

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/280839675>

# Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas

Article in *International Journal of Humanoid Robotics* · March 2016

DOI: 10.1142/S0219843616500079

CITATIONS

244

READS

8,417

8 authors, including:



**Twan Koolen**

Massachusetts Institute of Technology

28 PUBLICATIONS 2,599 CITATIONS

[SEE PROFILE](#)



**Sylvain Bertrand**

Florida Institute for Human and Machine Cognition

40 PUBLICATIONS 1,165 CITATIONS

[SEE PROFILE](#)



**Gray Thomas**

University of Michigan

58 PUBLICATIONS 625 CITATIONS

[SEE PROFILE](#)



**Tomas De Boer**

9 PUBLICATIONS 1,381 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Optimal Feedback Gain Selection for PD-type Series Elastic Actuator Impedance Controllers [View project](#)



Humanoid/Bipedal Locomotion [View project](#)

## Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas

Twan Koolen\*

*Department of Electrical Engineering and Computer Science,  
Massachusetts Institute of Technology, 77 Massachusetts Avenue,  
Cambridge, Massachusetts 02139, USA  
[tkoolen@mit.edu](mailto:tkoolen@mit.edu)*

Sylvain Bertrand

*Institute for Human and Machine Cognition, 40 Alcaniz Street,  
Pensacola, Florida 32502, USA  
[sbertrand@ihmc.us](mailto:sbertrand@ihmc.us)*

Gray Thomas

*University of Texas at Austin, 5706 Jeff Davis Ave,  
Austin, Texas 78756, USA  
[gthomas@ihmc.us](mailto:gthomas@ihmc.us)*

Tomas de Boer<sup>†</sup>, Tingfan Wu<sup>‡</sup> and Jesper Smith<sup>§</sup>

*Institute for Human and Machine Cognition, 40 Alcaniz Street,  
Pensacola, Florida 32502, USA  
<sup>†</sup>[tdeboer@ihmc.us](mailto:tdeboer@ihmc.us)  
<sup>‡</sup>[twu@ihmc.us](mailto:twu@ihmc.us)  
<sup>§</sup>[jsmith@ihmc.us](mailto:jsmith@ihmc.us)*

Johannes Engelsberger

*Deutsches Zentrum für Luft und Raumfahrt, Linder Höhe,  
51147 Köln, Germany  
[johannes.engelsberger@dlr.de](mailto:johannes.engelsberger@dlr.de)*

Jerry Pratt

*Institute for Human and Machine Cognition, 40 Alcaniz Street,  
Pensacola, Florida 32502, USA  
[jpratt@ihmc.us](mailto:jpratt@ihmc.us)*

Received 20 January 2015

Accepted 12 October 2015

Published 7 March 2016

\*Corresponding author.

This paper presents a momentum-based control framework for floating-base robots and its application to the humanoid robot “Atlas”. At the heart of the control framework lies a quadratic program that reconciles motion tasks expressed as constraints on the joint acceleration vector with the limitations due to unilateral ground contact and force-limited grasping. We elaborate on necessary adaptations required to move from simulation to real hardware and present results for walking across rough terrain, basic manipulation, and multi-contact balancing on sloped surfaces (the latter in simulation only). The presented control framework was used to secure second place in both the DARPA Robotics Challenge Trials in December 2013 and the Finals in June 2015.

*Keywords:* Motion control; Atlas; momentum; quadratic programming; capture point; bipedal walking; humanoid robot.

## 1. Introduction

As the field of legged robotics progresses, research focus is shifting toward having robots do useful work in the real world. Witnessed by e.g., the ongoing DARPA robotics challenge (DRC) and its disaster response scenario, it is no longer enough to perform a single task in a known lab environment. Instead, a wide variety of behaviors must be executed in an environment that is far less structured. These requirements motivate the development of a control framework that can be used to easily implement a rich class of behaviors for complex, floating-base robots in general contact situations, e.g., foregoing the assumption of flat ground. The purpose of this paper is to describe such a control framework and to present results on the physical Atlas humanoid robot designed by Boston Dynamics.

The proposed control framework was summarily introduced in earlier work.<sup>1</sup> It exploits the fact that the rate of change of whole-body centroidal momentum  $\dot{h} \in \mathbb{R}^6$  is both affine in the joint acceleration vector and linear in the external wrenches applied to the robot.<sup>2</sup> This fact is used to formulate a compact quadratic program, solved at every control time step, which reconciles motion tasks in the form of constraints on the joint acceleration vector with the available contacts between the robot and its environment in the form of unilateral contacts or force-limited grasps. Contact forces at unilateral contacts are constrained to remain within conservative polyhedral approximations of the expected friction cones.

