Contact Consistent Control Framework for Humanoid Robots

Jaeheung Park and Oussama Khatib
Artificial Intelligence Laboratory
Stanford University
Stanford, CA 94305, USA
{park73, ok}@robotics.stanford.edu

***Abstract*— This paper presents a framework for the dynamical formulation and control of humanoid systems. In this framework unactuated virtual joints are used to describe the humanoid's configuration with respect to the inertial frame. The dynamics of the system are then formulated in a general manner that considers arbitrary contact with the environment. A control structure is implemented for both motion and contact forces that accounts for under-actuation due to the virtual joints. A strategy is also implemented to address transitions between different contact states. Simulation results are presented that demonstrate this overall framework for many behaviors such as standing, walking, jumping, and hand manipulation with walking.**

***Index Terms*— Humanoid robot, Under-actuated system, Contact, and Control**

I. Introduction

Substantial research related to humanoid robots has been conducted in the past two decades. This has led to the development of enhanced humanoid robots which include Honda's ASIMO, Sony's SDR, HRP, and KHR-2 [6], [5], [10]. These and other humanoid systems share three common characteristics: many degrees of freedom, under-actuation, and contact with environment. These characteristics have motivated different approaches for controlling humanoid robots compared with fixed base robots. The control strategies employed in these systems have been limited to specialized behaviors for desired tasks - especially walking [7], [14], [11], [12]. Consequently, a general control methodology is sought that integrates various whole body behaviors into a single dynamically compensated control structure.

By formulating the dynamics of the system in a manner that accounts for contact with the environment a generic control structure can be composed which is consistent with the contact state. In this paper virtual joints are used to describe the robot in any contact condition including free space. Then, the dynamics of a humanoid in contact with the environment is obtained by treating the robot in contact as constrained to the environment. This dynamic equation includes the environment so that the controller based upon this dynamic equation can utilize the contact forces. This approach is especially effective for humanoid systems because they do not have fixed base. Without accounting for the contact forces, these robotic systems cannot be controlled properly, e.g. the center of mass of the system. In this approach, rather than specifying the contact forces, the necessary contact forces will be produced as the motion of the robot is controlled. The quantities to be controlled, such as the center of mass of the system and the position of certain links, form operational space coordinates. The operational space coordinates are controlled using the dynamics projected into the operational space coordinates from the constrained dynamics of the system [18], [16].

II. Dynamics of the whole system

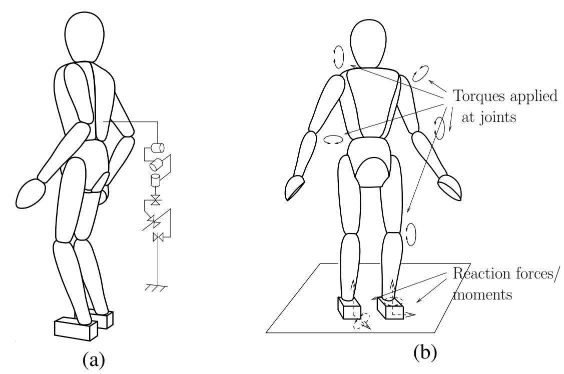


Fig. 1. **Robot representation**. (a) virtual joints (b) joint torques and reaction forces

Given a humanoid with k joints, the total system has $n = k+6$ degrees of freedom since we must account for the rigid body motion relative to an inertial reference frame. Without loss of generality, we can describe this rigid body motion using three revolute joints and three prismatic joints assigned to any link of the system. We will refer to these as virtual joints (Fig. 1 (a)). The dynamic equations for the robot in free space are then described by

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

where q is the $n \times 1$ vector of joint angles and Γ is the $n \times 1$ torque vector to the corresponding joints. The term $A(q)$ is the $n \times n$ joint space inertia matrix, $b(q,\dot{q})$ is the $n \times 1$ vector of Coriolis and centrifugal terms, and $g(q)$ is the $n \times 1$ vector of gravity terms.

When the robot is in contact, the contact forces (Fig. 1 (b)) should be included in the dynamic equations. This yields

$$A(q)\ddot{q} + b(q, \dot{q}) + g(q) + J_c^T f_c = \Gamma, \tag{2}$$

where f_c is the vector of contact forces and moments and J_c is the Jacobian for the contact positions and orientations.

