here the F's are force vectors that control the individual control primitives. In particular, $F_{t|c}$ is used to control the desired impedance, forces or trajectories of the tasks, $F_{\text{constraints}}$ is used to maintain a desired distance, position, or force at the constrained points, and F_{postures} is used to optimize some desired criteria while providing soft interactions.

D. Hybrid Control and Impedance Control

To characterize the dynamic behavior of the task within our hierarchy we define a prioritized task inertia matrix:

$$\boxed{\Lambda_{t|c} = (J_{t|c} A^{-1} J_{t|c}^{T})^{-1},} \quad (7)$$

where A is the joint space inertia matrix, and $J_{t|c}$ is the prioritized task Jacobian that was described in Equation (4). Notice that the constraints are directly integrated into $\Lambda_{t|c}$, and in turn this inertia matrix captures the dynamic effect of the constraints into the task (see Fig. 5). At the constraint level, we can also obtain an inertia matrix defined by

$$\Lambda_{\text{constraints}} = (J_{\text{constraints}}^{T} A^{-1} J_{\text{constraints}})^{-1}. \quad (8)$$

This matrix does not need to be further constrained since constraints take already the highest priority.

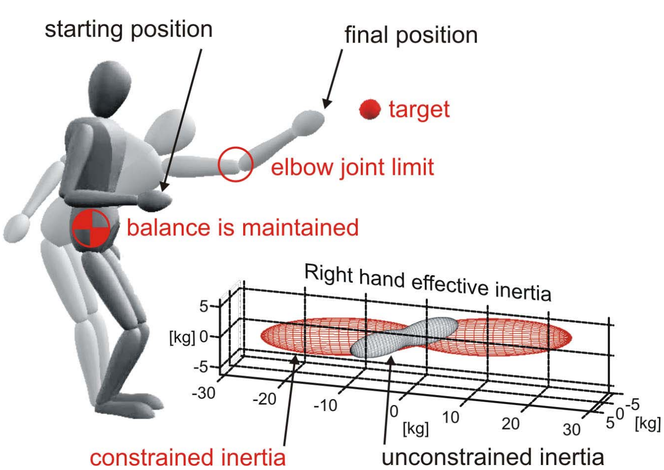

Fig. 5. **Effective inertia:** This overlayed sequence depicts a simulated humanoid reaching a target, while balancing and complying with joint-limit constraints. The posture is also controlled to keep body symmetry. The large ellipsoid depicts the effective inertia at the hand in the presence of balancing and joint-limit constraints, $\Lambda_{t|c}$. The smaller ellipsoid represents the task inertia in the absence of constraints Λ_{task}.

We derive the dynamic behavior of constrained tasks by multiplying the joint space dynamics by the dynamically consistent generalize inverse of the constrained jacobian, $\overline{J}_{t|c}^{T}$,

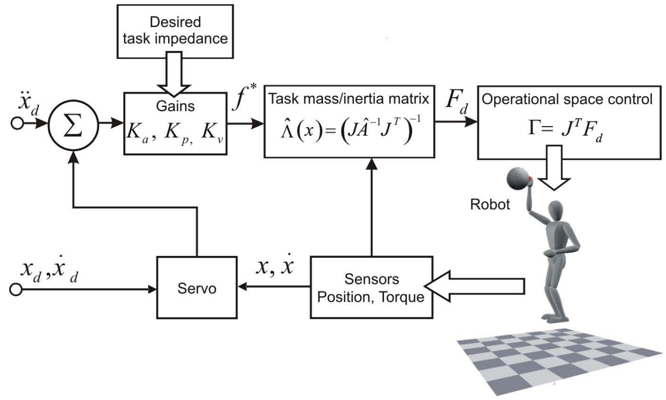

Fig. 6. **Impedance control layer:** Each control primitives in our framework implements a local force control strategy.

i.e.

