2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Whole-Body Humanoid Control from Upper-Body Task Specifications

Ghassan Bin Hammam[1], David E. Orin[1], Behzad Dariush[2]

Abstract— **This paper introduces a very efficient, modified resolved acceleration control algorithm for dynamic filtering and control of whole-body humanoid motion in response to upper-body task specifications. The dynamic filter is applicable for general upper-body motions when standing in place. It is characterized by modification of the commanded torso acceleration based on a geometric solution to produce a ZMP which is inside the support. The resulting feasible modified motion is synchronized to the reference motion when the computed ZMP for the reference motion again falls within the support. Contact forces at each foot are controlled through a dedicated force distribution module which optimizes the ankle roll and pitch torques. The proposed approach uses time-local information and is therefore targeted for online control. The effectiveness of the algorithm is demonstrated by means of simulated experiments on a model of the Honda humanoid robot** ASIMO **using a highly dynamic upper-body reference motion.**

I. INTRODUCTION

The emergence of the field of humanoid robotics in the last decade is largely attributed to the expectation that humanoid robots will eventually become an integral part of our everyday lives, serving as caretakers for the elderly and disabled, providing assistance in homes and offices, and assisting in surgery and physical therapy. From a control perspective, much effort has been aimed at addressing various aspects of humanoid robot control, such as motion execution, safety, constraint handling, multi-contact control, balance control, and obstacle avoidance [1]. Although the utility of robots operating in human environments rests, to a large extent, on execution of upper-body motion and manipulation tasks, the ability to sustain dynamic balance in response to upper-body motion is an important problem that is unique to humanoid robotics research.

Whole-body motion from upper-body task specifications has been examined by several research groups and significant advances have been reported. Given that the hardware platform of many humanoid robots is designed for position control, the majority of past and present developments in whole-body motion control have been centered around kinematic and inverse kinematic control techniques. Naksuk et al. [2] considered the problem of off-line motion editing when transferring pre-recorded motion from a human to a humanoid robot under the assumption of a single support stance. The proposed scheme yielded a balanced humanoid motion with minimal angular momentum at the center of mass. Ott et al. [3] proposed whole-body motion generation from upper-body human motion capture using a marker system; however, their approach does not handle highly dynamic motion since they use a constant center of gravity position centered inside the support polygon in their balance controller. More recently, Dariush et al. [4] demonstrated real time replication of complex upper-body human motion on Honda's ASIMO platform (Fig. 1). The robot imitated human motion while satisfying kinematic constraints such as joint limits, velocity limits, and self collision avoidance. Balance control was handled using ASIMO'S zero moment point (ZMP) controller which accommodated potential loss of balance due to upper-body motion by effectively translating the trunk. Balance control formulations based on such methods have been effective in handling quasi-static, and periodic movements (such as walking or running). However, there are limits to how well kinematic control can perform in handling highly dynamic and/or non-periodic movements.

Fig. 1. Honda's ASIMO.

Although dynamic control at the acceleration or force level is not possible on a majority of humanoid platforms which operate in a position command mode, new theoretical formulations for whole-body motion control based on force/torque commands have been demonstrated in simulated environments [5]. Sentis and Khatib [6] developed a unified framework for whole-body task control in the context of multi-point multi-link contacts, constraints, and obstacles in the extended operational space framework. However, control of dynamic balance was not rigorously treated. Yamane and Nakamura [7] developed a full-body dynamic filter that converts a physically infeasible reference motion into a feasible one. However, their approach alters all input motion, including the specified upper-body motion, and is not applicable where upper-body task specifications must

[1] G. Bin Hammam and D. E. Orin are with the Department of Electrical & Computer Engineering, The Ohio State University, Columbus, OH 43210, U.S.A., email: bin-hammam.1@osu.edu, orin.1@osu.edu
[2] B. Daruish is with the Honda Research Institute, Mountain View, CA 94041, U.S.A., email: dariush@hra.com
This work is supported in part by the Honda Research Institute, USA.

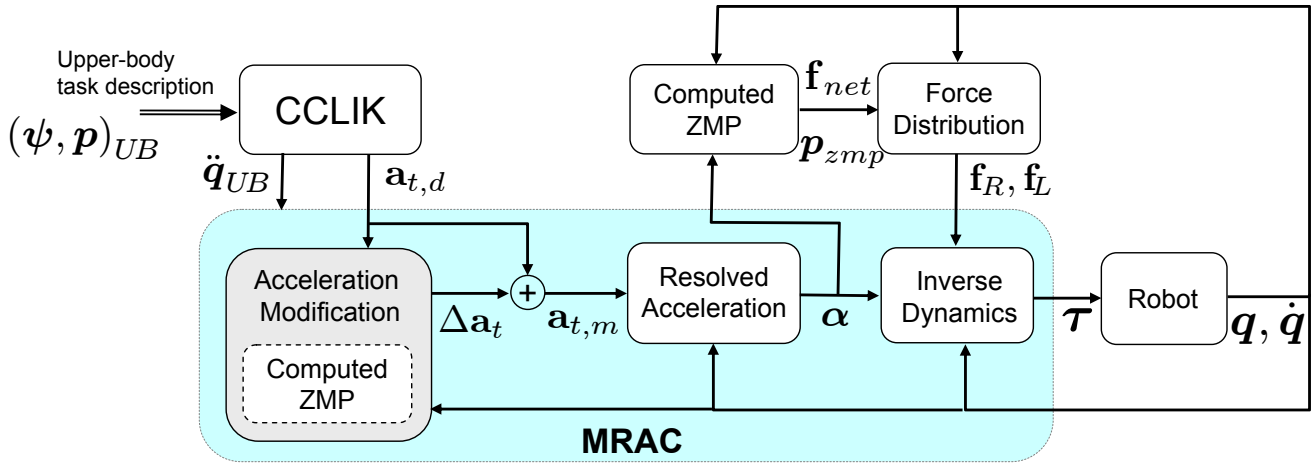


Fig. 2. System block diagram with modified resolved acceleration control (MRAC).

be preserved. Furthermore, their approach requires careful parameter tuning (feedback gains and weights for pseudo-inverses) for each behavior.