A. *Contact with environment*

Rigid body contact with the environment can be categorized as point, line, and plane contact. The contact forces/moments consist of contact normal forces/moments and tangential friction forces/moments. The contact on the robot and the environment can be considered constrained when the contact normal forces/moments between the robot and the environment are within certain limits and the tangential forces/moments due to friction do not exceed static friction values. That is, the contact point, line or plane is constrained to have the same position, velocity and acceleration as those of the contacted environment. These conditions or boundaries to maintain the contacts will be referred as *contact conditions*.

The specific contact conditions are shown in Figure 2. In Figure 2 (a), the normal force from the robot to the environment in the normal direction must be negative and the magnitude of the tangential force must be less than $\mu_{static}|F_z|$ to maintain the contact. In addition to the conditions for the point contact case, the line contact case requires that the magnitude of the applied normal moment in the x direction, M_x, be less than $|F_z| \times \frac{l_y}{2}$ and that the magnitude of the applied moment in the z direction, M_z, be less than $\mu'_{static}|F_z|$ (Figure 2 (b)). In addition to the conditions for the line contact case, the plane contact case requires that the magnitude of the applied normal moment in the y direction, M_y, be less than $|F_z| \times \frac{l_x}{2}$ (Figure 2 (c)).

To describe the contact dynamics, the Jacobian corresponding to the contact condition must be defined. For point contact, the contact Jacobian is defined as the Jacobian of the position of the contact point. In this case 3 degrees of freedom are constrained. For line contact, the contact Jacobian is defined as the Jacobian of the position/orientation of the geometrical center of the contact line, excluding the orientation about the contact line. In this case 5 degrees of freedom are constrained. For plane contact, the contact Jacobian is defined as the Jacobian of the position/orientation of the geometrical center of the contact plane. In this case 6 degrees of freedom are constrained.

The contact Jacobian, J_c, is defined as

$$\dot{x}_c = J_c \dot{q}. \tag{3}$$

Using the contact Jacobian, the dynamics of the system (2) is projected into the contact space

$$\Lambda_c \ddot{x}_c + \mu_c + p_c + f_c = \bar{J}_c^T \Gamma, \tag{4}$$

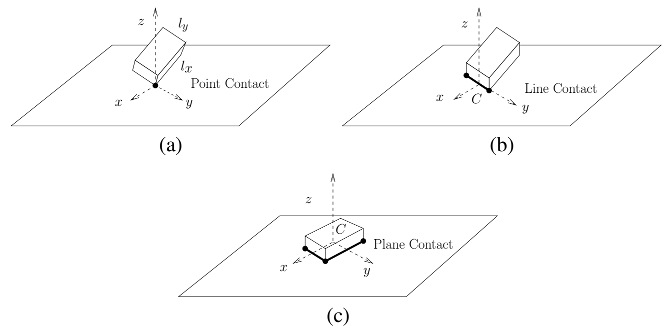


Fig. 2. **Contact Conditions**. (a) Point contact can be maintained if $F_z < 0$, and $\sqrt{F_x^2 + F_y^2} < \mu_s F_z$ where $f_c = [F_x \; F_y \; F_z]^T$. (b) Line contact can be maintained if $|M_x| < |F_z|\frac{l_y}{2}$, and $|M_z| < |F_z|\mu'_s$, additional to the conditions in (a). $f_c = [F_x \; F_y \; F_z \; M_x \; M_z]^T$. (c) Plane contact can be maintained if $|M_y| < |F_z|\frac{l_x}{2}$ in addition to the conditions in (b). $f_c = [F_x \; F_y \; F_z \; M_x \; M_y \; M_z]^T$. μ_s and μ'_s are the static friction coefficient for forces and moments, respectively.

where

$$\begin{aligned} \Lambda_c &= (J_c A^{-1} J_c^T)^{-1} && (5)\\ \mu_c &= \Lambda_c \{J_c A^{-1} b(q, \dot{q}) - \dot{J}_c \dot{q}\} && (6)\\ p_c &= \Lambda_c J_c A^{-1} g(q) && (7)\\ \bar{J}_c^T &= \Lambda_c J_c A^{-1}. && (8) \end{aligned}$$

The matrix Λ_c is the inertia matrix of the contact and $\bar{J}_c$ is the dynamically consistent inverse of J_c. The term μ_c is the projection of the Coriolis/centrifugal forces at the contact, and p_c is the projection of the gravity forces at the contact.