$$\overline{J}_{t|c}^{T}\left(A\ddot{q} + b + g + J_{\text{int}}^{T} f_{\text{int}} = \Gamma\right) \Longrightarrow$$
$$\Lambda_{t|c}\ddot{x}_{\text{task}} + \mu_{t|c} + p_{t|c} = \overline{J}_{t|c}^{T}\Gamma - \overline{J}_{t|c}^{T} J_{\text{int}}^{T} f_{\text{int}}. \quad (9)$$

Here, b and g are the Coriolis/Centrifugal and gravity joint-level torque components, $\mu_{t|c}$ and $p_{t|c}$ are the Coriolis/centrifugal and gravity task-level force components, f_{int} is a vector of interactive forces exert by a physical agent, and J_{int} is the Jacobian of the point where these forces are applied to. For this example, the external forces f_{int} are not treated as constraints; instead they are treated as perturbations. Also, for simplicity, we have ignored the torques corresponding to other tasks involved in the whole-body control of the robot. Note that in this section we are not using a free-floating description of the robot. Extensions of our framework for free-floating models will be discussed in Section IV.

To compensate for nonlinear dynamics, we use the following task control torque:

$$\Gamma_{\text{task}} = J_{t|c}^{T}\left(\hat{\Lambda}_{t|c} F_{\text{task}}^{*} + \hat{\mu}_{t|c} + \hat{p}_{t|c}\right) + J_{t|c}^{T}\overline{J}_{t|c}^{T} J_{\text{int}}^{T}\hat{f}_{\text{int}}. \quad (10)$$

Here the force-level input F_{task}^{*} is a feedback control law designed to achieve the task goal, and the quantities capped with a $\hat{}$ are estimated values. If the task is feasible under the acting constraints, this control will result in the decoupled behavior, $\ddot{x}_{\text{task}} = F_{\text{task}}^{*}$.

As an example of the capabilities of this control, we describe an impedance control [3] solution:

$$F_{\text{task}}^{*} = a_{\text{des}} - M_{\text{des}}^{-1} D_{\text{des}}(\dot{x}_{\text{task}} - v_{\text{des}})$$
$$- M_{\text{des}}^{-1} K_{\text{des}}(x_{\text{task}} - x_{\text{des}}) + M_{\text{des}}^{-1}\hat{f}_{\text{int}}, \quad (11)$$

where a_{des}, v_{des}, and x_{des} are task accelerations, velocities, and positions respectively obtained from a desired tracking law, and the desired apparent inertia M_{des}, damping D_{des}, and stiffness K_{des} matrices are given by the programmer (see

Fig. 6). This control will result in the targetted impedance behavior

$$M_{\rm des}(\ddot{x}_{\rm task} - a_{\rm des}) + D_{\rm des}(\dot{x}_{\rm task} - v_{\rm des}) + K_{\rm des}(x_{\rm task} - x_{\rm des}) = \hat{f}_{\rm int}. \quad (12)$$

Another example is an implementation of a hybrid position/force controller in task-space. Without going into much detail we propose a hybrid control solution given by

$$F^*_{\rm task} = \Omega_m F^*_m + \Omega_f F^*_f. \quad (13)$$

Here Ω_m and Ω_f are selection matrices for the free motion and contact spaces, and F^*_m and F^*_f are the associated feedback control laws (see the methods described in [5]).

One of the main advantages of describing the task kinematics in terms of $J_{t|c}$ is that it allows the program to monitor task feasibility at runtime. In a situation where the task cannot be accomplished without violating the acting constraints, $J_{t|c}$ becomes singular. This is a unique characteristic of prioritization, as $J_{\rm task}$ alone does not reflect the effect of the acting constraints and would not be necessarily singular in the same scenario. One possibility to solve infeasible scenarios is to only control the feasible directions. A procedure to do that was described in [14], and consists on discarding the singular directions. The residual null-space could be used to avoid the singularity (see [4]).

