Compliant Locomotion Using Whole-Body Control and Divergent Component of Motion Tracking

Michael A. Hopkins †, Dennis W. Hong ‡, and Alexander Leonessa †*

***Abstract*— This paper presents a compliant locomotion framework for torque-controlled humanoids using model-based whole-body control. In order to stabilize the centroidal dynamics during locomotion, we compute linear momentum rate of change objectives using a novel time-varying controller for the Divergent Component of Motion (DCM). Task-space objectives, including the desired momentum rate of change, are tracked using an efficient quadratic program formulation that computes optimal joint torque setpoints given frictional contact constraints and joint position / torque limits. In order to validate the effectiveness of the proposed approach, we demonstrate push recovery and compliant walking using THOR, a 34 DOF humanoid with series elastic actuation. We discuss details leading to the successful implementation of optimization-based whole-body control on our hardware platform, including the design of a "simple" joint impedance controller that introduces inner-loop velocity feedback into the actuator force controller.**

I. Introduction

Torque-controlled bipeds and quadrupeds are becoming increasingly prevalent as researchers attempt to mimic the speed and adaptability of locomotion behaviors found in nature [1–7]. This has created an interest in compliant locomotion strategies and inverse dynamics approaches that are robust to the unexpected forces and unmodeled dynamics encountered by hardware systems operating in real environments. The problem is further complicated in the field of humanoid robotics, where whole-body control is required to implement multi-objective behaviors such as maintaining balance while aiming a fire hose or operating a drill.

A number of researchers have proposed convex optimization techniques to solve the inverse dynamics and whole-body control problem subject to multi-contact constraints [8–15]. These approaches compute joint torque setpoints that minimize tracking errors for multiple motion tasks including desired momentum rates of change, end-effector accelerations, and joint accelerations relating to whole-body motions. In general, these formulations can serve as the basis for any locomotion, manipulation, or generic multi-contact behavior.

Implementing model-based whole-body control on real hardware platforms is exceptionally challenging due to numerous issues arising from communication delays, actuator stiction, mechanical bias, structural compliance, and a variety of electromechanical phenomena. In [11], the authors demonstrated successful balancing using hierarchical inverse dynamics on a purely torque-controlled biped. However, most implementations have required some form of joint-space position and / or velocity feedback to compensate for unmodeled dynamics. As an example, a hybrid inverse kinematics and inverse dynamics controller was implemented in [13] to achieve successful locomotion on the Atlas robot using PD feedback in conjunction with torque control.

Dynamic balancing is typically implemented through the design of feedback controllers to stabilize the robot's centroidal dynamics. This is made possible through control of the robot's centroidal momentum rate of change via direct optimization of external contact forces. Several authors implement Cartesian PD controllers to track desired center of mass (CoM) trajectories using linear momentum rate of change objectives [9, 11, 12]. In [15], the authors proposed an efficient quadratic program (QP) that minimizes a time-varying LQR cost function based on the Zero Moment Point (ZMP) dynamics subject to the inverse dynamics constraints.

In [16], the authors introduced the Capture Point (CP) transformation which allows the horizontal centroidal dynamics to be stabilized using a simple CMP-based control law [17, 18]. As shown in [19], this controller is equivalent to the best CoM-ZMP regulator [20], known to maximize the horizontal stability margins for CoM tracking. In [10], CP tracking was implemented using a momentum-based whole-body controller, and in [21], the authors introduced the three-dimensional Divergent Component of Motion (DCM), which constitutes an extension of the CP transformation that simplifies locomotion planning and control on uneven terrain.

This paper presents an implementation of optimization-based whole-body control and compliant locomotion on a new torque-controlled humanoid, THOR [22]. Inspired by the work of [10] and [11], we propose an efficient quadratic program formulation to solve the inverse dynamics problem given frictional contact constraints and joint position / torque limits. The centroidal dynamics are stabilized using a novel momentum controller based on the time-varying DCM dynamics, implemented for the first time on hardware. These components are assembled into a general framework for compliant locomotion on uneven terrain.

Successful whole-body control of the THOR hardware platform is achieved using a "simple" joint impedance controller that combines high performance torque control with low-gain velocity feedback. To improve the stability of the presented inverse dynamics-based approach, joint velocity setpoints are computed from the optimized joint accelerations and tracked using pre-transmission velocity estimates. The proposed framework is validated through push recovery and compliant walking experiments on the THOR platform.

†M. A. Hopkins and A. Leonessa are with the Terrestrial Robotics, Engineering & Controls Lab at Virginia Tech. (e-mail: michael.hopkins@vt.edu)

‡D. W. Hong is with the Robotics and Mechanisms Laboratory at UCLA.

*This material is based upon work supported by (while serving at) the National Science Foundation.

II. Divergent Component of Motion Tracking

In this section, we present an approach to stabilize the linear centroidal dynamics during locomotion by tracking a reference trajectory for the time-varying Divergent Component of Motion (DCM). The three-dimensional DCM, $\boldsymbol{\xi} = \mathbf{x} + \frac{1}{\omega_0}\dot{\mathbf{x}}$, is a linear transformation of the CoM state that divides the linear CoM dynamics into stable and unstable first-order components where $\omega_0 = \sqrt{\frac{g}{\Delta z}}$ represents the natural frequency of the linear inverted pendulum [21, 23]. For a constant CoM height, Δz, the horizontal projection of the DCM is equivalent to the two-dimensional Capture Point (CP) [16], defined as the point at which the Centroidal Moment Pivot (CMP) [24] must be placed at any time to allow the CoM to come to a complete rest. In [25], we proposed a time-varying extension of the three-dimensional DCM to simplify dynamic planning and control of vertical CoM trajectories. Here we briefly summarize those results.

A. Time-Varying DCM Dynamics

The time-varying Divergent Component of Motion is defined as

$$\boldsymbol{\xi} = \mathbf{x} + \frac{1}{\omega(t)}\dot{\mathbf{x}}, \tag{1}$$

where $\mathbf{x} = [\, x_{com}, y_{com}, z_{com}\,]^T$ is the CoM position, $\dot{\mathbf{x}}$ is the CoM velocity, and $\omega(t) > 0$ is the time-varying natural frequency of the DCM. Solving (1) for $\dot{\mathbf{x}}$ yields the first-order CoM dynamics,

$$\dot{\mathbf{x}} = \omega(\boldsymbol{\xi} - \mathbf{x}), \tag{2}$$

where we have introduced the relaxed notation, $\omega := \omega(t)$, omitting the natural frequency's explicit dependence on time. The CoM dynamics are asymptotically stable with respect to an equilibrium point at $\boldsymbol{\xi}$. Thus, the linear centroidal dynamics can be indirectly stabilized by regulating the DCM.