This paper introduces a very efficient, modified resolved acceleration control algorithm for dynamic filtering and control of whole-body humanoid motion in response to upper-body task specifications. The key contribution is the development of a novel method for online control of the ZMP position by modifying the torso acceleration under both single and double support stance. In order to arrive at an effective solution to this problem, we introduce four key components: 1) a modification of the commanded torso acceleration based on a geometric solution to produce a ZMP which is inside the support, 2) an algorithm to synchronize the feasible modified motion to the reference motion when the computed ZMP for the reference motion again falls within the support, 3) an algorithm to determine the computed ZMP based on a very efficient recursive Newton-Euler algorithm [8], and 4) an algorithm to distribute the resultant ground reaction force between the two feet by minimizing the ankle roll and pitch torques. We demonstrate the effectiveness of the algorithm by means of simulated experiments on a model of the Honda humanoid robot ASIMO (Figure 1) using a highly dynamic upper-body reference motion.

II. Overview

We assume a humanoid robot is in double support stance and commanded to execute an upper-body motion expressed by Cartesian (or task) variables. The objective is to realize the specified upper-body task motion while ensuring that the robot is balanced through appropriate modification of the desired torso acceleration. The block diagram of the overall system is illustrated in Figure 2. The inputs include Cartesian variables representing the desired upper-body orientation and position task descriptors, $(\psi, \boldsymbol{p})_{UB}$. The first step involves kinematic inversion of the specified task motion to joint motion commands subject to kinematic constraints, including joint limit constraints, joint velocity constraints, and self collision constraints. This procedure is performed using an online constrained closed loop inverse kinematics (CCLIK) algorithm that is described in detail elsewhere [4]. The output of this module includes the upper-body joint acceleration vector, $\ddot{\boldsymbol{q}}_{UB}$, and the desired torso acceleration vector, $\mathbf{a}_{t,d}$.

The next step is to assess the feasibility of the upper-body joint motion by analyzing the computed ZMP and making adjustments to the torso acceleration when necessary. This procedure is performed in the *Acceleration Modification* module. The modified torso, and upper-body, accelerations are then applied to a *Resolved Acceleration* module which uses position and velocity feedback to compute the resolved acceleration vector $\boldsymbol{\alpha}$ in terms of joint variables. The *Inverse Dynamics* module produces the torque commands, denoted by $\boldsymbol{\tau}$, to realize the desired motion. The tracking control based on acceleration modification is referred to as modified resolved acceleration control (MRAC). For the case that the robot is in double support stance, the inverse dynamics procedure requires the distribution of the net ground reaction forces and moments $\mathbf{f}_{net}$ between the two feet, denoted by $\mathbf{f}_R$ and $\mathbf{f}_L$. These computations are performed in the *Computed ZMP* and *Force Distribution* modules.

III. Humanoid Dynamic Model

In order to develop the dynamic model of a humanoid robot, the approach taken in [8] for rigid-body systems is used. Spatial notation [8], [9] is a concise vector notation for describing rigid-body velocity, acceleration, force, inertia, etc., using 6D vectors and tensors, and is an integral part of the approach.

A humanoid can be modeled as a set of $N + 1$ links interconnected by N joints, of up to six degrees of freedom each, forming a tree-structure topology. The motion of the links are referenced to a fixed base (inertial frame) which is labeled 0 while the links are labeled from 1 through N. In our model, the inertial frame is attached to the ground. The

spatial velocity and acceleration of link i are represented as:

$$\mathbf{v}_i = \begin{bmatrix} \boldsymbol{\omega}_i \\ \boldsymbol{v}_i \end{bmatrix}, \quad (1)$$

$$\mathbf{a}_i = \begin{bmatrix} \dot{\boldsymbol{\omega}}_i \\ \dot{\boldsymbol{v}}_i \end{bmatrix}, \quad (2)$$

where $\boldsymbol{\omega}_i$, $\boldsymbol{v}_i$, $\dot{\boldsymbol{\omega}}_i$, and $\dot{\boldsymbol{v}}_i$ are the angular, and linear velocities, the angular, and linear accelerations of link i, respectively, as referenced to the link coordinate frame. In order to model a humanoid when in flight, one of the links is modeled as a floating base (typically the torso) and numbered as link 1. A fictitious six degree-of-freedom (DoF) joint is inserted between the floating base and fixed base. The total number of degrees of freedom in the humanoid is n where $n = \sum n_i$, and n_i is the number of degrees of freedom for joint i which connects link i to its predecessor. Note that n includes the six DoFs for the floating base.

The spatial force acting on link i from its predecessor is represented as:

$$\mathbf{f}_i = \begin{bmatrix} \boldsymbol{n}_i \\ \boldsymbol{f}_i \end{bmatrix}, \quad (3)$$

where $\boldsymbol{n}_i$ is the moment about the origin of the link coordinate frame, and $\boldsymbol{f}_i$ is the translational force referenced to the link coordinate frame.

The spatial coordinate transformation matrix ${}^i\boldsymbol{X}_j$ may be composed from the position vector ${}^j\boldsymbol{p}_i$ from the origin of coordinate frame j to the origin of i, and the 3×3 rotation matrix ${}^i\boldsymbol{R}_j$ which transforms 3D vectors from coordinate frame j to i:

$${}^i\boldsymbol{X}_j = \begin{bmatrix} {}^i\boldsymbol{R}_j & \mathbf{0}_{3\times 3} \\ {}^i\boldsymbol{R}_j\, \boldsymbol{S}({}^j\boldsymbol{p}_i)^T & {}^i\boldsymbol{R}_j \end{bmatrix}. \quad (4)$$

The quantity $\boldsymbol{S}(\boldsymbol{p})$ is the skew-symmetric matrix that satisfies $\boldsymbol{S}(\boldsymbol{p})\,\boldsymbol{\omega} = \boldsymbol{p} \times \boldsymbol{\omega}$ for any 3D vector $\boldsymbol{\omega}$. This transformation matrix can be used to transform spatial quantities from one frame to another as follows:

$$\mathbf{v}_j = {}^j\boldsymbol{X}_i\, \mathbf{v}_i, \quad (5)$$

$$\mathbf{a}_j = {}^j\boldsymbol{X}_i\, \mathbf{a}_i, \quad (6)$$

$$\mathbf{f}_j = {}^j\boldsymbol{X}_i^{-T}\, \mathbf{f}_i\,. \quad (7)$$

The equations of motion of a robotic mechanism in joint-space can be written as:

$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})\, \ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\, \dot{\boldsymbol{q}} + \boldsymbol{\tau}_g(\boldsymbol{q}) + \boldsymbol{J}^T \mathbf{f}_e\,, \quad (8)$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, and $\boldsymbol{\tau}$ denote n-dimensional generalized vectors of joint position, velocity, acceleration and force variables, respectively. $\boldsymbol{H}(\boldsymbol{q})$ is an $(n \times n)$ joint-space inertia matrix. $\boldsymbol{C}$ is an $(n \times n)$ matrix such that $\boldsymbol{C}\,\dot{\boldsymbol{q}}$ is the vector of Coriolis and centrifugal terms. $\boldsymbol{\tau}_g$ is the vector of gravity terms. $\boldsymbol{J}$ is a Jacobian matrix, and $\mathbf{f}_e$ is the external spatial force acting on the system. When the feet are the only contacts for the system with the environment, the external force includes the foot spatial contact forces (ground reaction force/moment),

$$\mathbf{f}_e = \begin{bmatrix} \mathbf{f}_R \\ \mathbf{f}_L \end{bmatrix}, \quad (9)$$

where $\mathbf{f}_R$, and $\mathbf{f}_L$ are the right and left foot spatial contact forces, respectively. Friction and disturbance inputs can easily be added to these equations as well.

In the Inverse Dynamics (ID) problem, given the desired joint accelerations, the joint torques $\boldsymbol{\tau}$ are computed,

$$\boldsymbol{\tau} = ID(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \mathbf{f}_R, \mathbf{f}_L)\,, \quad (10)$$

and

$$\boldsymbol{\tau} = \left[\boldsymbol{\tau}_{UB}^T\ \mathbf{f}_t^T\ \boldsymbol{\tau}_R^T\ \boldsymbol{\tau}_L^T\right]^T, \quad (11)$$

where $\boldsymbol{\tau}_{UB}$, $\boldsymbol{\tau}_R$, and $\boldsymbol{\tau}_L$ are the joint torques for the upper body, right leg, and left leg, respectively. $\mathbf{f}_t$ is the force on the torso (the floating-base link), and it will be zero if the external (foot) forces are consistent with the given system acceleration since the torso is not actuated. In this paper we use the very efficient $O(n)$ Recursive Newton-Euler Algorithm (RNEA) for kinematic trees [8]. The RNEA is efficient because it calculates most of the quantities in local link coordinates and it includes the effects of gravity in an efficient manner.

IV. COMPUTED ZMP ALGORITHM

Whole-body humanoid control requires information about the feasibility of the motion to be performed according to the upper-body task specifications. Computing the Zero-Moment Point (ZMP) for a given motion can predict its feasibility and the balance of the humanoid. In this section, we propose a very efficient, yet simple algorithm to compute the ZMP for a given whole-body motion.

The ZMP is defined as the point on the support base at which the resulting reaction forces are acting [10], [11]. Therefore, if the resultant (net) spatial force $\mathbf{f}_{net} = \left[\boldsymbol{n}_{net}^T\ \boldsymbol{f}_{net}^T\right]^T$ is known as in Fig. 3b, then the ZMP position may be computed as ${}^0p_{zmp}^x = -n_{net}^y / f_{net}^z$, and ${}^0p_{zmp}^y = n_{net}^x / f_{net}^z$.

The algorithm developed here to compute the ZMP is based on determining the resultant foot force (force and moment) for a given system acceleration. By solving the inverse dynamics problem using the RNEA for a given system acceleration while applying zero foot forces (free-space inverse dynamics), the resultant spatial force on the system (the torso in the case of RNEA) can be computed as in Fig. 3a. According to Newton's laws of motion, this spatial force can be applied to any body of the system; therefore, if it is transformed into the inertial frame (ground), the resultant ground reaction force (resultant foot force) will be obtained (Fig. 3b) and then the ZMP position is computed. Algorithm 1 summarizes the proposed algorithm. Note that the resulting algorithm is quite efficient because the main computation is the RNEA for inverse dynamics for kinematic trees.

V. MODIFIED RESOLVED ACCELERATION

In this section, we present the tracking control method used in this paper. For the lower-body motion including the torso motion, we introduce Resolved Acceleration Control (RAC) for a humanoid in Section V-A. RAC has been

Algorithm 1: An efficient algorithm to compute the ZMP for a given whole-body motion.

Input: *model*, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$

Output: ${}^0\boldsymbol{p}_{zmp}$

begin

$$\boldsymbol{\tau} = ID\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \mathbf{0}, \mathbf{0}\right)$$

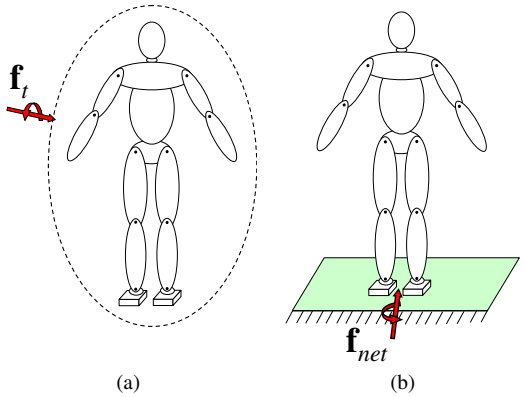
$$\mathbf{f}_{net} = {}^0\boldsymbol{X}_t^{-T}\, \mathbf{f}_t$$

$${}^0p^z_{zmp} = 0$$

$${}^0p^x_{zmp} = -n^y_{net} / f^z_{net}$$

$${}^0p^y_{zmp} = n^x_{net} / f^z_{net}$$

end



(a) (b)

Fig. 3. Force transformation to compute the ZMP. (a) Net force on the system (torso). (b) Net force on the feet.

used as a control for robot manipulators [12], [13], [14]. However, to the best of our knowledge it has yet to be developed to control the motion of the lower-body in a humanoid, especially in double support. Our approach allows coordination of the legs to give the desired motion of the torso, and coupled with the force distribution method given in the next section, provides for control of the individual foot forces.