Compared to earlier work,<sup>1</sup> this paper will expand more on the way a walking behavior was created using the control framework based on instantaneous capture point dynamics.<sup>3</sup> Furthermore, we have successfully transitioned the control framework from a simulation model to the physical humanoid robot Atlas, and present experimental results for walking across rough terrain, basic manipulation, and balance on sloped surfaces. The control framework was also used by Team IHMC to secure second place in the DRC Trials, held in December 2013 as well as in the DRC Finals, held in June 2015.

The remainder of this paper is structured as follows. Section 2 places the framework in the context of relevant prior work. Section 3 briefly recalls some key results concerning the centroidal dynamics of a robot. The main control framework is described in Sec. 4. Section 5 shows ways to use the proposed framework to control

humanoid walking. Section 6 notes the adjustments required for implementation on the physical Atlas robot. Section 7 presents the experimental results. Finally, a discussion is provided in Sec. 8.

## 2. Background

The proposed control framework is a model-based torque control scheme. One of the first such schemes is due to Khatib,<sup>4</sup> and was later extended and applied to humanoid robots.<sup>5–7</sup> Such a framework allows the specification of (a hierarchy of) multiple motion tasks, and can handle the case of multiple non-coplanar ground contacts. Joint torques are essentially found using unconstrained least squares (pseudo-inverse) techniques. A consequence of this unconstrained optimization is that limitations due to unilateral ground contact and friction cannot be taken into account directly. Other examples of this type of scheme include the work of Hyon *et al.*<sup>8,9</sup> and of Mistry *et al.*<sup>10,11</sup>

Taking unilateral ground contact into account naturally leads to the use of constrained optimization techniques. Although the friction cone constraints related to Coulomb friction are second-order cone constraints, these second-order cones are often approximated using linear constraints, reducing the problem of finding joint torques given motion tasks to a quadratic program (QP). The use of quadratic programming in online control algorithms has recently seen a sudden rise in popularity, at least in part due to the availability of fast and reliable QP solvers<sup>12–14</sup> and more powerful CPUs. An early example of such QP-based control schemes is the work of Kudoh *et al.*<sup>15</sup> More recent approaches include the work of Stephens and Atkeson,<sup>16</sup> Saab *et al.*,<sup>17</sup> Righetti *et al.*,<sup>18</sup> Kuindersma *et al.*,<sup>19</sup> Herzog *et al.*,<sup>20</sup> and Feng *et al.*<sup>21</sup>

Momentum-based control has also gained popularity in recent years. Kajita *et al.* were the first to propose a momentum based control scheme, called Resolved Momentum Control.<sup>22</sup> The authors recognized the fact that the whole-body momentum of a general robot is linear in the joint velocities, and used this to track a desired momentum reference while achieving other motion tasks. The Resolved Momentum Control scheme used unconstrained least squares to find joint velocity references to be tracked by a low level controller.

The relationship between momentum and joint velocities was studied in more detail by Orin and Goswami.<sup>2</sup> Based on this work, Lee, Goswami and Orin presented a momentum-based control framework for torque-controlled robots.<sup>23–25</sup> This work most directly inspired the proposed control framework. As opposed to Resolved Momentum Control, the authors employed constrained optimization to take into account the unilaterality of ground contacts. However, the framework is limited in the types of motion tasks that can be expressed. The authors additionally chose to split up the computation of the desired joint accelerations and ground reaction forces into separate optimization problems. Each of these optimization problems is easier to solve than a general QP, resulting in a reduced computation time, but also a less general and somewhat less elegant control scheme.

Similar to the work of Lee and Goswami, the proposed framework exploits the properties of whole-body momentum but recombines the problems of finding joint accelerations and ground reaction forces into a single, relatively compact QP. It also enables a simple way of specifying various types of motion tasks. The framework additionally ties in neatly with walking control based on instantaneous capture point dynamics<sup>3,26,27</sup> as will be shown in Sec. 5.2.

### 3. Centroidal Dynamics

This section briefly recalls some key properties of the centroidal dynamics of a robot. For a more in-depth discussion, see Orin *et al.*<sup>25</sup>

In this paper, we will consider a robot to be a rigid body mechanism with generalized joint torque vector  $\tau \in \mathbb{R}^n$ , joint configuration vector  $q \in \mathbb{R}^m$ , and joint velocity vector  $v \in \mathbb{R}^{n+6}$ .<sup>a</sup> We will assume that the only unactuated degrees of freedom correspond to the “floating joint” between the robot and the world. Let

$$h = \begin{pmatrix} k \\ l \end{pmatrix} \in \mathbb{R}^6 \quad (1)$$

denote the centroidal momentum of the robot, with  $k \in \mathbb{R}^3$  the centroidal angular momentum and  $l \in \mathbb{R}^3$  the centroidal linear momentum (notation from Orin *et al.*<sup>25</sup>). The term “centroidal” means that quantities are expressed in a frame that has the same orientation as the world frame, but has its origin at the robot’s instantaneous center of mass location. The robot’s centroidal momentum is simply the sum of the momenta of its links, expressed in this centroidal frame.