Differentiating (1) and substituting (2) gives

$$\begin{aligned}
\dot{\boldsymbol{\xi}} &= \dot{\mathbf{x}} - \frac{\dot{\omega}}{\omega^2}\dot{\mathbf{x}} + \frac{1}{\omega}\ddot{\mathbf{x}} \\
&= \left(1 - \frac{\dot{\omega}}{\omega^2}\right)\dot{\mathbf{x}} + \frac{1}{\omega}\ddot{\mathbf{x}} \\
&= \left(\omega - \frac{\dot{\omega}}{\omega}\right)(\boldsymbol{\xi} - \mathbf{x}) + \frac{1}{m\omega}\dot{\mathbf{l}} \\
&= \left(\omega - \frac{\dot{\omega}}{\omega}\right)\left(\boldsymbol{\xi} - \left(\mathbf{x} - \frac{\dot{\mathbf{l}}}{m\,(\omega^2 - \dot{\omega})}\right)\right),
\end{aligned}$$

where m is the total mass of the robot and $\dot{\mathbf{l}} \in \mathbb{R}^3$ is the linear momentum rate of change. Here we have substituted $\ddot{\mathbf{x}} = \frac{1}{m}\dot{\mathbf{l}}$ using Newton's second law. By defining the Virtual Repellent Point (VRP) [21] in terms of the CoM and linear momentum rate of change,

$$\mathbf{r}_{vrp} = \mathbf{x} - \frac{\dot{\mathbf{l}}}{m\,(\omega^2 - \dot{\omega})}, \tag{3}$$

we can derive first-order equations of motion for the DCM,

$$\dot{\boldsymbol{\xi}} = \left(\omega - \frac{\dot{\omega}}{\omega}\right)(\boldsymbol{\xi} - \mathbf{r}_{vrp})\,. \tag{4}$$

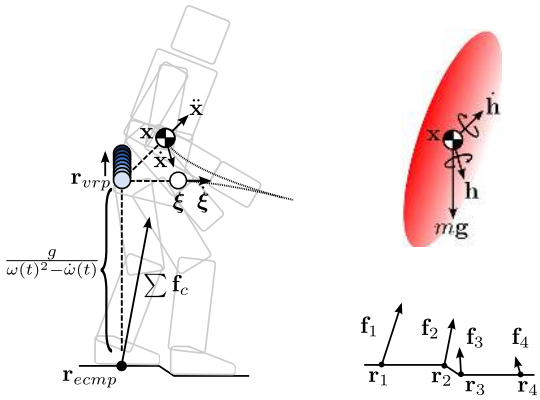

Fig. 1. Left: Time-varying DCM dynamics. Right: Centroidal dynamics.

First introduced in [21], the VRP represents the unstable equilibrium point of the DCM dynamics and can be viewed as a three-dimensional analog to the CMP. To see this, note that the linear momentum rate of change is given by $\dot{\mathbf{l}} = \sum \mathbf{f}_c - m\mathbf{g}$ where $\mathbf{g} = [\,0\;0\;g\,]^T$ is the gravity vector and $\mathbf{f}_c$ are the contact forces. Thus, we can express the VRP in terms of the external forces,

$$\mathbf{r}_{vrp} = \mathbf{x} - \frac{\sum \mathbf{f}_c - m\mathbf{g}}{m\,(\omega^2 - \dot{\omega})} = \mathbf{r}_{ecmp} + \frac{\mathbf{g}}{\omega^2 - \dot{\omega}}, \tag{5}$$

where

$$\mathbf{r}_{ecmp} = \mathbf{x} - \frac{\sum \mathbf{f}_c}{m\,(\omega^2 - \dot{\omega})} \tag{6}$$

is the time-varying extension of the enhanced Centroidal Moment Pivot (eCMP) [21].

The eCMP lies on the line passing through the CoM, parallel to the net contact force, $\sum \mathbf{f}_c$, and is equivalent to the CMP when it intersects the ground surface. As illustrated in Fig. 1, the horizontal position of the VRP is equivalent to the eCMP, while the vertical position varies depending on the natural frequency trajectory. If the eCMP lies in the base of support, it is possible to avoid generating a horizontal moment about the CoM by placing the center of pressure (CoP) at the eCMP. Thus, through appropriate planning of the eCMP and DCM reference trajectories, the linear centroidal dynamics can be stabilized without generating significant angular momentum during locomotion.

B. Time-Varying DCM Tracking

Assuming $\omega - \dot{\omega}/\omega > 0$, the time-varying DCM dynamics can be stabilized by defining an appropriate control law for the VRP. In [21], Englsberger et al. proposed a proportional tracking controller for the time-invariant DCM dynamics assuming $\omega(t) = \omega_0$, and in [19], Morisawa et al. introduced integral action into the DCM controller to compensate for steady-state errors due to model uncertainty.

We define the following control law based on the time-varying DCM dynamics (4),

$$\mathbf{r}_{vrp} = \boldsymbol{\xi} - \frac{1}{\omega - \frac{\dot{\omega}}{\omega}}\left(\dot{\boldsymbol{\xi}}_r + \mathbf{k}_\xi\,(\boldsymbol{\xi}_r - \boldsymbol{\xi}) + \mathbf{k}_\Xi \int (\boldsymbol{\xi}_r - \boldsymbol{\xi})\,dt\right),$$

where $\boldsymbol{\xi}_r$ and $\dot{\boldsymbol{\xi}}_r$ represent the reference DCM position and velocity [25]. The first term cancels the DCM drift dynamics, and the second term implements a proportional-integral controller with unity feedforward. The non-negative feedback gains $\mathbf{k}_\xi$ and $\mathbf{k}_\Xi$ determine the bandwidth and steady-state characteristics of the DCM controller.

III. Whole-Body Control

In this section we present a summary of humanoid dynamics and an overview of the proposed optimization-based whole-body controller implemented on the THOR platform.