B. *Constrained dynamics of the system*

Equation (2) describes the robot dynamics with the external forces, f_c. However, to be able to utilize the contact forces, they have to be expressed in terms of the commanding torques and the dynamic parameters. This is possible in special cases when the dynamics of the contact environment is known. One of these cases is when the robot is in contact with the *ground* (e.g., standing, walking, or running). In this case the dynamic equations for the system in contact can be obtained.

While the contact conditions are met and the environment is stationary and rigid (e.g., the ground), we have $\ddot{x}_c = 0$ and $\dot{x}_c = 0$. Therefore,

$$f_c = \bar{J}_c^T \Gamma - \mu_c - p_c. \tag{9}$$

Now, the equations of motion of the whole system, i.e. the dynamics of the robot and environment, can be written by substituting the expression for f_c from Equation (9) into Equation (2).

$$A(q)\ddot{q} + b(q, \dot{q}) + g(q) + h_c(q, \dot{q}) = (I - P_c)\Gamma, \tag{10}$$

where

$$
\begin{aligned}
P_c &= J_c^T \bar{J}_c^T && (11)\\
h_c &= -J_c^T(\mu_c + p_c). && (12)
\end{aligned}
$$

Note that the terms $h_c(q, \dot{q})$ and $-P_c\Gamma$ in Equation (10) are the effective torques due to the contact forces. The input torque to the system is not $(I - P_c)\Gamma$ but Γ.

Equation (9) and (10) completely describe the system in contact. Given Γ, q, and $\dot{q}$ we can compute $\ddot{q}$ and f_c. We refer to Equation (10) as the *constrained dynamic equation of motion*. By treating the contacts as constraints they are accounted for in the system dynamics. Thus, designing a control strategy based upon this equation of motion accounts for the contacts as well as the robot dynamics.

These equations can also be obtained by directly solving the equations of motion under the constraints of $\ddot{x}_c = J\ddot{q} + \dot{J}\dot{q} = 0$ [18]. Equation (9) and (10) are valid as long as Γ is chosen not to violate the contact conditions, which are determined by the friction coefficient and the geometry of the contacts as described in the previous sub-section *A. Contact with environment*.

C. Constrained dynamics of the operational space coordinate

The operational space coordinate is the coordinate to be controlled for desired behaviors. Given an operational space coordinate, x, the corresponding Jacobian is defined as

$$\dot{x} = J\dot{q}. \tag{13}$$

The dynamic equations of motion of the operational space coordinate, x, are then obtained from Equation (10).

$$\Lambda(q)\ddot{x} + \mu(q, \dot{q}) + p(q) = F \tag{14}$$

where

$$
\begin{aligned}
\Lambda(q) &= [JA^{-1}(I - P_c)J^T]^{-1} && (15)\\
\bar{J}^T &= \Lambda JA^{-1}(I - P_c) && (16)\\
\mu(q, \dot{q}) &= \bar{J}^T b(q, \dot{q}) - \Lambda\dot{J}\dot{q} + \Lambda JA^{-1}J_c^T\Lambda_c\dot{J}_c\dot{q} && (17)\\
p(q) &= \bar{J}^T g(q) && (18)
\end{aligned}
$$

Equation (14) describes the *constrained* dynamics of the operational space coordinate. That is, the contact is treated as a constraint while the dynamic equations account for the contact forces. The matrix $\Lambda(q)$ is the operational space inertia matrix. The term $\mu(q, \dot{q})$ is the operational space Coriolis/centrifugal force and $p(q)$ is the operational space gravity force.

III. CONTROL FRAMEWORK

Note that the dynamic equations (9) and (10) include virtual joints to completely describe the system. However, we need to compose a control torque only at the actuated joints. The actuated joint torques are denoted as the $k \times 1$ vector, Γ^k. By using a $k \times n$ selection matrix S_k we have

$$\Gamma = (S^k)^T\Gamma^k. \tag{19}$$

In the case that the virtual joints correspond to the first 6 joints,

$$S^k = [\begin{array}{cc} 0_{k\times 6} & I_{k\times k} \end{array}]. \tag{20}$$

Note that this selection matrix can also be used to include real un-actuated joints.

A. Task Control

The control torque for desired task of a humanoid system can be composed using the *constrained dynamic equation of motion* (10) within contact conditions. If the number of constraints is greater or equal to the number of un-actuated joints, the constrained motion of the system will be fully controllable.