It is well known that the centroidal dynamics of a robot are simple, no matter how complex the robot is. Indeed, Euler’s laws of motion state that the rate of change of centroidal momentum is equal to the sum of all external wrenches applied to the robot. We can write

$$\dot{h} = W_g + \sum_i W_{\text{gr},i} + \sum_i W_{\text{ext},i}, \quad (2)$$

where  $W_g = (0_{3 \times 1} \quad mg^T)^T$  is the wrench due to gravity, the  $W_{\text{gr},i}$  are ground reaction wrenches exerted upon the robot’s bodies due to contact between these bodies and the environment, and the  $W_{\text{ext},i}$  are other external wrenches applied to the robot. Note that the rate of change of momentum does not depend on forces or torques internal to the robot.

Orin and Goswami showed that there is a simple linear relationship between the joint velocity vector  $v$  and the robot’s centroidal momentum<sup>2</sup>:

$$h = A(q)v, \quad (3)$$

<sup>a</sup>The reason for differentiating between  $v$  and  $\dot{q}$  is to allow an overparametrization of joint configurations so as to avoid singularity issues (e.g., using a quaternion instead of roll/pitch/yaw), while using a minimal set of coordinates for the velocity vector (e.g., using angular velocity instead of the derivative of a quaternion).

where  $A(q)$  is called the centroidal momentum matrix (CMM). Substituting this into Eq. (2) results in

$$A\dot{v} + \dot{A}v = W_g + \sum_i W_{gr,i} + \sum_i W_{ext,i}, \quad (4)$$

which shows that there is an affine relationship between the joint acceleration vector  $\dot{v}$  and the external wrenches applied to the robot. We exploit this fact in the proposed control framework.

#### 4. Control Framework

Figure 1 shows the high level flow of information in our control framework. A high level behavior, such as a walking or driving behavior, sets up a quadratic program (QP) with the following data: (i) desired motions in the form of motion tasks, (ii) information about available contacts, and (iii) a rate of change of centroidal momentum objective. The QP reconciles the motion tasks with the available contact by exploiting Eq. (4). The QP solver outputs joint accelerations and ground reaction wrenches exerted at the robot links that are in contact with the environment. This information is then used to calculate desired joint torques using an inverse dynamics algorithm. The computed torques are in turn used to determine the motor commands for the robot.

The remainder of this section is structured as follows. Section 4.1 discusses the way desired motions can be specified. External wrenches due to ground contact and object manipulation are discussed in Sec. 4.2. The quadratic program is formulated in Sec. 4.3. Finally, the inverse dynamics algorithm is discussed in Sec. 4.4.

##### 4.1. Desired motions

In the proposed framework, motion tasks are used to specify desired robot motions. Motions appear on the left-hand side of the momentum rate of change equation, Eq. (4). We define a motion task as an equation that is linear in the desired robot joint acceleration vector,  $\dot{v}_d \in \mathbb{R}^{n+6}$ , which includes the spatial acceleration of the

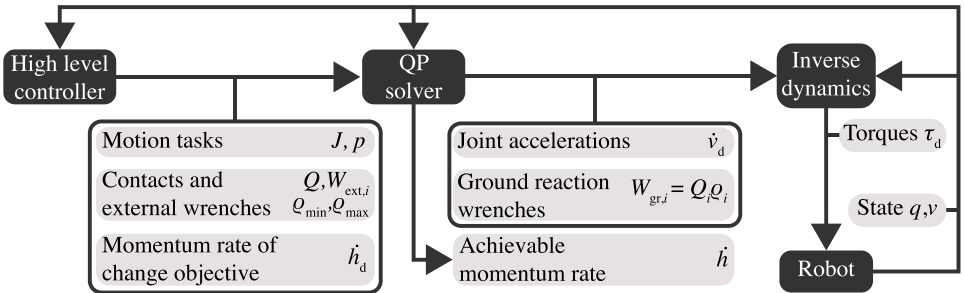


Fig. 1. High level overview of information flow in controller framework.

floating-base joint with respect to the world. Here,  $n$  is the number of actuated degrees of freedom. A motion task with index  $i$  can be written as

$$J_i \dot{v}_d = p_i. \quad (5)$$

We will first define some basic types of motion tasks in Sec. 4.1.1. Typical usage of motion tasks will be discussed in Sec. 4.1.2.