A. Humanoid Dynamics

The configuration of an articulated humanoid with n actuated degrees of freedom (DOF) can be expressed by the vector, $\mathbf{q} = \begin{bmatrix} \mathbf{q}_0^T & \mathbf{q}_n^T \end{bmatrix}^T \in \mathbb{R}^{6+n}$, where $\mathbf{q}_0 \in \mathbb{R}^6$ encodes the 6 DOF translation and orientation of the floating-base frame and $\mathbf{q}_n \in \mathbb{R}^n$ represents the n DOF vector of actuated joint positions. The full rigid-body equations of motion are given by

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{H}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \sum_c \mathbf{J}_c^T \mathbf{f}_c, \tag{7}$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of actuated joint torques, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of centrifugal, Coriolis and gravity torques, $\mathbf{H}(\mathbf{q})$ is the joint-space inertia matrix, and $\mathbf{J}_c$ and $\mathbf{f}_c \in \mathbb{R}^3$ are the point Jacobians and corresponding reaction forces at each contact point, $\mathbf{r}_c \in \mathbb{R}^3$ [26]. Here we define the contact subscript, $c = 1 : N$, given N frictional contact points.

Alternatively, the centroidal dynamics define the reduced equations of motion for the center of mass, $\mathbf{x} = \begin{bmatrix} x_{com}, y_{com}, z_{com} \end{bmatrix}^T$, and linear and angular momentum of the system, $\mathbf{l} \in \mathbb{R}^3$ and $\mathbf{k} \in \mathbb{R}^3$, as illustrated in Fig. 1. The total momentum rate of change of the robot is given by

$$\dot{\mathbf{h}} = \begin{bmatrix} \dot{\mathbf{l}} \\ \dot{\mathbf{k}} \end{bmatrix} = \sum_c \mathbf{W}_c \mathbf{f}_c + \mathbf{w}_g, \tag{8}$$

where $\mathbf{W}_c \in \mathbb{R}^{6\times 3}$ maps contact forces to wrenches acting about the CoM and $\mathbf{w}_g = \begin{bmatrix} 0, 0, -mg, 0, 0, 0 \end{bmatrix}^T$ encodes the force of gravity [9, 10]. The momentum rate of change is related to the joint velocities and accelerations by the equality,

$$\dot{\mathbf{h}} = \dot{\mathbf{A}}\dot{\mathbf{q}} + \mathbf{A}\ddot{\mathbf{q}}, \tag{9}$$

where $\mathbf{A}$ represents the centroidal momentum matrix [27].

B. Task-Space Formulation

Arbitrary motion tasks such as the acceleration of a Cartesian frame or contact point can be expressed as

$$\dot{\mathbf{v}}_t = \dot{\mathbf{J}}_t \dot{\mathbf{q}} + \mathbf{J}_t \ddot{\mathbf{q}}, \tag{10}$$

where $\mathbf{J}_t$ is the associated task-space Jacobian matrix. In this work, we model contact points using Coulomb friction and linear acceleration constraints, i.e. $\dot{\mathbf{v}}_c = 0$. Tipping and slipping of supporting surfaces is avoided by ensuring that reaction forces at each contact point remain inside the corresponding friction cone, i.e. $\mathbf{f}_c \in \mathcal{C}_c$ [28], as illustrated in Fig. 2.

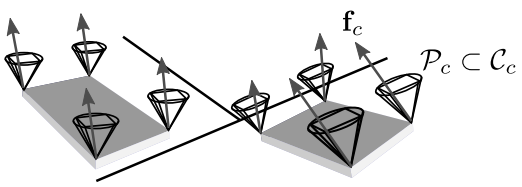

Fig. 2. Friction cone and friction pyramid approximations for foot contacts.

As in [9, 10, 15, 29], we employ a polyhedral approximation of the friction cone, $\mathcal{P}_c \subset \mathcal{C}_c$, at each contact point, $c = 1 : N$, allowing contact forces to be computed using a linear generating function,

$$\mathbf{f}_c = \boldsymbol{\beta}_c \boldsymbol{\rho}_c. \tag{11}$$

Here the columns of $\boldsymbol{\beta}_c \in \mathbb{R}^{3\times 4}$ are unilateral bases that span the volume of admissible forces in inertial coordinates, and $\boldsymbol{\rho}_c \in \mathbb{R}^4$ is a vector of non-negative coefficients to be determined via the whole-body controller.

C. Model-Based Optimization (Quadratic Program)

Given multiple motion tasks, $\dot{\mathbf{v}}_t$, the goal of whole-body control is to compute joint torques, $\boldsymbol{\tau}$, that minimize the tracking error for each task while satisfying dynamic constraints including available control authority, range of motion, and limits on the frictional contact forces. Motion tasks may include desired momentum rates of change, spatial accelerations of coordinate frames attached to individual links, or joint-space accelerations computed using an arbitrary high-level controller. Inspired by the work of [10] and [11], the proposed whole-body controller optimizes desired joint accelerations, $\ddot{\mathbf{q}}$, and generalized contact forces, $\boldsymbol{\rho} = \begin{bmatrix} \boldsymbol{\rho}_1^T & \dots & \boldsymbol{\rho}_N^T \end{bmatrix}^T$, using a linearly constrained quadratic program (QP) in the form,

$$\min_{\ddot{\mathbf{q}}, \boldsymbol{\rho}} \left\| \mathbf{C}_b \left(\mathbf{b} - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}\ddot{\mathbf{q}} \right) \right\|^2 + \lambda_{\ddot{\mathbf{q}}} \|\ddot{\mathbf{q}}\|^2 + \lambda_{\boldsymbol{\rho}} \|\boldsymbol{\rho}\|^2 \tag{12}$$

subject to

$$\dot{\mathbf{A}}\dot{\mathbf{q}} + \mathbf{A}\ddot{\mathbf{q}} = \sum_c \mathbf{W}_c \mathbf{f}_c + \mathbf{w}_g \tag{13}$$

$$\underline{\mathbf{q}} \le \mathbf{q} + T\dot{\mathbf{q}} + \frac{1}{2}T^2\ddot{\mathbf{q}} \le \bar{\mathbf{q}} \tag{14}$$

$$\underline{\boldsymbol{\tau}} \le \boldsymbol{\tau} \le \bar{\boldsymbol{\tau}} \tag{15}$$

$$\mathbf{0} \le \boldsymbol{\rho}, \tag{16}$$

where $\mathbf{b}$ is the vector of desired motion tasks and $\mathbf{J}$ is the corresponding matrix of stacked Jacobians, i.e.

$$\mathbf{b} = \begin{bmatrix} \dot{\mathbf{h}}_d \\ \ddot{\mathbf{q}}_d \\ \dot{\mathbf{v}}_{d_1} \\ \vdots \end{bmatrix}, \qquad \mathbf{J} = \begin{bmatrix} \mathbf{A} \\ \boldsymbol{I} \\ \mathbf{J}_1 \\ \vdots \end{bmatrix}. \tag{17}$$

The QP cost function and constraint equations are described in the subsections below.