The RAC requires feasible motion to track; therefore, a mechanism for motion modification is necessary for RAC motion tracking when the motion is infeasible (not balanced). In Section V-B a method is developed to modify the system acceleration so that the motion is feasible. In order to track the desired motion of the torso from the upper-body task specifications, the modified motion needs to be quickly and smoothly moved to the desired motion when it is again feasible. A method to get the modified motion in synchronization with the desired motion is proposed in Section V-C.

A. Resolved Acceleration Control

The equations for RAC for a humanoid are developed in this section. The output of RAC plus the foot forces from the force distribution module described in Section VI are used to compute the lower-body joint torques required to produce the modified motion of the torso, using Inverse Dynamics as in Eq. 10.

Let $\boldsymbol{\psi}_t$ and $\boldsymbol{p}_t$ describe the torso orientation (described by Euler angles) and position, respectively. The task-descriptor vector $\mathbf{x}_t = \left[\boldsymbol{\psi}_t^T\, \boldsymbol{p}_t^T\right]^T$ is related to the spatial velocity of the torso $\mathbf{v}_t$ via an equation of the form

$$\dot{\mathbf{x}}_t = \boldsymbol{E}(\mathbf{x}_t)\, \mathbf{v}_t\,, \tag{12}$$

where $\boldsymbol{E}$ is a 6×6 transformation matrix that depends on the Euler sequence used. The resolved acceleration is set by $\boldsymbol{\alpha} = \ddot{\boldsymbol{q}}$ where

$$\ddot{\boldsymbol{q}} = \left[\, \ddot{\boldsymbol{q}}_{UB}^T\, \mathbf{a}_t^T\, \ddot{\boldsymbol{q}}_R^T\, \ddot{\boldsymbol{q}}_L^T\right]^T \tag{13}$$

is the commanded whole-body acceleration (system acceleration) which includes compensation for feedback errors. The torso acceleration $\mathbf{a}_t$ is computed as

$$\mathbf{a}_t = \mathbf{a}_{t,m} + K_{v,t}\, (\mathbf{v}_{t,m} - \mathbf{v}_t) + K_{p,t}\, \boldsymbol{e}_t\,, \tag{14}$$

where

$$\mathbf{a}_{t,m} = \mathbf{a}_{t,d} + \Delta \mathbf{a}_t\,. \tag{15}$$

$\mathbf{v}_{t,m}$ and $\mathbf{a}_{t,m}$ are the modified torso spatial velocity and acceleration, respectively, and will be discussed in Section V-B. $\mathbf{v}_t$ is the current torso spatial velocity and $\mathbf{a}_{t,d}$ is the desired torso spatial acceleration from the original motion. $\boldsymbol{e}_t$ is the position and orientation error and it is given by [12]:

$$\boldsymbol{e}_t = \begin{bmatrix} 0.5 \left(\sum_{j=1}^{3} \mathbf{c}_j \times \mathbf{c}_{j,m}\right) \\ {}^0\boldsymbol{p}_{t,m} - {}^0\boldsymbol{p}_t \end{bmatrix}, \tag{16}$$

where ${}^0\boldsymbol{p}_{t,m}$ and ${}^0\boldsymbol{p}_t$ are the modified and current torso positions, and $\mathbf{c}_j$, and $\mathbf{c}_{j,m}$ are jth columns of the current orientation ${}^0\boldsymbol{R}_t$, and the modified orientation ${}^0\boldsymbol{R}_{t,m}$ matrices, respectively. $K_{p,t}$ and $K_{v,t}$ are matrices of proportional and derivative gains for torso control.

For a tree-structure the right and left foot velocities $(\mathbf{v}_R, \mathbf{v}_L)$ can be computed from

$$\mathbf{v}_R = \boldsymbol{J}_R\, \dot{\boldsymbol{q}}_R + {}^R\boldsymbol{X}_t \mathbf{v}_t\,, \tag{17}$$

$$\mathbf{v}_L = \boldsymbol{J}_L\, \dot{\boldsymbol{q}}_L + {}^L\boldsymbol{X}_t \mathbf{v}_t\,, \tag{18}$$

where $\boldsymbol{J}_R$, $\boldsymbol{J}_L$, $\dot{\boldsymbol{q}}_R$, $\dot{\boldsymbol{q}}_L$ denote the right and left leg Jacobian matrices, and the right and left leg velocities, respectively. Note that ${}^R\boldsymbol{X}_t \mathbf{v}_t$ represents the velocity of the torso at a point which instantaneously coincides with the right foot position, etc.. By taking the derivative of Eqs. 17 and 18 and then re-ordering the resulting equations, the right and left leg accelerations ($\ddot{\boldsymbol{q}}_R$ and $\ddot{\boldsymbol{q}}_L$) can be computed as:

$$\ddot{\boldsymbol{q}}_R = \boldsymbol{J}_R^{\dagger} \left(\mathbf{a}_R - {}^R\boldsymbol{X}_t \mathbf{a}_t - \dot{\boldsymbol{J}}_R\, \dot{\boldsymbol{q}}_R\right), \tag{19}$$

$$\ddot{\boldsymbol{q}}_L = \boldsymbol{J}_L^{\dagger} \left(\mathbf{a}_L - {}^L\boldsymbol{X}_t \mathbf{a}_t - \dot{\boldsymbol{J}}_L\, \dot{\boldsymbol{q}}_L\right), \tag{20}$$

where $\boldsymbol{J}^{\dagger}$ denotes the pseudo-inverse of $\boldsymbol{J}$. For the case when each leg has 6 DoFs, as in many humanoids including Honda's ASIMO, the pseudo-inverse $\boldsymbol{J}^{\dagger}$ can simply be replaced by the inverse $\boldsymbol{J}^{-1}$. For the case of constrained feet

(as in single or double support cases), $\mathbf{a}_R$ and/or $\mathbf{a}_L = \mathbf{0}$. The control law is the same as Eq. 8 after replacing $\ddot{\boldsymbol{q}}$ with $\boldsymbol{\alpha}$.