#### 4.1.1. Basic types

We mainly use three basic types of motion tasks: (i) joint space acceleration tasks, (ii) spatial acceleration tasks, and (iii) point acceleration tasks.

In the case of joint space acceleration tasks,  $J_i \in \mathbb{R}^{n_j \times (n+6)}$  is simply a selection matrix, where  $n_j$  is the number of degrees of freedom of joint  $j$ , and  $p_i$  is the desired joint acceleration  $\dot{v}_{d,j}$ :

$$J_i = (0 \quad \cdots \quad 0 \quad I \quad 0 \quad \cdots \quad 0), \quad p_i = \dot{v}_{d,j}. \quad (6)$$

To express a spatial acceleration task, we note that the twist

$$T_k^{m,j} = \begin{pmatrix} \omega_k^{m,j} \\ v_k^{m,j} \end{pmatrix} \in \mathbb{R}^6 \quad (7)$$

of body  $k$  with respect to body  $j$ , expressed in body  $m$ 's reference frame (notation taken from Duindam<sup>28</sup>) can be written as

$$T_k^{m,j} = J_k^{m,j} v, \quad (8)$$

where  $J_k^{m,j}$  is a geometric Jacobian<sup>28,29</sup> (or basic Jacobian<sup>4</sup>). Differentiating and rearranging, we find

$$J_k^{m,j} \dot{v} = \dot{T}_k^{m,j} - \dot{J}_k^{m,j} v. \quad (9)$$

Replacing the actual joint acceleration vector  $\dot{v}$  by the desired joint acceleration vector  $\dot{v}_d$  and the spatial acceleration  $\dot{T}_k^{m,j}$  by its desired value  $\dot{T}_{k,d}^{m,j}$ , we can write this as a motion task with  $J_i = J_k^{m,j}$  and

$$p_i = \dot{T}_{k,d}^{m,j} - \dot{J}_k^{m,j} v. \quad (10)$$

Note that spatial acceleration tasks can be expressed between two of the robot's rigid bodies, not just between a body and the world.

Point acceleration tasks are derived similarly to spatial acceleration tasks. In this case  $J_i \in \mathbb{R}^{3 \times (n+6)}$  is a Jacobian that maps the joint velocity vector to the Cartesian velocity of a body-fixed point and  $p_i$  is the desired linear acceleration of this point minus a velocity-dependent term, similar to the right-hand side of Eq. (10).

If only certain components of motion should be constrained, each of these basic motion tasks can be premultiplied with a selection matrix  $S$ , i.e.,  $SJ_i \dot{v}_d = Sp_i$ . For example, to only constrain the angular acceleration between two bodies, one can use a spatial acceleration task in conjunction with the selection matrix  $S = (I_3 \quad 0_{3 \times 3})$ .

#### 4.1.2. Typical usage

We typically use motion tasks to track joint space or task space trajectories, in which case the desired joint, spatial, or point acceleration contained in  $p_i$  is determined using PD control with an added trajectory-based feedforward acceleration term.

As an example, consider tracking an  $SE(3)$  trajectory specifying the reference configuration of body  $k$  with respect to body  $j$ . The reference configuration can be represented by a homogeneous transform

$$H_{k_r}^j(t) = \begin{pmatrix} R_{k_r}^j(t) & p_{k_r}^j(t) \\ 0 & 1 \end{pmatrix}. \quad (11)$$

The corresponding reference twist and spatial acceleration trajectories are written in body frame  $k$  as  $T_{k_r}^{k,j}(t)$  and  $\dot{T}_{k_r}^{k,j}(t)$ , respectively. In this case, we use  $p_i = \dot{T}_{k,d}^{m,j} - \dot{J}_k^{m,j} v$  from Eq. (10)<sup>b</sup> with  $m = k$  and with the desired spatial acceleration  $\dot{T}_{k,d}^{m,j}$  computed using a double-geodesic PD control law,<sup>30</sup> i.e.,

$$\dot{T}_{k,d}^{k,j} = \begin{pmatrix} K_{P,\omega} \log_{SO(3)}(R_{k_r}^k) \\ K_{P,v} p_{k_r}^k \end{pmatrix} + K_D T_{k_r}^{k,k} + \dot{T}_{k_r}^{k,j}, \quad (12)$$

where  $R_{k_r}^k$  and  $p_{k_r}^k$  are the rotation matrix and translation corresponding to the error transform  $H_{k_r}^k = H_j^k H_{k_r}^j$ ,  $T_{k_r}^{k,k} = T_{k_r}^{k,j} - T_k^{k,j}$  is the error twist, and  $K_{P,\omega}$ ,  $K_{P,v}$ , and  $K_D$  are positive definite gain matrices. Explicit time dependence was omitted. In our experience, adding the feedforward reference acceleration term allows the use of lower PD gains and improves tracking performance significantly for dynamic motions as long as smooth trajectories are used. We use smooth fifth-order polynomials as reference trajectories. Slerp interpolation is used for orientations.