D. Optimization Costs

The cost function (12) is designed to minimize the weighted quadratic error of the task-space objectives, $\mathbf{b}$, given a semi-positive definite weighting matrix, $\mathbf{Q}_b = \mathbf{C}_b^T \mathbf{C}_b$. Joint acceleration and contact force regularization terms are also included to ensure that the QP is strictly convex given $\lambda_{\ddot{\mathbf{q}}},\ \lambda_{\boldsymbol{\rho}} > 0$. The weighting matrix allows soft prioritization of motion tasks depending on the current behavior. High weights are assigned to contact acceleration objectives to approximate no slip conditions, thereby ensuring contact points remain stationary relative to support surfaces, while low weights are often assigned to angular momentum rate of change objectives to improve dynamic stability by permitting large restoring forces and accelerations that induce moments about the CoM. Although this approach does not allow a strict hierarchy of task prioritizations as in [11, 12], we have found that highly-weighted costs can be more forgiving than hard constraints in scenarios where one or more tasks are ill-conditioned.

E. Optimization Constraints

Constraints (13)-(16) are linear in the decision variables and ensure admissibility of the optimized joint accelerations and contact forces. Equation (13) follows from (8) and (9) and enforces the centroidal dynamic constraints [10]. The final inequality (16) enforces Coulomb friction constraints at each contact point. As proposed in [14], joint range of motion constraints are implemented using (14) where $\underline{\mathbf{q}}$ and $\bar{\mathbf{q}}$ represent the joint position limits and T represents a time constant determining the maximum rate of convergence to either limit. For the experiments presented in Section V we use $T \approx 0.15$ s.

A number of QP formulations include joint torques, $\boldsymbol{\tau}$, as additional decision variables to permit explicit torque limits in the optimization [12–14]. In these approaches, the full rigid body equations of motion (7) are included as linear equality constraints. As discussed in [11], it is possible to eliminate these additional decision variables and constraints noting that the torque vector is a linear function of the joint acceleration and contact force vectors. From (7) and (11), we have

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \mathbf{H}(\mathbf{q}) & -\sum_c \mathbf{J}_c^T \boldsymbol{\beta}_c \mathbf{S}_c \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\rho} \end{bmatrix} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), \quad (18)$$

where $\mathbf{S}_c$ is a selection matrix that projects $\boldsymbol{\rho}$ to $\boldsymbol{\rho}_c$. Joint torque limits are implemented by substituting the lower n equations of (18) into (15) where $\underline{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{\tau}}$ represent lower and upper torque limits. Admissibility of the joint torques, joint accelerations, and contact forces is ensured by the centroidal dynamics constraint (13). This is a departure from [11] where the floating-base dynamics are constrained by the upper 6 equations of (7); however, this is an equivalent formulation that also constrains the net momentum rate of change.

F. Implementation

At each time step, admissible joint accelerations, $\ddot{\mathbf{q}}_a$, and contact forces, $\boldsymbol{\rho}_a$, are computed using the proposed model-based optimization (12). Corresponding joint torque setpoints, $\boldsymbol{\tau}_a$, are computed from (18). The QP is solved using an Eigen implementation of QuadProg++ which is based on the active set method described in [30]. The optimization runs at 800 Hz on an i7 processor with 8 active contact points and a 30 DOF rigid body model. This enables real-time whole-body control for tasks such as standing manipulation and dynamic walking.

IV. Compliant Locomotion

This section provides an overview of the proposed locomotion framework developed using the presented whole-body controller. For reference, a high-level block diagram is shown in Fig. 3. Dynamic behaviors such as standing and stepping are implemented using state machines that respond to external events such as toe-off and heel-strike. Following each event, state-specific parameters such as desired end-effector waypoints and step durations are passed to planning modules to generate joint-space and task-space trajectories for dynamic locomotion.

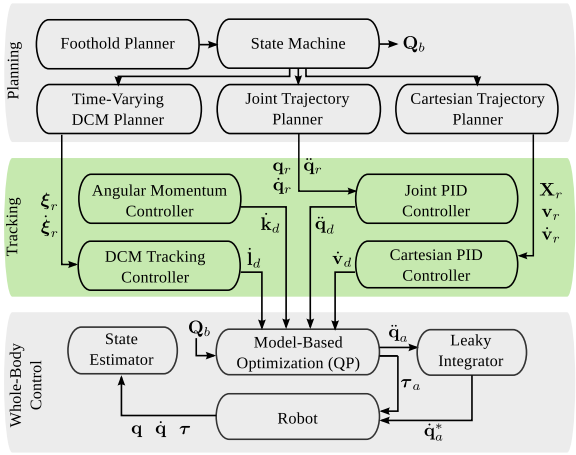

Fig. 3. Control block diagram of locomotion subsystem.

At each time step, desired motion tasks are computed using DCM, angular momentum, end-effector, and joint-space tracking controllers. Optimal joint torques are computed using the whole-body controller framework presented in Section III, and compliant joint-space control is implemented to track the resulting joint trajectories. Finally, a Kalman filter-based state estimator is implemented to compute the necessary states for whole-body control. The following subsections present a detailed overview of the compliant standing and stepping behaviors implemented using the proposed framework. Note that the presented approach can be applied to a large variety of tasks including climbing, fall recovery, and whole-body manipulation.

A. Standing

The stand controller enables the robot to maintain balance in double or single support while tracking upper body joint trajectories for generic manipulation tasks.