The dynamic equation can be projected to any coordinate to be controlled, such as the center of mass, hip orientation, the position of a certain point, etc. When we deal with many control points or coordinates, these can be concatenated into a single coordinate vector, x. Alternatively, the separate control points can be handled using priorities in a recursive way[1], [13], [17], [4], [9].

Given x as the coordinate vector to be controlled the dynamics in this coordinate space is given by Equation (14). Therefore, for the desired acceleration, f^*, the necessary control force is

$$F = \Lambda f^* + \mu + p \tag{21}$$

This control force can be generated by the torque, $\Gamma = J^T F$, in the case of a fully actuated system. However, when virtual joints are employed this torque may not be applicable since no actuation is provided at those virtual joints.

Given the control torque, Γ, the force, F, is given by (Fig. 3)

$$F = \bar{J}^T\Gamma. \tag{22}$$

Using the selection matrix to exclude the un-actuated joints,

$$F = \bar{J}^T(S^k)^T\Gamma^k. \tag{23}$$

Now, Γ^k must be chosen to produce the desired control force, F. If we denote m as the number of DOF of x then there are three cases: $m > k$, $m = k$, and $m < k$. In the case of $m > k$, F can not be produced as desired since there is an actuator deficiency. When $m = k$, Γ^k is uniquely determined if $\bar{J}^T(S^k)^T$ is invertible.

The most common and interesting situation in a high DOF system like a humanoid system is the case where $m < k$. In this case there is an infinite number of solutions of Γ^k that can achieve the desired force F. An intuitive way to resolve the redundancy is to minimize motion. That is, by minimizing

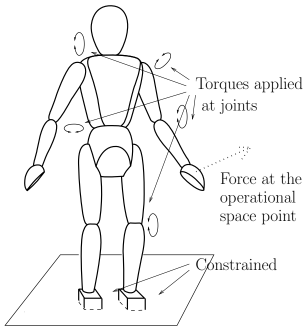


Fig. 3. **Representation of robot in contact as a constrained system**

the *acceleration energy* of the system, no unnecessary null-space motion will be produced. The *acceleration energy* is defined [2] as

$$E_a = \frac{1}{2}\ddot{\tilde{q}}^T A \ddot{\tilde{q}}, \tag{24}$$

where $\ddot{\tilde{q}}$ is the joint space acceleration induced by the control torque, i.e.,

$$\ddot{\tilde{q}} = A^{-1}(I - P_c)(S^k)^T \Gamma^k. \tag{25}$$

Therefore, the *acceleration energy* can be expressed in terms of Γ^k as

$$E_a = \frac{1}{2}\Gamma^{k^T} W \Gamma^k, \tag{26}$$

where

$$W = S^k(I - P_c)^T A^{-1}(I - P_c)(S^k)^T \tag{27}$$
$$= S^k A^{-1}(I - P_c)(S^k)^T. \tag{28}$$

The control torque, Γ^k, can be chosen as a solution to Equation (23) minimizing E_a. When W is rank-deficient, there can be an infinite number of solutions that minimize the acceleration energy and produce the desired control force on the operational space coordinate. This is due to redundancy in the contact force space. This can be resolved by specifying some of the contact forces or minimizing an additional quantity, e.g. the 2-norm of torque.

We now define the matrix, $(J^k)^T$, as one of the generalized inverses of $\bar{J}^T(S^k)^T$, which minimizes E_a.

$$(J^k)^T = \overline{\bar{J}^T(S^k)^T}. \tag{29}$$

The torque can then be expressed as

$$\begin{aligned}\Gamma^k &= (J^k)^T F \\ &= (J^k)^T \Lambda\{f^* + \mu + p\}.\end{aligned} \tag{30}$$

When $\bar{J}^T(S^k)^T$ is not singular, perfect estimates of all the system matrices will provide

$$\ddot{x} = f^*. \tag{31}$$

B. Null space projection matrix

The null-space projection matrix in the actuated torque space is defined as

$$(N^k)^T = I - (J^k)^T \overline{(J^k)^T} \tag{32}$$

where

$$\overline{(J^k)^T} = \bar{J}^T(S^k)^T \tag{33}$$

The overall torque required to control the operational space coordinate(task) and the null-space(posture) is, then,

$$\Gamma^k = (J^k)^T F + (N^k)^T \Gamma_0^k \tag{34}$$

This torque, Γ^k, in Equation (34) will be simply referred as Γ^k_{task} in the following subsection.