Load-bearing bodies are typically constrained to have zero spatial acceleration with respect to the world. Alternatively, it is possible to constrain the motion of a load-bearing body such that it simulates damping in Cartesian space, as in Kuindersma *et al.*<sup>19</sup> Instead of using a spatial acceleration task, one can also use several point acceleration tasks, e.g., one for each of the corners of a rectangular foot. While this task formulation is not as compact as a single spatial acceleration task, it does offer more flexibility. For example, if we want the robot to only keep its toes on the ground during heel-to-toe walking, we can simply omit the point acceleration tasks for the heels, allowing an additional degree of freedom.

When using spatial acceleration or point acceleration constraints, it is of course possible for a motion task Jacobian  $J_i$  to become singular. The way we currently deal with these singularities is to simply switch to a different set of motion tasks. Continuing the example of heel-to-toe walking, the decision to switch off the heel point acceleration constraints can be made based on Jacobian determinants. This leaves the angular acceleration of the foot about the line passing through the toe points free.

<sup>b</sup>The bias term  $\dot{J}_k^{m,j} v$  is computed based on measured, not desired velocities.



## 4.2. External wrenches

This section concerns the right-hand side of the momentum rate of change equation, Eq. (4): the external wrenches. Ground reaction wrenches  $W_{\text{gr},i}$  are parametrized in terms of decision variables in the QP subject to certain constraints. Two forms of ground contact are available: unilateral ground contacts (Sec. 4.2.1), and force-limited grasping contacts (Sec. 4.2.2). Other external wrenches  $W_{\text{ext},i}$  to compensate for are discussed in Sec. 4.2.3.

### 4.2.1. Unilateral ground contacts

Unilateral ground contacts are modeled as point contacts with static Coulomb friction. Flat, polygonal contact surfaces with a distributed loading can be equivalently modeled using such point contacts at the vertices.

For point contact  $i$  with normal  $n_i$  the Coulomb friction constraint on the contact force  $f_i$  can be written as

$$\|f_i - (n_i \cdot f_i)n_i\| \leq \mu n_i \cdot f_i, \quad (13)$$

where  $\mu$  is the coefficient of friction. The second-order cone constraint in Eq. (13) cannot be incorporated in a quadratic program, and second-order cone programs (SOCPs) do not yet have fast solvers, so we use a standard conservative polyhedral approximation to the friction cone. The approximated friction cone is parametrized using a set of  $m$  extreme rays of the original second-order cone as basis vectors<sup>31</sup> (see Fig. 2), i.e., we replace Eq. (13) with:

$$\begin{aligned} f_i &= \sum_{j=1}^m \rho_{ij} \beta_{ij}, \\ \rho_{ij} &\geq 0, \end{aligned} \quad (14)$$

where the  $\beta_{ij} \in \mathbb{R}^3$  are the basis vectors and  $\rho_{ij}$  are basis vector multipliers (force intensities). For the results presented in this paper, we use four basis vectors and set  $\mu = 0.8$ .

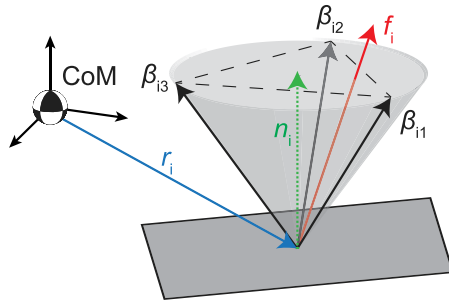


Fig. 2. A unilateral contact at position  $r_i$  in centroidal frame with normal  $n_i$ . The friction cone is approximated by basis vectors  $\beta_{ij}$ .