In our framework, the control of the constrained points and the control of the posture structures are also based on similar force-level controllers. For example, for the control of an obstacle avoidance constraint we use the feedback control law

$$F_{\rm constraint} = \hat{\Lambda}_{\rm body}(-K_p(d_{\rm body} - d_{\rm des}) - k_v \dot{x}_{\rm body}) + \hat{\mu}_{\rm body} + \hat{p}_{\rm body}, \quad (14)$$

where $\hat{\Lambda}_{\rm body}$ is an estimate of the inertia of the point in the body closest to the constraining object, $d_{\rm body}$ is the distance to this object, $d_{\rm des}$ is the distance that we want to maintain, and $x_{\rm body}$ is the velocity of the body at the constraint point. The feedback law used here is obtained from the gradient of the potential field $V = ||d_{\rm body} - d_{\rm des}||^2$ (for potential fields, see [4]). The details about the dynamic behavior and control of posture tasks will be discussed in a future paper.

Having discussed the fundamental principles of prioritized control, let us proceed with a more detailed discussion on the constraints we account for and the particularities of their control.

III. Dealing with Non-Contact Constraints

A great deal of work in obstacle avoidance can be found in the robotics literature [1], [4], [12]. On the other hand, much fewer techniques to deal with joint limit constraints can be found besides the first gradient projection method [10]. Work on self-collision avoidance, the third class of non-contact constraints that we address here, is nearly absent, being the only reference that we were able to find in the context of motion planning [11]; however, several algorithms for fast detection of self collisions are available [8], [9] and can be used to halt the robot movement if two links come close.

We describe here methods to deal with several important constraints that do not involve physical contact with the robot's body, i.e. avoidance of joint-limits, near-body obstacles, and self-collisions. These constraints appear when some of the commanded behaviors are not pre-planned. For example, if the robot's hand is teleoperated the arm and hip can easily reach their joint limits. Another example occurs in dynamic environments, where people may quickly come close to the robot's body, leaving no time for replanning motion. In this context, we propose to use our reactive techniques to deal with obstacles and contact points. Note, that it would also be possible to deal with joint-limits, and near-body obstacles as contact tasks. However in this paper we will only address them as avoidance tasks. In the next section, we will propose extensions to our framework for integrating contact constraints arising from the ground reactions at the feet.

A. Joint-Limits

Because we synthesize whole-body behaviors on-demand by aggregating multiple control primitives, and because we define these primitives with respect to whole-body kinematics, we cannot predict the resulting joint movements. To prevent violating joint limits during motion, we describe here a robust method to deal safely with these constraints without interrupting the robot's global task. Although joint-limit avoidance could have been addressed as a contact task, our strategy here is to avoid hitting the actual hard limits.

One of the reasons why joint limits have not received much attention, is because manipulation tasks have been traditionally defined with respect to local kinematics, allowing instead to work with workspace limits. Also many control strategies are based on following prerecorded movement trajectories. In contrast, our approach is based on direct monitoring and control of joint limits, while ensuring the completion of the operational task. This strategy provides greater movement flexibility, for whole-body tasks can be accomplished without a priori knowledge of the joint trajectories. In Fig. 7 we illustrate a control scenario with joint-limit avoidance, for a task of moving the tool towards a target position. In this example, the elbow joint limit is reached first and as a result we activate a constraint-avoidance control. For simplicity, we only consider the operational task and the control of the constraint, while other tasks such as balancing or posture control are not considered. During the transition phase from unconstrained to constrained control the following control