C. Reaction Force Control

The control torque for the desired task formulated in the previous sections assumes that the contact forces are not limited. However, the contact forces are bounded by certain contact conditions. When they exceed some of the limits, the contact state will change and the dynamics of the system will be altered. If we wish to preserve the contact state the contact forces generated by the control torque must be monitored and controlled to remain within the boundaries.

Given the composed task control torque, Γ^k_{task}, the expected contact forces are computed from Equation (9).

$$f_{c,task} = \bar{J}_c^T (S^k)^T \Gamma^k_{task} - \mu_c - p_c. \tag{35}$$

Some of the contact forces may exceed the boundaries associated with the contact conditions. In this case we wish to add additional torques such that the resulting contact forces lie within the boundaries, i.e.,

$$\Gamma^k = \Gamma^k_{task} + \Gamma^k_{contact}. \tag{36}$$

Since the contact forces must be controlled to remain within the boundaries a selection matrix, S_c, can be used to select the components of the contact forces that are exceeding the limits.

$$\begin{aligned}f_{c,selected} &= S_c f_c \\ &= S_c f_{c,task} + S_c \bar{J}_c^T (S^k)^T \Gamma^k_{contact}.\end{aligned} \tag{37}$$

When some of the contact forces, $S_c f_c$, exceed the boundary values we set the desired values of those contact forces to the boundary values. This ensures that the contact forces satisfy the contact conditions. Then,

$$\begin{aligned}\tilde{f}_{c,selected} &= f_{c,selected}|_{desired} - S_c f_{c,task} \\ &= S_c \bar{J}_c^T (S^k)^T \Gamma^k_{contact},\end{aligned} \tag{38}$$

Having defined a matrix $(J_c^k)^T$ as

$$(J_c^k)^T = \overline{S_c \bar{J}_c^T S_k^T}, \tag{39}$$

The additional torque to control contact forces is given by

$$\Gamma^k_{contact} = (J_c^k)^T \tilde{f}_{c,selected}. \tag{40}$$

However, the torque for controlling the contact forces, $\Gamma^k_{contact}$, will affect the task of the robot unless $\Gamma^k_{contact}$ is in the null-space of W. This disturbance to the motion control from $\Gamma^k_{contact}$ can be compensated for by applying additional torques in the null-space of the contact force control.

IV. Transition of Contact State

In Section III we discussed a control framework for task and contact forces in a given contact state, such as one-foot in plane contact or two-feet in plane contact. However, realization of more complex behaviors will involve different contact states. For example, walking will involve one foot in contact as well as two feet in contact. Therefore, the transition between different contact states is necessary.

A proposed strategy involves making the contacts compliant in the transition by limiting the contact forces. When the robot makes contact, the task has to be designed such that the robot approaches the environment compliantly so as to not create a large impact force. That is, the control force for the task has to be designed with very small position feedback. In addition, right after the contact is made the robot has to smoothly increase the use of contact forces for the desired task. Otherwise, there can be a discrete transition in the control torque. A reverse strategy must be applied in the case when the robot loses contact.

V. Simulation Results

The proposed contact consistent control framework has been verified in the SAI simulation environment [8]. The dynamics engine in the SAI environment uses fast algorithms for dynamics and collision computations [3] and [15]. Therefore, it provides not only a simulation environment but also an interactive interface for the user. The red lines in Figures 4, 6, 8, and 9 represent the contact forces simulated in SAI.

A. Standing on two feet

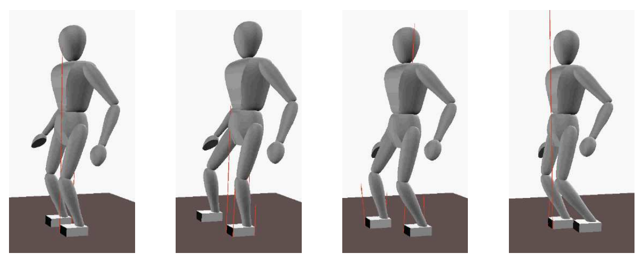
Standing on two feet involves many possible contact configurations which result in stable standing. One natural configuration involves plane contact on both feet. In this case, each foot has 6 motion constraints, resulting in a total of 12 constraints. These constraints create a contact force space of 12 degrees of freedom. Since the number of constraints exceeds the number of virtual joints (6), 6 DOF in the 12 DOF contact force space can be controlled without disturbing motion of the robot.