1) Momentum Control: Dynamic balancing is implemented by tracking a DCM reference trajectory, $\boldsymbol{\xi}_r(t)$, that nominally lies above the center of the support polygon. At each time step, the desired VRP position is computed using the DCM tracking controller defined in Section II-B. The linear momentum rate of change is then derived from the VRP definition (3), and the angular momentum rate of change is computed using a simple damping controller such that

$$\dot{\mathbf{h}}_d = \begin{bmatrix} \dot{\mathbf{l}}_d \\ \dot{\mathbf{k}}_d \end{bmatrix} = \begin{bmatrix} m\left(\omega^2 - \dot{\omega}\right)\left(\mathbf{x} - \mathbf{r}_{vrp}\right) \\ -b_{\mathbf{k}}\mathbf{k} \end{bmatrix}. \quad (19)$$

The damping coefficient, $b_{\mathbf{k}} \geq 0$, regulates the total angular momentum in the system.

2) Upper Body Control: Joint-space acceleration objectives are used to track upper-body joint trajectories. Joint tracking is achieved using a standard PD controller in the form,

$$\ddot{\mathbf{q}}_d = \ddot{\mathbf{q}}_r + k_{\mathbf{q}}\left(\mathbf{q}_r - \mathbf{q}\right) + b_{\mathbf{q}}\left(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}\right), \quad (20)$$

where $\mathbf{q}_r$, $\dot{\mathbf{q}}_r$, and $\ddot{\mathbf{q}}_r$ are the reference joint position, velocity, and acceleration vectors. The PD gains, $k_{\mathbf{q}} \geq 0$ and $b_{\mathbf{q}} \geq 0$, can be adjusted online depending on the current task. The reference trajectories may be specified by an external planner to allow standing manipulation, in which case high optimization weights are assigned to support accurate tracking. Low optimization weights may also be assigned to allow the arms to assist in balancing by generating angular momentum and / or shifting the center of mass.

3) Lower Body Control: Pelvis and swing foot trajectories are generated using piecewise 5$^{\text{th}}$ order minimum jerk polynomials based on the desired lower body motion. The state of each Cartesian frame is expressed by a transform $\mathbf{X}_r = \begin{bmatrix} \mathbf{R}_r & \mathbf{r}_r \\ \mathbf{0} & 1 \end{bmatrix}$, twist $\mathbf{v}_r = \begin{bmatrix} \dot{\mathbf{r}}_r^T & \boldsymbol{\omega}_r^T \end{bmatrix}^T$, and spatial acceleration $\dot{\mathbf{v}}_r = \begin{bmatrix} \ddot{\mathbf{r}}_r^T & \dot{\boldsymbol{\omega}}_r^T \end{bmatrix}^T$. The 6 DOF trajectories are tracked using a Cartesian PD controller in the form,

$$\dot{\mathbf{v}}_d = \begin{bmatrix} \ddot{\mathbf{r}}_d \\ \dot{\boldsymbol{\omega}}_d \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{r}}_r + \mathbf{K}_{\mathbf{r}}\left(\mathbf{r}_r - \mathbf{r}\right) + \mathbf{B}_{\mathbf{r}}\left(\dot{\mathbf{r}}_r - \dot{\mathbf{r}}\right) \\ \dot{\boldsymbol{\omega}}_r + \mathbf{K}_{\mathbf{R}}\left(\theta\hat{\mathbf{e}}\right) + \mathbf{B}_{\mathbf{R}}\left(\boldsymbol{\omega}_r - \boldsymbol{\omega}\right) \end{bmatrix}, \quad (21)$$

where $\theta\hat{\mathbf{e}}$ is the axis-angle representation of the rotational error. Here $\mathbf{K_r}$, $\mathbf{K_R}$, $\mathbf{B_r}$, and $\mathbf{B_R}$ are diagonal stiffness and damping matrices. Note that the pelvis linear accelerations are uncontrolled to avoid over-constraining the optimization.

B. Stepping

The step controller implements single-step and multi-step behaviors given desired foothold poses and step durations from a high-level footstep planner.

1) Trajectory Planning: The DCM reference trajectory is updated at the onset of each step using the time-varying DCM planner described in [25]. First, explicit CoP and vertical CoM trajectories are computed from the desired foothold poses. Next, reverse-time integration is used to plan a dynamically feasible DCM trajectory over a multi-step time horizon satisfying the final boundary conditions. Finally, Model Predictive Control is used to compute an admissible reference trajectory over a short preview window satisfying the initial boundary conditions.

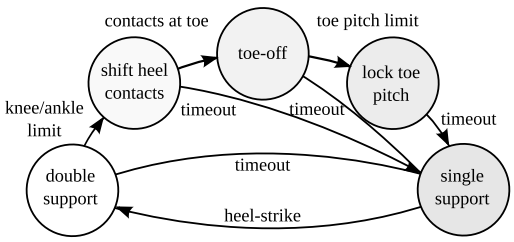

Fig. 4. Step controller state machine.

2) State Machine: The step behavior employs the same subset of controllers as the stand behavior. However, an additional finite state machine is implemented to handle transitions between various contact phases as illustrated in Fig. 4. During the double support phase, both feet remain in contact with the support surface. Eight active contact constraints are added to the whole-body controller optimization corresponding to the four corners of each foot. Following the preplanned double support duration, the state machine transitions to the single support phase, breaking contact with the swing foot and configuring appropriate optimization weights for swing foot tracking. Heel-strike events are detected using a force-torque sensor in the sole of the swing foot. When the swing foot is sufficiently loaded, the state machine transitions back to double support.

In order to increase the maximum stride length, we adopt a reactive toe-off strategy to compensate for range of motion limits. If the measured knee or ankle pitch of the swing leg reaches a soft position limit during double support, the two contact points on the heel of the swing foot are shifted to the toe in order to appropriately bound the center of pressure. Next the heel contacts are disabled to enable toe-off by allowing the foot to pivot about the two toe contacts. If the toe pitch reaches a critical limit, the swing foot tracking controller is configured to prevent the foot from rotating further during the double support phase.

C. Low-level control

The proposed whole-body locomotion framework was tested using THOR, a 1.8 m tall, 60 kg compliant humanoid designed as part of the DRC (DARPA Robotics Challenge) and ONR SAFFiR (Shipboard Autonomous Firefighting Robot) projects [22]. The THOR lower body is equipped with series elastic actuators and custom motor controllers for low-gain impedance control [31]. The upper body is equipped with stiff actuators for high-gain velocity control.