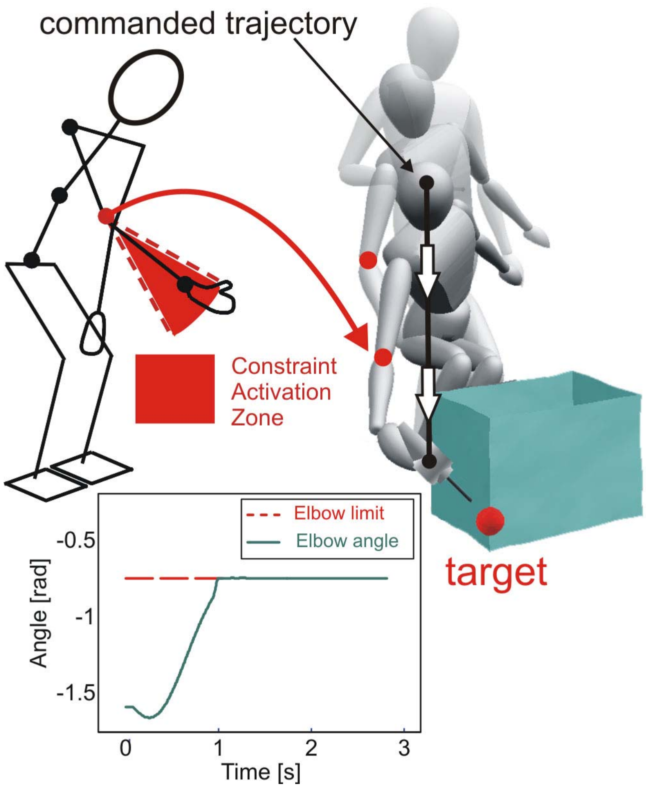

Fig. 7. **Joint-limits:** In this sequence the hand-tool is commanded to move towards an unreachable target. Joint-limit avoidance is activated to prevent penetrating several joint-limit safety bounds during motion, including the right elbow, the hip, and the knees. The hand task is achieved efficiently as long as it is feasible.

laws are used:

$$\Gamma = \left(J^T_{\text{task}} F_{\text{task}}\right) \Longrightarrow \text{"upon activation"} \Longrightarrow \quad (15)$$

$$\Gamma = \left(J^T_{\text{jointLimits}} F_{\text{jointLimits}}\right) + \left(J^T_{t|c} F_{t|c}\right). \quad (16)$$

Here the upper equation corresponds to the control of the robot's hand before the elbow joint limit is reached, and the lower equation corresponds to the control upon activation. Notice that $J_{t|c}$ is equal to $J_{\text{task}} N_{\text{jointLimits}}$ as discussed in Section II, and the Jacobian J_{task} is defined with respect to the free-floating global reference (not shown in the image). The force $F_{\text{jointLimits}}$ is used to maintain a safety distance to the hard limit, while $F_{t|c}$ is used to provide force control of the robot's hand within the constraint null-space. The resulting behavior is a smooth trajectory towards the target while the elbow joint-limit is maintained at all times.

B. Near-Body Obstacle Avoidance

Obstacles can be avoided in different ways depending on the control scenario. For locomotion tasks and coarse point-to-point reaching tasks, obstacles should be dealt with motion planners such as [7]. However, for short range manipulation in the presence of near-body obstacles, or when unexpected obstacles approach quickly the robot's body, we propose to rely on reactive techniques.

Near-body obstacles can strongly shape the robot's motion. By dealing with them at the highest priority level, the controller can detect task feasibility before collisions occur, and consequently a supervisory system can modify or halt the commanded trajectories. When an obstacle comes close enough to any point of the robot's body (provided that the robot can estimate the distance to the obstacle), the program activates an obstacle avoidance task, and we project the operational task into the constraint null-space, i.e.

$$\Gamma = \left(J^T_{\text{obstacle}} F_{\text{obstacle}}\right) + \left(J^T_{t|c} F_{t|c}\right). \quad (17)$$

Here J_{obstacle} is the Jacobian of the closest point to the obstacle. For simplicity, in the previous example we have ignored postures and other tasks necessary for whole-body control. In Fig. 8 a simulated scenario under near-body obstacles is shown. Using interactive tools, obstacles are approached to the robot's body causing the robot's posture to change while maintaining the hand trajectory control. A proper feedback law implemented through F_{obstacle} will ensure that the robot maintains a minimum safety clearance with respect to the obstacle.