The upper-body accelerations for the arms are set by using computed-torque control [15]

$$\ddot{\boldsymbol{q}}_{UB} = \ddot{\boldsymbol{q}}_{UB,d} + K_{v,UB}\ \dot{\boldsymbol{e}}_{UB} + K_{p,UB}\,\boldsymbol{e}_{UB}\,, \tag{21}$$

where

$$\boldsymbol{e}_{UB} = \boldsymbol{q}_{UB,d} - \boldsymbol{q}_{UB}\,, \tag{22}$$

$$\dot{\boldsymbol{e}}_{UB} = \dot{\boldsymbol{q}}_{UB,d} - \dot{\boldsymbol{q}}_{UB}\,. \tag{23}$$

B. Acceleration Modification

The RAC tracks a feasible motion; therefore, when the computed ZMP is outside the support base, the humanoid can become dynamically unstable and the RAC will fail. We implement a method to modify the torso (task-descriptor) acceleration so that the motion is feasible, balanced, and controllable. This method filters the dynamic effects of the motion that cause the ZMP to leave the support base. This is accomplished through the following steps:

1) Modify the computed ZMP: the computed ZMP should be brought back inside the area of support. The computed ZMP is projected inside the support area with an appropriate safety margin. The new ZMP is designated as the modified ZMP ${}^0\boldsymbol{p}_{zmp,m}$.
2) Compute the modified net force $\mathbf{f}_{net,m}$ and moment $n^z_{net,m}$: the net force at ${}^0\boldsymbol{p}_{zmp,m}$ is derived from the net force $\mathbf{f}_{net}$ so that the modified force has the same normal component as the net force, and the same moment about the torso origin. In particular,

$$f^z_{net,m} = f^z_{net}\,, \tag{24}$$

$${}^t\boldsymbol{p}_{zmp,m}\times\boldsymbol{f}_{net,m}+\begin{bmatrix}0\\0\\n^z_{net,m}\end{bmatrix} = {}^t\boldsymbol{p}_{zmp}\times\boldsymbol{f}_{net}+\begin{bmatrix}0\\0\\n^z_{net}\end{bmatrix}\,. \tag{25}$$

3) Compute the bias torso force: the bias force on the torso due to the net force change is computed as

$$\Delta\mathbf{f}_t \;=\; {}^t\boldsymbol{X}_0^{-T}\;(\mathbf{f}_{net,m} - \mathbf{f}_{net})\;. \tag{26}$$

The force is chosen to be applied to the torso since it is the most massive link of the system; hence its dynamic effect is significant. Fig. 4 illustrates Steps 1-3 of this method.

4) Compute the bias torso acceleration (change in acceleration): the acceleration change caused by $\Delta\mathbf{f}_t$ may be computed with the dynamic equation for the torso

$$\Delta\mathbf{f}_t \;=\; \boldsymbol{I}(\boldsymbol{q})\;\Delta\mathbf{a}_t \tag{27}$$

where $\boldsymbol{I}(\boldsymbol{q})$ is the operational-space inertia matrix as seen at the torso which is approximated here by the articulated-body inertia for the tree-structure system at the torso [8]. The torso acceleration is modified by $\Delta\mathbf{a}_t$.

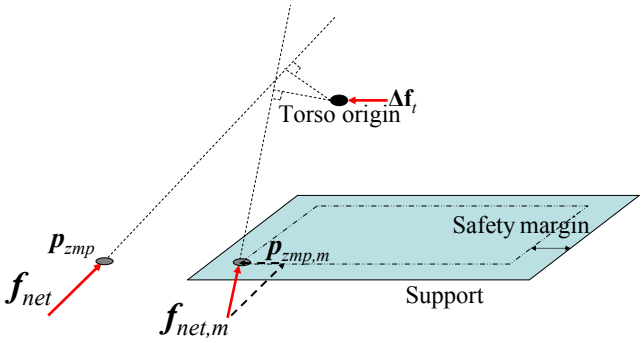


Fig. 4. Force change $\Delta\mathbf{f}_t$ associated with change of ZMP for motion modification.

5) Compute the new modified motion: based on the change of torso acceleration, the torso motion is updated as follows

$$\ddot{\mathbf{x}}_{t,m}(t) \;=\; \ddot{\mathbf{x}}_{t,d}(t) + \Delta\ddot{\mathbf{x}}_t(t)\,, \tag{28}$$

where

$$\Delta\,\ddot{\mathbf{x}}_t(t) \;=\; \dot{\boldsymbol{E}}(\mathbf{x})\;\mathbf{v}_t(t) + \boldsymbol{E}(\mathbf{x})\;\Delta\mathbf{a}_t(t)\,. \tag{29}$$

Note that whenever the computed ZMP for the motion is outside the support area, the motion controller does not track the desired motion; instead it tracks the modified motion that brings the ZMP back to inside the support.

C. Synchronization of Modified Trajectory

Once the computed ZMP is inside the support area after the motion has been modified, the modified and the desired original motion states may be significantly different. Therefore, trying to track the desired motion immediately might cause instability to the RAC controller due to the large acceleration injected into the system. Even worse, this might cause the computed ZMP to be outside the support area. Note that in this case the ZMP is outside the support area due to the large feedback error (see Eq. 14).

We synchronize the modified motion using a bang-bang solution which results in the minimum acceleration required over the synchronization period [16]. The synchronization period is chosen so that the synchronization acceleration $\ddot{\mathbf{x}}_s$ does not introduce additional significant dynamic effects that might cause the ZMP to go outside the support area. The original motion is modified as follows:

$$\ddot{\mathbf{x}}_{t,m}(t) \;=\; \ddot{\mathbf{x}}_{t,d}(t) \pm \ddot{\mathbf{x}}_s\,. \tag{30}$$

The switching point for this bang-bang solution (t_{sw}) is found from $t_{sw} = t$ when

$$\Delta\dot{x}^j(t) \;\;=\;\; -sgn(\Delta\dot{x}^j(t))\sqrt{2\,|\ddot{x}^j_s|\,|\Delta x^j(t)|}\;,$$

where $x^j(t)$ is the jth component of $\mathbf{x}$ at time t.

VI. FORCE DISTRIBUTION

When the feet are in double support, in order to compute the joint torques that are required to track the commanded motion, the right and left foot spatial contact forces ($\mathbf{f}_R$ and $\mathbf{f}_L$) need to be computed (see Eq. 10), in addition to the desired joint accelerations ($\boldsymbol{\alpha}$). The computed ZMP algorithm computes the net force $\mathbf{f}_{net}$ but not the individual foot forces (or ZMPs); therefore, this force should be distributed between the two feet to set the individual foot forces. This is an under-specified problem; therefore, we introduce extra constraints to solve it.