The ground reaction wrench  $W_{\text{gr},i}$  expressed in the centroidal frame is related to the contact force  $f_i$  by

$$W_{\text{gr},i} = \begin{pmatrix} \hat{r}_i \\ I \end{pmatrix} f_i, \quad (15)$$

where  $r_i$  is the location of contact point  $i$  in the centroidal frame, and  $\hat{r}_i \in \mathbb{R}^{3 \times 3}$  is the skew symmetric matrix such that  $\hat{r}_i f_i \equiv r_i \times f_i$  for any  $f_i$ . Combining Eqs. (14) and (15), we can write:

$$\begin{aligned} W_{\text{gr},i} &= Q_i \rho_i, \\ \rho_i &\geq 0, \end{aligned} \quad (16)$$

where

$$Q_i = \begin{pmatrix} \hat{r}_i \beta_{i1} & \cdots & \hat{r}_i \beta_{im} \\ \beta_{i1} & \cdots & \beta_{im} \end{pmatrix} \quad (17)$$

is a grasp map<sup>32</sup> and  $\rho_i = (\rho_{i1} \ \cdots \ \rho_{im})^T$ .

In practice, we use a surface normal  $n_j$  that is fixed in the frame of the contacting robot body. We prefer the extreme ray parametrization from Pollard and Reitsma<sup>31</sup> to the alternative parametrization in Stewart and Trinkle<sup>33</sup> because the extreme ray parametrization requires one fewer decision variable per contact point.

#### 4.2.2. Force-limited grasping ground contacts

For a grasping contact, forces and torques may be exerted in all directions, but they are modeled as being bilaterally bounded, i.e.,

$$\begin{aligned} W_{\text{gr},i} &= Q_i \rho_i, \\ \rho_{\min,i} &\leq \rho_i \leq \rho_{\max,i}, \end{aligned} \quad (18)$$

where  $Q_i$  transforms wrenches from a task frame (e.g., a hand-fixed frame) to the centroidal frame, i.e.,

$$Q_i = \begin{pmatrix} R & \hat{p}R \\ 0 & R \end{pmatrix},$$

where  $\begin{pmatrix} R & \hat{p} \\ 0 & 1 \end{pmatrix}$  is the homogeneous transform from task frame to centroidal frame.

So far, we have only used grasping contact constraints in simulation, in tasks such as ingress and egress of a vehicle.

#### 4.2.3. Pre-specified external wrenches

It is also possible to achieve force control tasks in this framework by pre-specifying wrenches that are exerted on the robot's links. This allows the robot to counter known disturbance wrenches on the robot, for example due to the fact that the robot is holding a tool. A known external wrench  $W_{\text{ext},i}$  can simply be incorporated in the right-hand side of the momentum rate balance Eq. (4) and in the inverse dynamics

algorithm. For tool manipulation, the mass and CoM location of the tool can be estimated online using a force/torque sensor in the wrist, and used to compute an external wrench to compensate, reducing the burden placed on feedback control using motion tasks.

### 4.3. QP formulation

The individual motion tasks are concatenated into a single matrix equation:

$$J\dot{v}_d = p, \quad (19)$$

where  $J = (J_1^T \cdots J_k^T)^T$ , and  $p = (p_1^T \cdots p_k^T)^T$ .

Likewise, the sum of all ground reaction wrenches can be written as a single matrix equation:

$$\begin{aligned} \sum W_{\text{gr},i} &= Q\rho, \\ \rho_{\min} &\leq \rho \leq \rho_{\max}, \end{aligned} \quad (20)$$

where  $Q = (Q_1 \cdots Q_m)$  and  $\rho = (\rho_1^T \cdots \rho_m^T)^T$ .

Combining Eq. (19), Eq. (4) with desired joint accelerations  $\dot{v}_d$  substituted for  $\dot{v}$ , and Eq. (20), we arrive at the following quadratic program:

$$\begin{aligned} \underset{\dot{v}_d, \rho}{\text{minimize}} \quad & (A\dot{v}_d - b)^T C_h (A\dot{v}_d - b) + \rho^T C_\rho \rho + \dot{v}_d^T C_v \dot{v}_d, \\ & J\dot{v}_d = p, \\ \text{subject to} \quad & A\dot{v}_d + \dot{A}v = W_g + Q\rho + \sum_i W_{\text{ext},i}, \\ & \rho_{\min} \leq \rho \leq \rho_{\max}, \end{aligned} \quad (21)$$

where  $b = \dot{h}_d - \dot{A}v$ , with  $\dot{h}_d \in \mathbb{R}^6$  a desired rate of change of centroidal momentum. The matrices  $C_h$ ,  $C_\rho$ , and  $C_v$  are cost function weighting matrices also determined by the high level controller. Example values of these weighting matrices are given for the walking behavior presented in Sec. 5.4.

This quadratic program is solved in every controller time step for the desired joint acceleration vector  $\dot{v}_d$  and the basis vector multiplier vector  $\rho$ . Given the multipliers  $\rho$ , ground reaction wrenches at each contact point are computed as  $W_{\text{gr},i} = Q_i \rho_i$ . We used CVXGEN<sup>14</sup> to generate a custom solver.