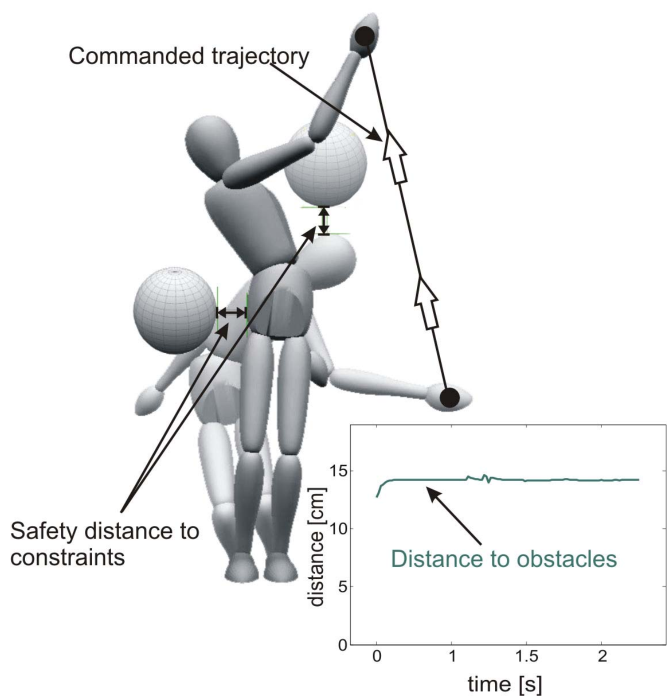

Fig. 8. **Obstacle avoidance:** These overlayed images depict obstacle avoidance while the robot's right hand tracks a given trajectory. The control of the hand is projected into the constraint null-space, eliminating possible conflicts between the task and the obstacle. An active controller is used to maintain a desired safety distance to the obstacle.

Multiple constraints can simultaneously appear. To account for all acting constraints, we propose to concatenate individual constraint-handling tasks into a single control primitive. This can be done by appending together multiple constraint Jacobians as well as the desired feedback control laws; i.e

$$J_{\text{constraints}} = \left(J_{\text{jointLimits}}; J_{\text{obstacles}} \right), \tag{18}$$

$$F_{\text{constraints}} = \left(F_{\text{jointLimits}}; F_{\text{obstacles}} \right). \tag{19}$$

Here the semicolon separation means that both Jacobians are appended vertically. At this stage it is still unclear wether there should be a prioritized ordering between the constraints themselves. We will leave the study of this problem for a future paper.

In humanoids, to guarantee stability, stand-up behaviors are based on the control of the center of gravity. We are currently researching wether balancing tasks should be accounted for as constraints or as operational tasks. An intermediate solution would be to create an additional priority level for balancing control, i.e.

$$\Gamma = \Gamma_{\text{constraints}} + N^T_{\text{constraints}}(\Gamma_{\text{balancing}} + N^T_{\text{balancing}}(\Gamma_{\text{tasks}} + N^T_{\text{tasks}}\Gamma_{\text{postures}})) \tag{20}$$

To complete our framework let us incorporate models that account for free-floating degrees of freedom (DOF) and supporting contacts.

IV. Accounting for free-floating DOF and Supporting Contacts

As we mentioned earlier, we model humanoids as free-floating systems with 6 unactuated degrees of freedom attached to the base link. We use a virtual reference frame at the origin of these unactuated DOF to describe whole-body kinematics and dynamics. Gravity, pushes down the robot's body onto the floor, grounding the feet into the support contacts (see Fig. 9). These supports need to be indirectly used to control the body balance, and the other low-level behaviors. The global dynamic behavior of the free-floating system can be described as

$$A \begin{pmatrix} \ddot{x}_{\text{base}} \\ \ddot{q} \end{pmatrix} + b + g + J^T_{\text{support}} F_{\text{support}} = \begin{pmatrix} 0 \\ \Gamma_{\text{act}} \end{pmatrix}, \tag{21}$$

where the term $J^T_{\text{support}} F_{\text{support}}$ is due to the reaction forces at the supporting points, and Γ_{act} is the vector of torque actuation.