First, in order to minimize the interaction forces between the feet that might cause slippage, we assume that the foot forces are parallel to the net force [17]. That is, each foot force is set to a percentage η of the net force. This percentage is based on the distance from the foot ZMP (local ZMP) to the computed ZMP:

$$ {}^0\boldsymbol{f}_{zmp,k} = \eta_k \boldsymbol{f}_{net}\,, \tag{31} $$

$$ {}^0 n^z_{zmp,k} = \eta_k\, n^z_{net}, \text{ for } k \in \{R, L\} \tag{32} $$

and

$$ \eta_R = \left(\frac{d_L}{d_R + d_L} \right), \tag{33} $$

$$ \eta_L = \left(\frac{d_R}{d_R + d_L} \right) = 1 - \eta_R\,, \tag{34} $$

where

$$ d_k = || \, {}^0\boldsymbol{p}_{zmp,k} - {}^0\boldsymbol{p}_{zmp} \, ||\,. \tag{35} $$

Note that n^x_{net} and n^y_{net} are zero at the ZMP. This force, as represented in the ankle's frame, is computed as:

$$ {}^{A,k}\boldsymbol{f}_{zmp,k} = {}^{A,k}\boldsymbol{R}_0 \; {}^0\boldsymbol{f}_{zmp,k}\,, \tag{36} $$

where (A, k) is a coordinate system that is aligned with ankle k's joint axes and its origin is at the ankle joint origin. This local force has the following moment about the ankle's center:

$$ \boldsymbol{n}_{A,k} = {}^{A,k}\boldsymbol{p}_{zmp,k} \times {}^{A,k}\boldsymbol{f}_{zmp,k} + {}^{A,k}\boldsymbol{R}_0 \, {}^0\boldsymbol{n}_{zmp,k}\,, \tag{37} $$

where ${}^{A,k}\boldsymbol{p}_{zmp,k}$ is the position of the ZMP for foot k as represented in the ankle's frame, and it can be computed as:

$$ {}^{A,k}\boldsymbol{p}_{zmp,k} = {}^{A,k}\boldsymbol{R}_0 \; (\, {}^0\boldsymbol{p}_{zmp,k} - {}^0\boldsymbol{p}_{A,k} \,). \tag{38} $$

Second, we search for a solution that minimizes the roll and pitch ankle torques, to minimize ankle energy. This problem can be formulated as an optimization problem in which minimizing the roll ($n^x_{A,k}$) and pitch ($n^y_{A,k}$) ankle torques is the objective criteria (function). The first constraints are the foot boundaries. Satisfaction of the moment balance equations in the support plane about the line connecting the foot ZMPs, enforces an additional constraint that the foot ZMPs and overall ZMP must be on a line [17].

$$ \left({}^0\boldsymbol{p}_{zmp,R} - {}^0\boldsymbol{p}_{zmp} \right) \times \left({}^0\boldsymbol{p}_{zmp,L} - {}^0\boldsymbol{p}_{zmp} \right) = \mathbf{0}\,. \tag{39} $$

The following summarizes this solution:
$minimize \left(norm \left[n^x_{A,R} \; n^y_{A,R} \right] + norm \left[n^x_{A,L} \; n^y_{A,L} \right] \right)$
$subject\,to$:

1) ${}^0\boldsymbol{p}_{zmp,R}$ is within the right foot boundaries,
2) ${}^0\boldsymbol{p}_{zmp,L}$ is within the left foot boundaries, and
3) ${}^0\boldsymbol{p}_{zmp,R}$ and ${}^0\boldsymbol{p}_{zmp,L}$ satisfy the ZMP line constraint.

The above formulation provides a general solution for the force distribution problem. A simple geometric solution for the common case in which the feet are side by side can be computed by eliminating the roll ankle torque only. The solution is based on having each foot force vector intersect with its roll axis. Further, the line that connects the foot ZMPs and the overall ZMP is set parallel to the line between the ankle centers. The resulting equations are:

$$ {}^0 p^x_{zmp,k} = {}^0 p^x_{zmp}\,, \tag{40} $$

$$ {}^0 p^y_{zmp,k} = \frac{-{}^0 p^z_{A,k} \, f^y_{net}}{f^z_{net}} + {}^0 p^y_{A,k}\,. \tag{41} $$

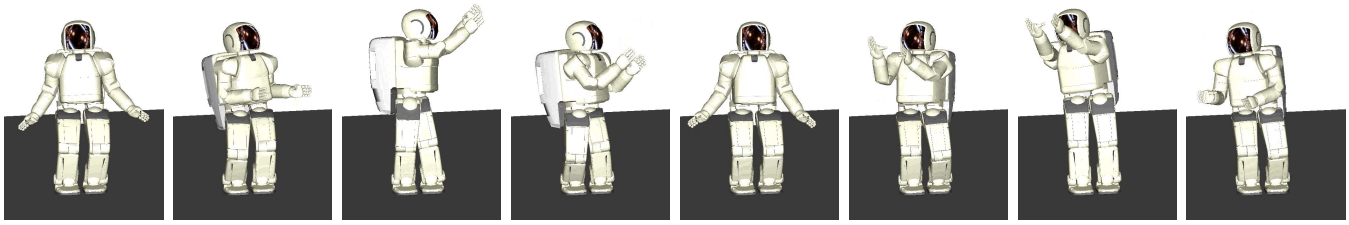
VII. RESULTS

The proposed control is applied to an ASIMO model [18] (see Fig. 1). It is simulated using `RobotBuilder`, which is a 3D graphical dynamics simulation package based on the *DynaMechs* library [19]. An approximately 45-second upper-body "reaching" motion from the CMU motion database is chosen as an input to the proposed control algorithm [20]. This motion is chosen because it is non-periodic, complex, and a somewhat fast motion. For this motion, we assume the feet are always in double support and their positions are chosen to be side by side with a hip-width distance between them.