Utilizing this space, the moments acting on both feet can be set to zero, with the exception of the moment about the axis connecting the center of mass points of the two feet. This is a 4 DOF condition resembling natural bipedal standing, where the moments on the foot are only used for balance in the forward/backward direction. The desired contact forces are composed from this 4 DOF condition and the contact conditions for maintaining current contacts .

Compensation for the gravity torques and adhering to the above conditions are sufficient to enable the robot to stand on two feet with plane contacts. This assumes that the starting configuration of the robot is statically balanced, i.e. the robot's center of mass is in between the feet. To maintain balance in the presence of disturbance, the minimal control required is control over the center of mass point. This can be realized by choosing the position of the robot's center of mass as the operational space coordinate.

Applying operational space control only for the center of mass creates compliant behavior for the robot in response to disturbances or external forces. When external forces are applied to a given link of the robot, the controller moves the other links to maintain the center of mass at the desired position rather than maintaining a fixed joint configuration. This kind of compliant behavior cannot be realized when all the joints of the robot are controlled to follow specific trajectories.

B. Walking



(a) (b) (c) (d)

Fig. 4. **Walking.** (a) Starting on two feet contact. The center of mass is controlled to move toward the left foot. (b) Left foot supporting phase. The right foot is controlled to move to the desired foot position. (c) Two feet supporting phase. The center of mass is controlled to move toward the right foot. (d) Right foot supporting phase. The left foot is controlled to move to the desired foot position.

In order to realize a walking behavior, the control variables, i.e. operational space coordinates, are chosen as the center of mass position and the orientation of the head, chest, and hip. In the one foot support phase, the foot in free space is also controlled. The primary control variables for walking (locomotion) are the center of mass and individual foot positions. The other coordinates are chosen to maintain the desired posture of the robot while walking.

Figure 4 and Figure 5 display snapshots and trajectories, respectively. Figure 5 shows how the center of mass and the feet are coordinated during walking. The gait cycle is designed first to produce the desired foot trajectories. Then, the desired center of mass motion is composed based upon the foot motion. In the two foot support phase, the center of mass is controlled to move toward the foot which will be the support foot in the next phase. In the one foot support phase the foot in free space is controlled to move to the desired foot placement. In this phase, the center of mass position is also

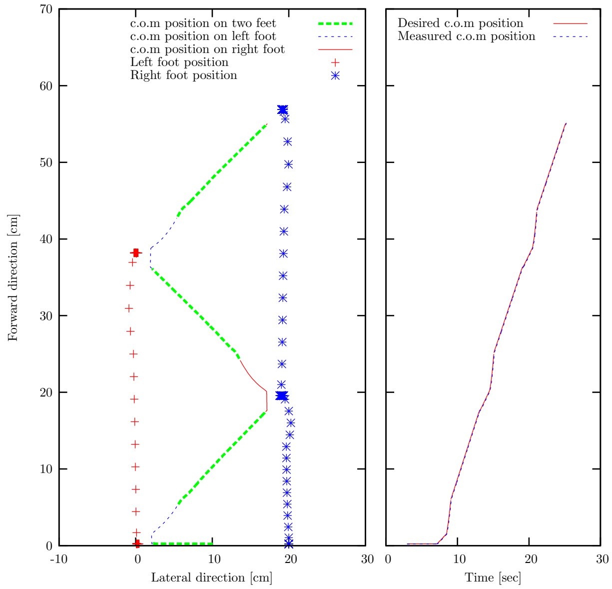


Fig. 5. **Walking.** The robot starts from a standing configuration where both feet are side by side. The center of mass of the robot (green trajectory) is then moved over the left foot. The right foot is then moved foward (blue trajectory) while the left foot acts as support. In this phase, the center of mass is also controlled to move toward the right foot, resembling dynamic walking. The final phase involves transition between left and right foot contact. Both feet do not move but the center of mass moves toward the next support, in this case, the right foot. The gait cycle is then repeated.

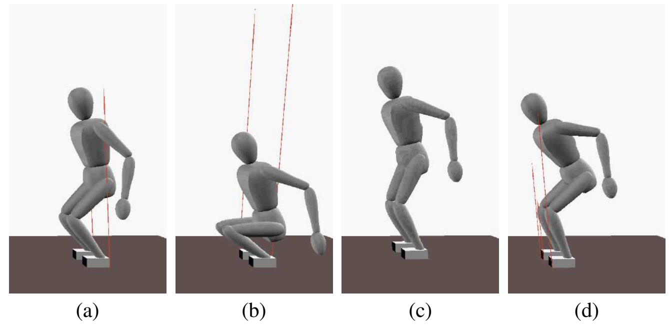


Fig. 6. **Jumping.** (a) Squatting phase. The robot is preparing to leap. (b) Leaping phase. (c) No contact. The robot is preparing for landing. (d) Landing.

controlled to move toward the next supporting foot so that the robot moves its center of mass in advance before making two foot contact. This motion resembles human dynamic walking.