1) Leaky Integrator: We employ a similar approach to [32] to compute joint velocity setpoints, $\dot{\mathbf{q}}_a^*$, using leaky integration of the optimized joint accelerations, $\ddot{\mathbf{q}}_a$. The leaky integrator dynamics are expressed by the first-order ODE,

$$\ddot{\mathbf{q}}_a^* = \alpha\left(\dot{\mathbf{q}} - \dot{\mathbf{q}}_a^*\right) + \ddot{\mathbf{q}}_a, \quad (22)$$

where $\dot{\mathbf{q}}_a^* = \int \ddot{\mathbf{q}}_a^*\, dt$. The leaking rate, $\alpha \geq 0$, determines the rate at which the integral drifts towards the estimated velocity. In the experiments presented in the following section, we use $\alpha = 75$ for the lower body joints and $\alpha = 0$ for

the upper body joints to account for the different actuator impedances.

2) Joint Control: The lower body joint setpoints are tracked using joint torque control with low-gain velocity feedback. Joint velocity feedback tends to increase the overall stability of the whole-body controller given the unmodeled dynamics inherent in real systems. On our platform, linear series elastic actuators are used to generate net torques about the robot's joints via linear to rotary and parallel mechanisms [33, 34]. The whole-body controller torques, $\boldsymbol{\tau}_a$, and integrated velocity setpoints, $\dot{\mathbf{q}}_a^*$, are updated at 150 Hz and relayed to embedded joint controllers with a sample rate of 2 kHz. Actuator force control is implemented using PID feedback with unity feedforward and a model-based disturbance observer based on spring force measurements from in-line load cell sensors.

The authors experimented with two variations of "simple" joint impedance control. The first controller introduces outer-loop damping based on joint velocity estimates from post-transmission absolute encoders at the joints. The desired actuator forces are given by

$$\mathbf{f}_a = \mathbf{J}_{\boldsymbol{\ell}}^{-T} \left(\boldsymbol{\tau}_a + b_{\mathbf{q}} \left(\gamma \dot{\mathbf{q}}_a^* - \dot{\mathbf{q}} \right) \right) \tag{23}$$

where $\mathbf{J}_{\boldsymbol{\ell}}$ is the mechanism Jacobian that maps angular joint velocities, $\dot{\mathbf{q}}$, to linear actuator velocities, $\dot{\boldsymbol{\ell}}$. The joint damping coefficient $b_{\mathbf{q}} \geq 0$ determines the velocity feedback gain, and the scalar $\gamma \in [0, 1]$ biases the joint velocity setpoint towards zero in order to improve the stability of the derivative action. If $b_{\mathbf{q}} = 0$, (23) reduces to a simple torque controller. If $\gamma = 0$, the velocity feedback term introduces viscous damping.

The second controller introduces inner-loop damping based on actuator velocity estimates from pre-transmission incremental encoders at the motors, i.e.

$$\mathbf{f}_a = \mathbf{J}_{\boldsymbol{\ell}}^{-T} \boldsymbol{\tau}_a + b_{\boldsymbol{\ell}} \left(\gamma \dot{\boldsymbol{\ell}}_a^* - \dot{\boldsymbol{\ell}} \right). \tag{24}$$

In this scenario, joint-space damping is indirectly achieved by tracking desired linear actuator velocities computed via forward kinematics, $\dot{\boldsymbol{\ell}}_a^* = \mathbf{J}_{\boldsymbol{\ell}} \dot{\mathbf{q}}_a^*$. Although the effective joint damping varies depending on the configuration, this approach significantly improves the stability of the derivative action, presumably due to the collocation of the motor and velocity sensor. In the experiments presented in the following section, we chose $b_{\boldsymbol{\ell}} = 10000$ N/m/s and $\gamma = 0.625$.

V. Experimental Results

This section presents experimental results for the proposed locomotion framework implemented on the THOR hardware platform [22]. Seen in Fig. 5, THOR features joint position and velocity sensors for each actuated DOF and an Attitude and Heading Reference Sensor (AHRS) mounted to the pelvis to provide inertial orientation, angular velocity, and linear acceleration measurements. A simple state estimator is employed to estimate the floating-base pose and twist using a combination of leg kinematics and inertial measurements. The locomotion framework utilizes a 30 DOF rigid body model of the robot obtained from detailed CAD. The state estimator and locomotion controllers operate at 150 Hz on an i7 computer mounted in the chest.

TABLE I

Whole-Body Controller Weights and Gains

Motion Task	Units	Weight	Stiffness	Damping
$\dot{\mathbf{l}}$	N	10	-	-
$\dot{\mathbf{k}}$	Nm	1	0	0
$\ddot{\mathbf{q}}_{\text{arms}}$	rad/s^2	6	15	10
$\ddot{\mathbf{q}}_{\text{waist}}$	rad/s^2	100	40	20
$\ddot{\mathbf{r}}_{\text{contact points}}$	m/s^2	10000	0, 0, 0	0, 0, 0
$\ddot{\mathbf{r}}_{\text{swing foot}}$	m/s^2	1000	150, 150, 200	20, 20, 20
$\dot{\boldsymbol{\omega}}_{\text{swing foot}}$	rad/s^2	1000	100, 100, 100	10, 10, 20
$\dot{\boldsymbol{\omega}}_{\text{pelvis}}$	rad/s^2	30	70, 70, 30	30, 30, 15

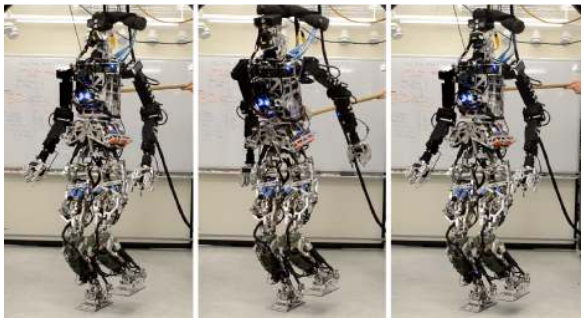

Fig. 5. THOR balancing on one foot. DCM tracking is used to stabilize the CoM dynamics following a disturbance impulse applied to the back.

A comprehensive list of whole-body controller weights and feedback gains is provided in Table I. This set of parameters was chosen experimentally to support standing and stepping behaviors. Although unlisted, integral action was included in the swing foot position, swing foot orientation, and pelvis orientation controllers with integral gains of diag (100, 100, 100), diag (100, 100, 100), and diag (50, 50, 30), respectively. For the experiments presented below, the DCM proportional gain was set to $k_{\boldsymbol{\xi}} = 2.5$ in the horizontal axes and $k_{\boldsymbol{\xi}} = 7.5$ in the vertical axis. We found that a higher stiffness in the vertical axis was necessary to achieve comparable performance due to the effect of gravity. The DCM integral gain was set to $k_{\boldsymbol{\Xi}} = 0.75$ in order to compensate for steady-state errors.