First, we compute the lower-body joint positions by solving the inverse kinematics for the double support case. Figure 5 shows the computed ZMP for the entire motion for different speeds. The original speed of the motion (Fig. 5b) shows that the ZMP is outside the support for certain periods; thus the motion was infeasible and the RAC failed to track it. With a speed of 1.5 times faster than the original speed (Fig. 5a), the ZMP is outside the support for a longer period, and greater distance, due to the increase of the dynamic effects. On the other hand, with a speed of 1.5 times slower (Fig. 5c), the ZMP comes completely inside the support which shows that dynamic effects are what cause this motion to be infeasible. With 10 times slower speed (Fig. 5d), all the dynamic effects were eliminated; thus the projection of the CoM coincides with the ZMP.

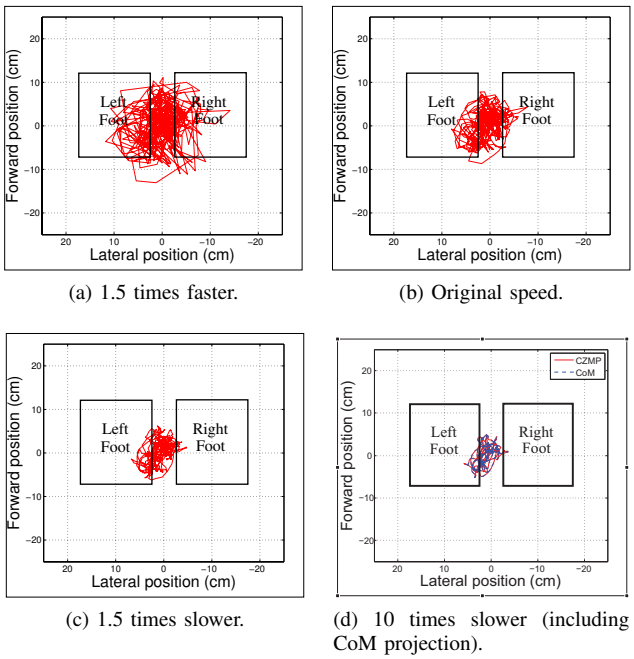
Figure 6 shows how MRAC modified the fore/aft and lateral torso position, as an example. The motion has been modified just after the 26th second when the computed ZMP was outside the support and the synchronization starts as soon as the computed ZMP returned to the support at t=26.5 sec. Snapshots of this portion of the motion are shown in Fig. 7. Note that similar results were obtained for faster speeds.

Figure 8 plots the actual and desired motion ZMPs while Fig 9 plots the actual and modified ZMPs. Using the MRAC, the motion was tractable and balanced. Note that the actual ZMP never leaves the support. The foot ZMPs for a portion of the motion are shown in Fig. 10. Each foot ZMP is



(a) $t = 26\,sec$ (b) $t = 26.25\,sec$ (c) $t = 26.5\,sec$ (d) $t = 26.75\,sec$ (e) $t = 27\,sec$ (f) $t = 28\,sec$ (g) $t = 29\,sec$ (h) $t = 29.5\,sec$

Fig. 7. Snapshots from the modified motion ($t = 26\,sec$ to $t = 29.5\,sec$).



(a) 1.5 times faster. (b) Original speed.

(c) 1.5 times slower. (d) 10 times slower (including CoM projection).

Fig. 5. Computed ZMP for different motion speeds of an upper-body "reaching" motion [20] that has been investigated in this paper.

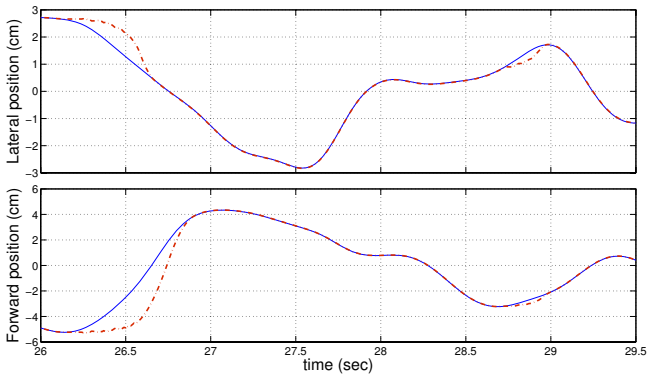


Fig. 6. Desired (solid) vs. modified (dashed) motion for a portion of the torso motion.

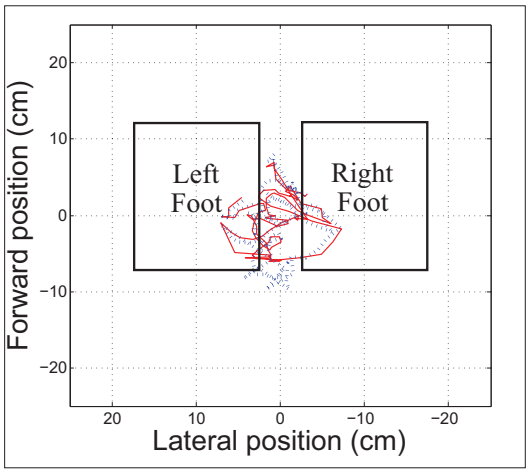


Fig. 8. Actual ZMP (solid) vs. desired motion ZMP (dotted) for a segment of the motion that has been modified.

located near the roll axis of the ankle. This is because the vertical (normal) component f_{net}^{z} of the net force $\boldsymbol{f}_{net}$ is the dominant force as shown in Fig. 11.

VIII. SUMMARY AND CONCLUSION

We implement a very efficient and effective dynamics filter which is applicable for general upper-body motion when standing in place. The dynamics filter is based on efficient inverse dynamics calculations. It involves modification of the motion of the torso so as to constrain the commanded motion to produce a ZMP which is inside the support. The feasible, commanded motion is synchronized to the desired reference motion when the computed ZMP for the reference motion again falls within the support. Resolved Acceleration Control is developed for a humanoid to track the desired/modified motion.

This algorithm produces a feasible motion that satisfies the foot constraints for highly dynamic upper-body task specifications. It may be extended to handle the situations when the projection of the CoM drifts outside the support due to the desired upper-body motion. This can be achieved by adding the CoM as a task-descriptor and constraining its projection to be inside the support. If the task is unreachable or cannot be executed, a step can be triggered.

Based on a desired upper-body task specification, computations of the upper-body and torso motion is currently achieved using a constrained closed loop inverse kinematics procedure. In the current implementation, the dynamic filter is decoupled from the upper-body motion generator and can potentially modify the desired torso trajectory, introducing upper-body task errors. In future implementations, dynamic filtering will be integrated with upper-body inverse kinematics in order to simultaneously satisfy upper-body task specifications as well as dynamic balance.