We take care to avoid conflicting motion tasks, which would lead to an infeasible QP. Recently, the use of hierarchical quadratic programming<sup>34</sup> has been proposed in a torque control setting.<sup>35,36</sup> Hierarchical quadratic programming allows the specification of a prioritized hierarchy of possibly conflicting tasks. We believe that it is indeed more natural to specify tasks in a (shallow) prioritized hierarchy. Unfortunately, fast and reliable hierarchical QP solvers were not available when the algorithm was implemented. In addition, we were worried about the possibility of chatter as tasks switch between being marginally feasible and infeasible.

#### 4.4. Inverse dynamics

The pre-specified external wrenches  $W_{\text{ext},i}$ , solved ground reaction wrenches  $W_{\text{gr},i}$  and desired joint accelerations  $\ddot{v}_d$  are used as the input to a recursive Newton–Euler inverse dynamics algorithm,<sup>37</sup> resulting in desired joint torques  $\tau_d$  for all joints, as well as a residual wrench that should be exerted across the “6 DoF joint” that connects a floating body (in our case, the pelvis) to the world. Because the desired joint accelerations are compatible with the total ground reaction wrench, this residual wrench is identically zero. Using a recursive Newton–Euler algorithm obviates the need to compute the mass matrix.

### 5. Application to Humanoid Walking

This section details how the proposed control framework was used to implement a walking behavior on Atlas.

Section 5.1 discusses the state machine used as a basis for the walking behavior. Balance control using instantaneous capture point dynamics is discussed in Sec. 5.2. Section 5.3 details the “touchdown” transition from single support to double support during walking. Finally, Sec. 5.4 discusses the selection of the weighting matrices in the QP.

#### 5.1. Walking state machine

A finite state machine is employed to control the various phases of walking. See Fig. 3 for a graphical description of this state machine, as well as a description of the state transition conditions. Each state specifies a different set of motion tasks (Sec. 4.1) and active contacts that can be used for balance (Sec. 4.2), which are aimed at placing the feet at user-specified footstep locations stored in a “footstep queue”.

During the single support states, smooth spline trajectories are used as references for motion tasks aimed at moving the foot to the next desired footstep from the queue. In addition, the transition into single support initializes: (i) an instantaneous capture point trajectory (see Sec. 5.2), (ii) a minimum jerk trajectory used as a

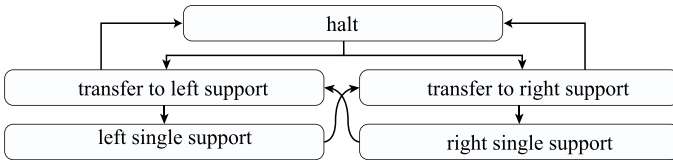


Fig. 3. Walking state machine. Boxes represent states and arrows represent possible state transitions. The halt and transfer states assume both feet are in contact with the ground. The transition from the halt state to a transfer state is triggered by the existence of unprocessed footsteps in an upcoming footstep queue. A transition back to the halt state occurs when the upcoming footstep queue is empty. The transition from transfer to single support is triggered once the instantaneous capture point trajectory (see Sec. 5.2) for the transfer state is finished. Finally, the transition from single support to transfer is detailed in Sec. 5.3.

reference for a pelvis orientation motion goal, and (iii) a minimum jerk trajectory for pelvis height. The final orientation for the pelvis orientation trajectory matches the yaw of the upcoming desired footstep and has zero pitch and roll.

Entry into a transfer state also triggers the initialization of a new instantaneous capture point trajectory. During execution of a transfer state, a heuristic determines whether it is safe to lift up the heel, in which case the contact points at the heel are disabled, and the rotation about the toe contact points is left unconstrained in the QP.

In all walking states, the upper body joints are controlled to a fixed joint position using joint space motion goals.

## 5.2. Instantaneous capture point control

During walking, the overall motion of the robot is planned and controlled based on instantaneous capture point (ICP) dynamics.