A. Balancing

Fig. 5 shows the robot balancing on one foot following a disturbance force applied to the back. The robot is able to recover from the unexpected disturbance by pitching its hip and shoulder joints in order to generate the necessary reaction forces to stabilize the DCM. This is made feasible by compliant joint control in the lower body. Low damping coefficients allow the leg joints to accelerate immediately in response to the impulse without shifting the CoP to the edge of the foot.

The controller setpoints and estimates for two consecutive pushes are shown in Fig. 6. The x-axis of the inertial frame is oriented to the front of the robot and the y-axis is oriented to the left. The region between the virtual toe and heel contacts is marked for reference. The whole-body controller

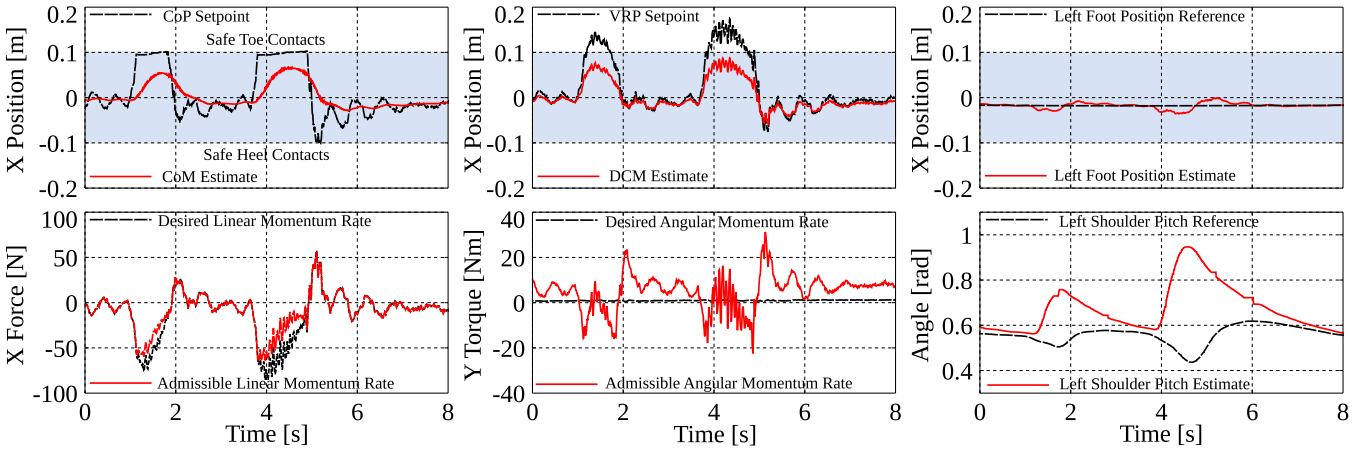

Fig. 6. Controller response following two disturbance impulses applied to the back of the robot while balancing on the right foot. Here the x-axis is oriented to the front of the robot and the y-axis is oriented to the left. Admissible values correspond to the outputs of the whole-body QP optimization.

formulation ensures that the optimized CoP setpoint does not exceed this safety region, thereby preventing the support foot from tipping by satisfying the ZMP criterion. The VRP setpoint, however, is allowed to leave the base of support in order to stabilize the DCM dynamics.

Note that, given the relatively low weight of the arm acceleration objectives, the left shoulder pitch joint diverges significantly from the desired reference trajectory to generate the necessary spin angular momentum. In this scenario, the shoulder reference trajectory is determined by a heuristic based on the rotation of the hip joint. Contrarily, the high weights of the swing foot acceleration objectives allow accurate reference tracking throughout the disturbance. Once the linear dynamics are stabilized, the pelvis and arm objectives dominate, allowing the robot to return to its original pose.

B. Walking

Fig. 7 shows the desired and estimated DCM and VRP signals for continuous walking with a step duration of 3.5 s, stride length of 0.175 m, and swing foot apex of 0.09 m. The robot is able to accurately track the DCM and swing foot reference trajectories on flat terrain. Although the proposed framework is designed to support dynamic locomotion on uneven terrain as discussed in [25], we are currently limited by the available knee torque in the THOR platform. We are in the process of performing the necessary hardware upgrades to support rough terrain locomotion.

C. Robustness to Unmodeled Terrain

The use of compliant joint control in the lower body results in behaviors that are inherently robust to uncertain and unstable terrain. Fig. 8 shows the robot standing and stepping onto various terrain, initially modeled as a flat surface. In the leftmost image, the robot stands on a balance board that rotates about a pivot, and in the images to the right, the robot steps onto unexpected debris. In each scenario, the controller is able to safely adapt to uncertain terrain using low-impedance task-space feedback. Note that the control gains and optimization weights are identical in each trial.

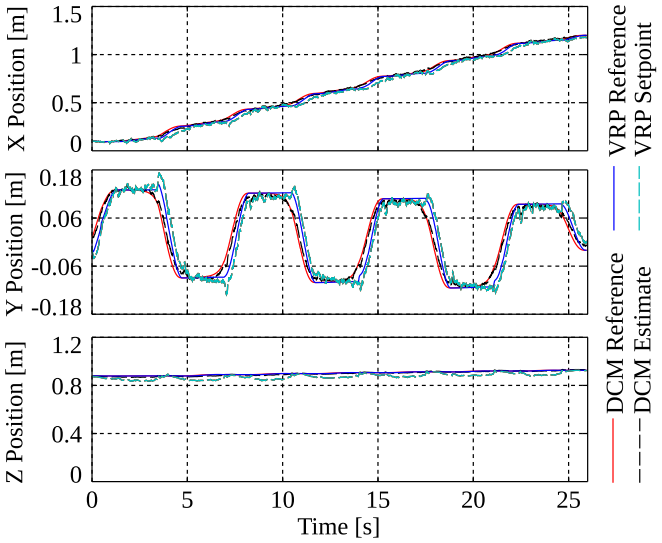

Fig. 7. Estimated and reference DCM and VRP trajectories while walking on flat terrain using the THOR hardware platform.