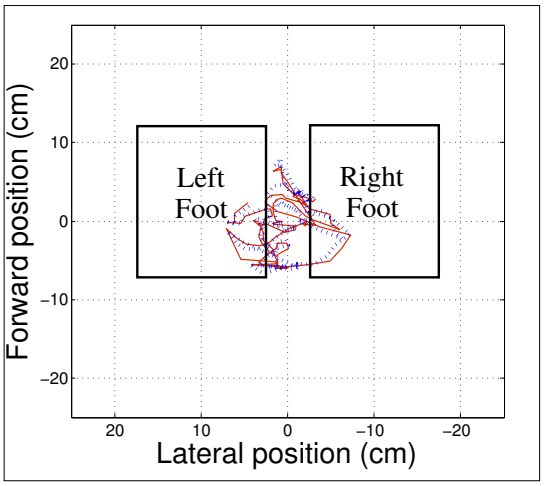


Fig. 9. Actual (solid) vs. modified (dotted) ZMP for a segment of the motion that has been modified.

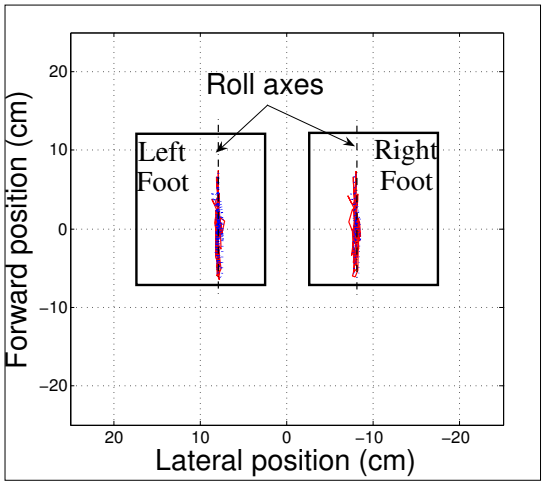


Fig. 10. Actual (solid) vs. modified (dotted) foot ZMPs for a segment of the motion that has been modified.

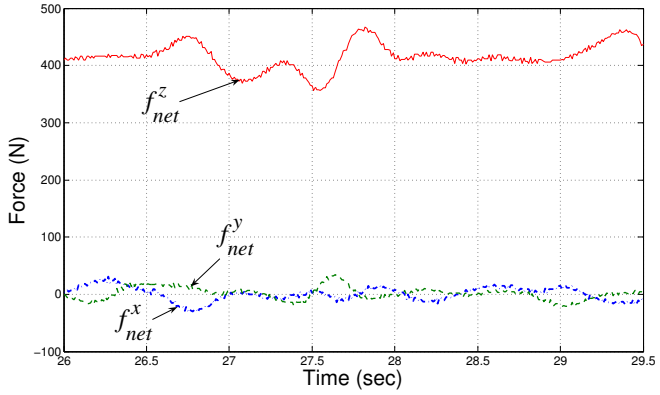


Fig. 11. Net force $\boldsymbol{f}_{net}$ for a segment of the motion that has been modified.

IX. ACKNOWLEDGMENTS

Part of the data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217. The authors wish to thank Victor Ng-Thow-Hing for use of his ALIVE graphics software.

REFERENCES

[1] C. Kemp, P. Fitzpatrick, H. Hirukawa, K. Yokoi, K. Harada, and Y. Matsumoto, "Chapter 56: Humanoids," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer, 2008.

[2] N. Naksuk, C. G. Lee, and S. Rietdyk, "Whole-body human-to-humanoid motion transfer," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, Tsukuba, Japan, December 2005, pp. 104–109.

[3] C. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, Daejeon, Korea, December 2008, pp. 399–405.

[4] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick, "Online transfer of human motion to humanoids," *International Journal of Humanoid Robotics*, vol. 6, pp. 265–289, 2009.

[5] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-body dynamic behavior and control of human-like robots," *Int. J. Humanoid Robotics*, vol. 1, no. 1, pp. 29–43, 2004.

[6] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proc. of Int. Conf. on Robotics and Automation (ICRA06)*, Orlando, FL, 2006.

[7] K. Yamane and Y. Nakamura, "Dynamics filter - concept and implementation of online motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 421–432, June 2003.

[8] R. Featherstone and D. Orin, "Chapter 2: Dynamics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer, 2008.

[9] R. Featherstone, *Robot Body Dynamics Algorithms*. New York-Boston: Springer, 2008.

[10] M. Vukobratović and D. Juricic, "Contributions to the synthesis of biped gait," *IEEE Transactions on Biomedical Engineering*, vol. 17, no. 1, pp. 25–36, 1969.

[11] M. Vukobratović and B. Borovac, "Zero-moment point – thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[12] J. Luh, M. Walker, and R. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, pp. 468–474, 1980.

[13] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Resolved-acceleration control of robot manipulators: A critical review with experiments," *Robotica*, vol. 16, no. 5, pp. 565–573, 1998.

[14] B. Siciliano and L. Villani, *Robot Force Control*. Norwell, MA: Kluwer Academic Publishers, 1999.

[15] W. Chung, L.-C. Fu, and S.-H. Hsu, "Chapter 6: Motion control," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer, 2008.

[16] L. M. Sonneborn and F. S. van Vleck, "The bang-bang principle for linear control systems," *Journal of the Society for Industrial and Applied Mathematics. Series A, On control*, vol. 2, no. 2, pp. 151–159, 1964.

[17] K. Waldron, "Force and motion management in legged locomotion," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 4, pp. 214–220, 1986.

[18] 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*, vol. 3, Lausanne, Switzerland, October 2002, pp. 2478–2483.

[19] S. McMillan, D. E. Orin, and R. B. McGhee, "DynaMechs: an object oriented software package for efficient dynamic simulation of underwater robotic vehicles," in *Underwater Robotic Vehicles: Design and Control*. Albuquerque, NM: TSI Press, 1995, pp. 73–98.

[20] Carnegie Mellon University, "CMU graphics lab motion capture database," Internet page, jump up to grab, reach for, tiptoe motion. [Online]. Available: http://mocap.cs.cmu.edu