The dimension of the operational space is less than the total DOF of the system, leaving both arms without any specific controls. Therefore, the arms are compliant to disturbances during walking. Also, the controls for the posture, which include the orientation of head, chest, and hip, use very low gains such that the robot recovers its desired posture over time in the presence of disturbances, but is compliant at the instance of the disturbance. Therefore, it provides robustness to unexpected external forces imparted to the robot during walking.

C. Jumping

A jumping behavior is realized by controlling the same operational space coordinates as those in the *walking* behavior. The trajectory of the center of mass is designed such that it has a squatting phase, a leaping phase, a no-contact (airborne) phase, and a landing phase. The center of mass cannot be controlled when the robot is in the no-contact phase. However, once the feet have left the ground the system is controlled to generate a robust landing posture. Both feet are controlled to move to the expected landing position. The compliant behavior of the robot at the beginning of the landing is implemented by choosing low gains for all the controls of the operational space coordinates, particularly for the center of mass. Also, the contact forces applied to the environment are monitored and controlled to increase smoothly.

D. Climbing a ladder

The proposed contact consistent control framework can be applied to any robot link in contact. This is demonstrated by implementing a *ladder climbing* behavior (Figure 8). In this example, the hands are controlled to be in contact in order to maintain balance and also to control the ascension of the center of mass. A similar design procedure to that of the walking behavior is implemented to produce this complicated motion. The climbing cycle is produced by designing the foot and hand positions at the ladder contacts. The trajectory of the center of mass is chosen to maintain stable balance at all times.

E. Manipulation combined with walking

The final example involves the execution of a manipulation task while walking (Figure 9). In addition to walking, the hand is controlled to follow a trajectory in a contact plane and also to maintain contact forces in the normal direction of contact. This demonstrates the generality of the proposed control structure. That is, the proposed control framework enables us to implement hybrid motion/force control on humanoid robots using the operational space framework.

VI. Conclusion

This paper presents the complete dynamics of a humanoid system or, more generally, any robotic system in contact with the environment using virtual joints. The dynamic equations are derived by considering the system to be constrained by the contacts. This contact consistent dynamic formulation is essential for composing an appropriate control strategy for the robotic system. The proposed control strategy utilizes operational space coordinates, which may include the *center of mass* position as well as positions and orientations of the *hip*, *foot*, and *head*. The operational space control framework

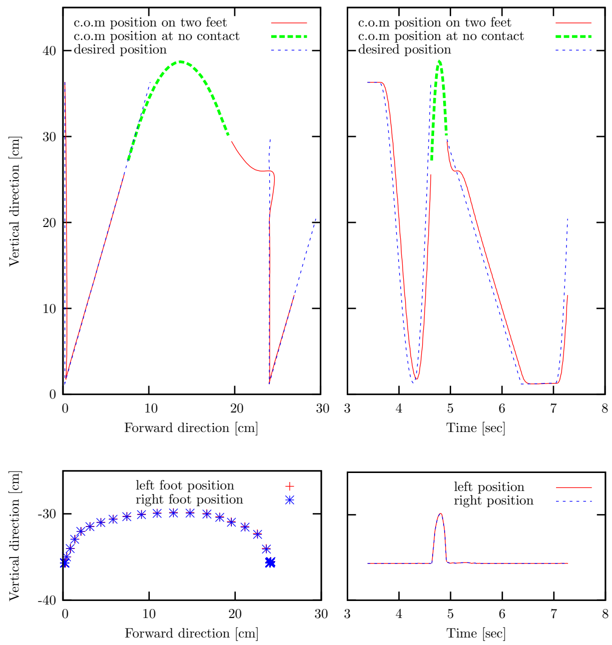
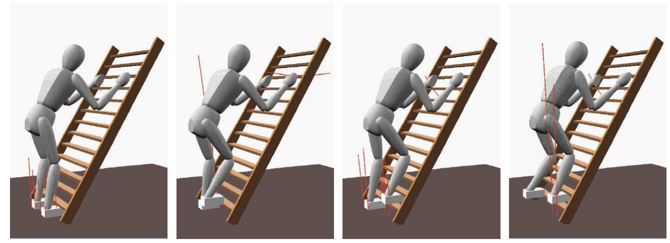