Considering a model where the linear accelerations at the support points are zero ($\ddot{x}_{\text{support}} = 0$), which implicitly assumes that the floor is very stiff and its friction very high, we can further develop Equation (22) into

$$A \begin{pmatrix} \ddot{x}_{\text{base}} \\ \ddot{q} \end{pmatrix} + b + g = N^T_{\text{support}} S^T_{\text{act}} \Gamma_{\text{act}}. \tag{22}$$

Here N^T_{support} is the dynamically-consistent null-space of the Jacobian at the support points and $S_{\text{act}} = [0 \quad I]$ is a selection

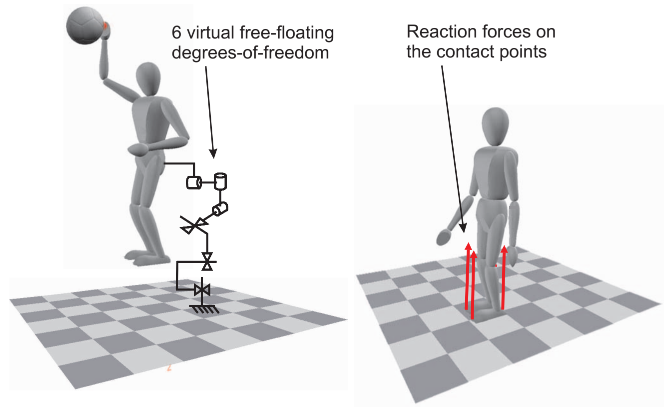

Fig. 9. **Free-floating DOF and reaction forces:** Humanoids can be modeled as free-floating systems. As such we introduce six unactuated DOF attached to the base link (the hip). Because of gravity, multiple body parts are in contact with the ground, creating a support contact area. Our framework accounts for free-floating dynamics and supporting contacts.

matrix with zeros in the unactuated DOF, used to simplify notation.

To control constrained tasks in operational space subject to free-floating dynamics (i.e. conservation of angular momentum), let us first derived the free-floating Jacobian of the task [13]:

$$J_{\text{taskFF}} = J_{\text{task}} \big(\overline{S}_{\text{act}} S_{\text{act}} \big). \tag{23}$$

Here $\overline{S}_{\text{act}}$ is the dynamically-consistent generalized inverse of S_{act}. We then prioritized this Jacobian using the support null-space:

$$J_t = J_{\text{taskFF}} N_{\text{support}} = J_{\text{task}} \big(\overline{S}_{\text{act}} S_{\text{act}} \big) N_{\text{support}}, \tag{24}$$

where $J_{\text{task}} = \partial x_{\text{task}}(x_{\text{base}}, q)/\partial q$ is the whole-body Jacobian of the task. In addition, to fulfill the control hierarchy given in Equation (2) we need to further incorporate the null-spaces corresponding to the non-contact constraints, further transforming the previous equation into

$$\boxed{J_{t|c} = J_{\text{task}} N_{\text{constraints}} \big(\overline{S}_{\text{act}} S_{\text{act}} \big) N_{\text{support}}.} \tag{25}$$

This Jacobian is in essence an extension of Equation (4) incorporating the dynamic effect due to free-floating DOF and the support contacts. We omit the expression of the prioritized posture Jacobian $J_{p|t|c}$, noting that it can be obtained by adding the free-floating term and support null-space identically than for $J_{t|c}$. The task dynamics can be obtained by projecting Equation (22) with the dynamically consistent generalized inverse of $J^T_{t|c}$. The details for the dynamic behavior of this free-floating system under supporting contacts and its whole-body control will be discussed in the journal version of this paper.