VI. Conclusion

This paper demonstrated successful balancing and walking on a torque-controlled humanoid using optimization-based whole-body control and a momentum controller based on the time-varying Divergent Component of Motion. The introduction of low-gain joint velocity feedback using leaky integration of the optimized joint accelerations enabled accurate tracking of the task-space objectives using compliant joint-space control. Future research efforts will focus on the application of adaptive control algorithms for dynamic step recovery. We are also investigating improved joint-space torque control approaches using a novel model for linear series elastic actuators.

Acknowledgments

This material is supported by ONR through grant N00014-11-1-0074 and by DARPA through grant N65236-12-1-1002.

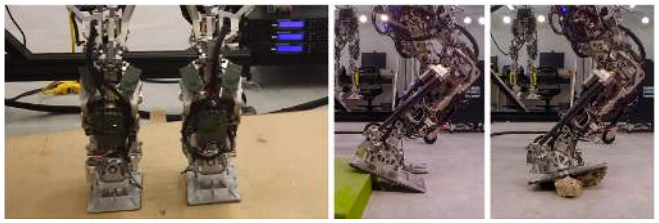

Fig. 8. Left: THOR standing on a balance board that rotates about a pivot. Right: Stepping onto unexpected debris and adapting to the terrain.

We would like to thank Derek Lahr, Viktor Orekhov, and Robert Griffin for their discussions on dynamics and controls. We would also like to thank Bryce Lee, Coleman Knabe, Steve Ressler, Jacob Webb, Jack Newton, and the remaining team members who contributed to the design of THOR.

REFERENCES

[1] J. Pratt and B. Krupp, "Design of a bipedal walking robot," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6962, Apr. 2008.

[2] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schffer, "Overview of the torque-controlled humanoid robot TORO," in *Humanoid Robots (Humanoids), 14th IEEE-RAS International Conference on*, Nov 2014.

[3] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ - a hydraulically and electrically actuated quadruped robot," *Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[4] M. Hutter, C. Gehring, M. Bloesch, M. Hoepflinger, C. D. Remy, and R. Siegwart, "StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *15th International Conference on Climbing and Walking Robot (CLAWAR)*, 2012.

[5] N. Tsagarakis, S. Morfey, G. Cerda, L. Zhibin, and D. Caldwell, "COMpliant huMANoid COMAN: Optimal joint stiffness tuning for modal frequency control," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2013, pp. 673–678.

[6] M. Slovich, N. Paine, K. Kemper, A. Metger, A. Edinger, J. Weber, and L. Sentis, "Building HUME: A bipedal robot for human-centered hyper-agility," in *Dynamic Walking Meeting*, 2012.

[7] D. Lahr, V. Orekhov, B. Lee, and D. Hong, "Development of a parallely actuated humanoid, SAFFiR," in *ASME International Design Engineering Technical Conference*, 2013.

[8] M. de Lasa and A. Hertzmann, "Prioritized optimization for task-space control," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Oct 2009, pp. 5755–5762.

[9] S.-H. Lee and A. Goswami, "Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Oct 2010, pp. 3157–3162.

[10] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-L. Ho Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt, "Summary of team IHMC's virtual robotics challenge entry," in *Humanoid Robots (Humanoids), 13th IEEE-RAS International Conference on*, Oct 2013.

[11] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid," *arXiv preprint arXiv:1305.2042*, 2013.

[12] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2013, pp. 3103–3109.

[13] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the Atlas robot," in *Humanoid Robots (Humanoids), 14th IEEE-RAS International Conference on*, Nov 2014.

[14] L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 346–362, April 2013.

[15] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2014.

[16] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Humanoid Robots (Humanoids), 6th IEEE-RAS International Conference on*, Dec 2006, pp. 200–207.

[17] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, Aug 2012.

[18] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on Capture Point dynamics," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Sept 2011, pp. 4420–4427.

[19] M. Morisawa, S. Kajita, F. Kanehiro, K. Kaneko, K. Miura, and K. Yokoi, "Balance control based on Capture Point error compensation for biped walking on uneven terrain," in *Humanoid Robots (Humanoids), 12th IEEE-RAS International Conference on*, Nov 2012, pp. 734–740.

[20] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2009, pp. 1966–1971.

[21] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-dimensional bipedal walking control using Divergent Component of Motion," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Nov 2013, pp. 2600–2607.

[22] B. Lee, "Design of a humanoid robot for disaster response," Master's thesis, Virginia Polytechnic Institute and State University, April 2014.

[23] J. Englsberger, T. Koolen, S. Betrand, J. Pratt, C. Ott, and A. Albu-Schaffer, "Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Sept 2014.

[24] M. B. Popovic and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *Int. J. Robot. Res*, vol. 24, no. 12, p. 10131032, 2005.

[25] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying Divergent Component of Motion," in *Humanoid Robots (Humanoids), 14th IEEE-RAS International Conference on*, Nov 2014.

[26] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer Berlin, 2008, vol. 49.

[27] D. Orin and A. Goswami, "Centroidal momentum matrix of a humanoid robot: Structure and properties," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Sept 2008, pp. 653–659.

[28] Y. Abe, M. da Silva, and J. Popovic, "Multiobjective control with frictional contacts," in *Symposium on Computer Animation*, M. Gleicher and D. Thalmann, Eds. Eurographics Association, 2007, pp. 249–258.

[29] C. Ott, M. Roa, and G. Hirzinger, "Posture and balance control for biped robots based on contact force optimization," in *Humanoid Robots (Humanoids), 11th IEEE-RAS International Conference on*, Oct 2011, pp. 26–33.

[30] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, 1983.

[31] C. Knabe, B. Lee, V. Orekhov, and D. Hong, "Design of a compact, lightweight, electromechanical linear series elastic actuator," in *ASME International Design Engineering Technical Conference*, 2014.

[32] S. Bertrand, T. Wu, and J. Pratt, "Momentum-based control framework and capturibility-based walking control - application to the humanoid robot Atlas." Presented at the Institute for Human and Machine Cognition (IHMC), Pensacola, Florida, March 14, 2014.

[33] C. Knabe, B. Lee, and D. Hong, "An inverted straight line mechanism for augmenting joint range of motion in a humanoid robot," in *ASME International Design Engineering Technical Conference*, 2014.

[34] B. Lee, C. Knabe, V. Orekhov, and D. Hong, "Design of a human-like range of motion hip joint for humanoid robots," in *ASME International Design Engineering Technical Conferences*, 2014.