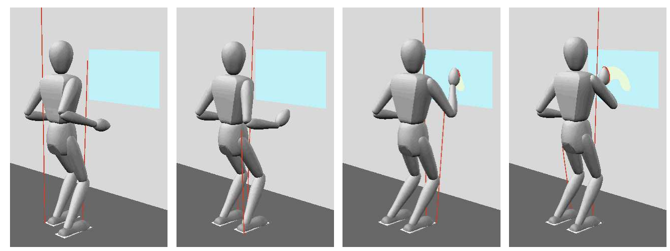
Fig. 7. **Jumping.** The motion starts with a squatting phase, where the center of mass is controlled to move down. The leaping phase generates acceleration in the vertical direction. In the next phase the robot is airborne and the center of mass can no longer be controlled. The green line shows the parabolic trajectory of the center of mass in this phase. Both feet are controlled to move forward into a landing position. At the beginning of the landing phase, very small gains are set to control the center of mass such that the robot is compliant. This prevents bouncing on the ground due to the stiff motion of the body.

enables us to design different control for each control variable (e.g. such as choosing different gains). Therefore, more complicated human behaviors can be implemented by allowing for more possibilities in the controller design.



(a) (b) (c) (d)

Fig. 8. **Climbing a ladder.** (a) The robot begins to climb. It has contacts on both hands and feet. (b) The right foot is then controlled to move up one step. (c) Next, the center of mass is controlled to move to the right in order to maintain balance with two hands and the right foot. (d) The left foot is then controlled to move up one step.



(a) (b) (c) (d)

Fig. 9. **Walking with manipulation.** (a) The robot walks to the window. (b) The right hand is then controlled to make contact with the window. (c) Next, the robot begins cleaning the window by applying a specified normal force and following a trajectory. (d) The robot is shown walking while controlling the hand motion.

References

[1] P. Baerlocher and R. Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 323–329, Victoria, B.C., Canada, 1998.

[2] H. Bruyninckx and O. Khatib. Gauss' principle and the dynamics of redundant and constrained manipulators. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 470–475, 2000.

[3] K. C. Chang and O. Khatib. Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms. In *Proc. of the Int. Conf. on Robotics and Automation*, 2000.

[4] Stefano Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *Int. J. on Robotics and Automation*, 13(3):398–410, June 1997.

[5] K. Daneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi. Design of prototype humanoid robotics platform for hrp. In *Proc. of the Int. Conf. on Intelligent Robots and Systems*, pages 2431–2436, 2002.

[6] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 1321–1326, 1998.

[7] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 1620–1626, 2003.

[8] 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.

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

[10] J. Kim, I. Park, J. Lee, M. Kim, B. Cho, and J. Oh. System design and dynamic walking of humanoid robot khr-2. In *Proc. of the Int. Conf. on Robotics and Automation*, 2005.

[11] A. D. Kuo. An optimal control model for analyzing human postural balance. *IEEE Transactions on Biomedical Engineering*, 42(1):87–101, Jun 1995.

[12] Napoleon, S. Nakaura, and M. Sampei. Balance control analysis of humanoid robot. In *Proc. of the Int. Conf. on Intelligent Robots and Systems*, pages 2437–2442, 2002.

[13] Dragomir N. Nenchev and Zlatko M. Sotirov. Dynamic task-priority allocation for kinematically redundant robotic mechanisms. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 518–524, 1994.

[14] J. Pratt and G. Pratt. Intuitive control of a planar biped walking robot. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 2014–2021, 1998.

[15] D. Ruspini and O. Khatib. Collision/contact models for dynamic simulation and haptic interaction. In *The 9th International Symposium of Robotics Research*, pages 185–195, Snowbird, USA, 1999.

[16] V. De Sapio and O. Khatib. Operational space control of multibody systems with explicit holonomic constraints. In *Proc. of the Int. Conf. on Robotics and Automation*, 2005.

[17] Bruno Siciliano and Jean-Jacques E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Proc. of the Int. Conf. on Advanced Robotics*, pages 1211–1216, 1991.

[18] F. E. Udwadia and R. E. Kalaba. *Analytical Dynamics: A New Approach*. Cambridge: Cambridge University Press, 1996.