In Fig. 10 we show a simulated experiment of interactive control of the robot's right hand in the presence of a wall. To accomplish this behavior we control the body center

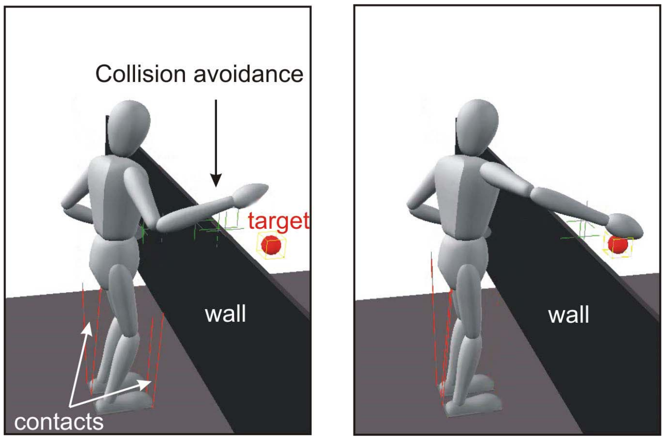

Fig. 10. **Reaching beyond a wall:** This sequence of images taken from a realtime simulation, depicts a reactive behavior for reaching a target beyond a wall. Obstacle avoidance is activated during motion without affecting the task trajectory. Contact forces at the feet and free-floating dynamics are accounted for at the kinematic and dynamic levels.

of gravity, the right hand position, the hip height, the feet compliance, the body posture, a wall avoidance task, and the feet compliance as part of a single whole-body behavior. The robot's hand is controlled interactively while the constraints and postures are automatically accounted for by the control hierarchy. Further details on this experiment will be discussed in a future paper.

V. Summary and Discussion

Aiming to assist the human, we have developed a behavior-based whole-body control framework that endows humanoids with the ability to accomplish precision tasks and optimal postures while autonomously handling multiple constraints. The key aspects of this framework are

1) *Implements reactive techniques (potential fields) for handling constraints*
2) *Categorizes task primitives into priority levels, and assigns top priority to constraints*
3) *Implements operational-space control at all levels and monitors task feasibility at runtime*
4) *Accounts for free-floating DOF and support contacts*
5) *Builds new behaviors as aggregations of multiple control primitives.*

As part of a long-term collaboration with Honda, our lab is fully engaged in the implementation of this framework into the humanoid robot Asimo. In this process, uncertainty is a major concern. If joint-limits, self-collisions, and balancing constraints can be estimated with high accuracy, contacts and near-body obstacles are hard to locate. For robustness, we need to implement impedance control at all control levels in the hierarchy. At the same time, we are developing body-image acquisition techniques that will enable Asimo to self-localize points of contact in any place in its entire body. At this time, it is too early to show results on the implementation into Asimo.

Prioritization opens up an new body of research. Tasks are not controlled anymore using free-space kinematics. Instead, we use prioritization. In this context, kinematic singularities can now appear due to the acting constraints. We exploit this new characteristic to interrupt or modify the task without violating the constraints.

Although the extensions for free-floating DOF and supporting contacts have been lightly presented, in the journal version of this paper we will present them in much greater detail.

References

[1] O. Brock, O. Khatib, and S. Viji. Task-consistent obstacle avoidance and motion behavior for mobile manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 388–393, Washingtion, USA, 2002.

[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] N. Hogan. Impedance control: An approach to manipulation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–24, 1985.

[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 for motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics Research*, 3(1):43–53, 1987.

[6] O. Khatib, J. Warren, V. De Sapio, and L. Sentis. Human-like motion from physiologically-based potential energies,. In *Advances in Robot Kinematics*, pages 149–163. Kluwer Academic Publishers, First edition 2004.

[7] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Motion planning for humanoid robots. In *Proceedings of the International Symposium of Robotics Research*, Siena, Italy, October 2003.

[8] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshil, M. Inabal, and H. Inouel. Self-collision detection and prevention for humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2265–2270, Washington, USA, May 2002.

[9] 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.

[10] A. Liegois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7:868–871, 1977.

[11] 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.

[12] 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.

[13] L. Sentis and O. Khatib. Control of free-floating humanoid robots through task prioritization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1730–1735, Barcelona, Spain, April 2005.

[14] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):505–518, December 2005.