We first provide a brief derivation of the ICP dynamics for a general robot. Let  $r$  be the CoM position of the robot and  $m$  its total mass. Since the rate of change of whole-body linear momentum is  $\dot{l} = m\dot{r}$ , the last three rows of Eq. (2) can be written as

$$m\ddot{r} = mg + \sum_i f_{\text{gr},i} + \sum_i f_{\text{ext},i}. \quad (22)$$

Let us assume that there are no external forces other than the ground reaction forces during walking, i.e.,  $f_{\text{ext},i} = 0$ . Furthermore, we assume that the ground is flat and that the CoM remains at a constant height  $z$  above the ground. In this case, the sum of the ground reaction forces can be parametrized in terms of the centroidal moment pivot (CMP),<sup>38</sup> which can be defined as the location where a line through the CoM with direction defined by the total ground reaction force vector intersects with the ground plane (see Fig. 4). Due to the vertical force balance associated with maintaining a constant height, we can write

$$\sum_i f_{\text{gr},i} = \frac{r - r_{\text{CMP}}}{z} mg_z, \quad (23)$$

where  $r_{\text{CMP}}$  is the location of the CMP and  $g_z$  is the gravitational acceleration along the negative vertical axis (a positive number).

Substituting Eq. (23) into Eq. (22), using the projection matrix  $P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$  to select the horizontal part and simplifying, we obtain

$$P\ddot{r} = \omega_0^2 P(r - r_{\text{CMP}}), \quad (24)$$

where  $\omega_0 = \sqrt{\frac{g_z}{z}}$ . In earlier work,<sup>3</sup> we showed that the unstable part of the horizontal CoM dynamics for some simple models is described by the motion of the ICP, defined as

$$r_{\text{ic}} := Pr + \frac{\dot{r}}{\omega_0}. \quad (25)$$

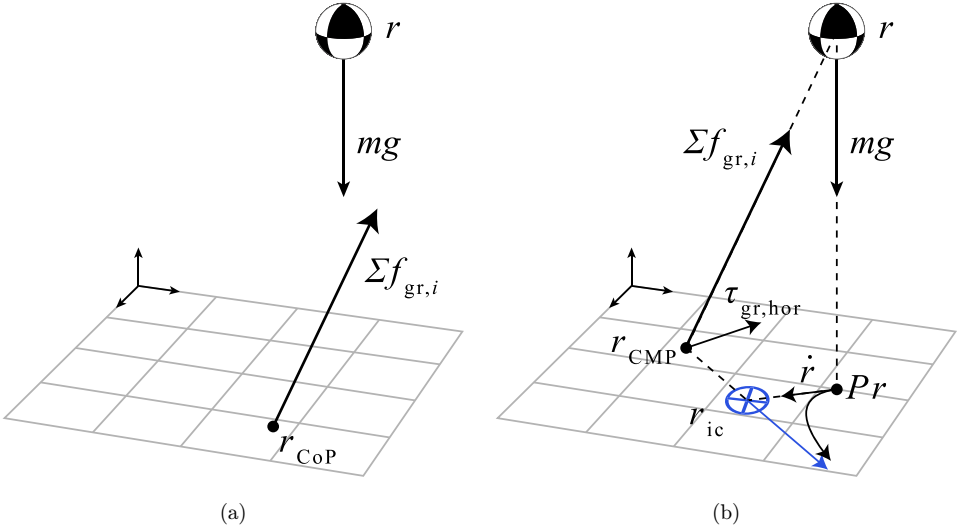


Fig. 4. Forces acting on robot: (a) gravitational force and total ground reaction force acting at the CoP, (b) total ground reaction force translated to CMP and a horizontal torque, together equivalent to forces in (a). Motion of the instantaneous capture point is also shown in (b).

Differentiating this definition and comparing with Eq. (24) shows that the ICP dynamics are described by

$$\dot{r}_{ic} = \omega_0(r_{ic} - Pr_{CMP}). \quad (26)$$

Figure 4 visualizes the forces acting on a general robot and the ICP dynamics. Earlier work<sup>3</sup> showed that Eq. (26) describes the ICP dynamics for an extension of the Linear Inverted Pendulum Model<sup>22</sup> consisting of a single rigid body kept at a constant height. In contrast, the above derivation shows that these dynamics hold for any rigid body system as long as the ground is flat and the CoM remains at a constant height.

We argue that it is wise to plan walking motions that always maintain the CMP inside the base of support, since the CoP is limited to lie within the base of support and any difference between the CoP and CMP results in a moment about the center of mass (see Fig. 4 and Popovic *et al.*<sup>38</sup>) and a corresponding nonzero rate of change of centroidal angular momentum, which may disturb posture control and cannot be maintained. This observation, in conjunction with the linear ICP dynamics Eq. (26), allows the generation of a desired ICP trajectory based on a series of upcoming footsteps. See Englsberger *et al.*<sup>39</sup> for details. The trajectory generation method results in a desired ICP position  $r_{ic,d}(t)$  and velocity  $\dot{r}_{ic,d}(t)$ .

We use Eq. (26) as the basis for an ICP control law aimed at tracking these desired values:

$$Pr_{CMP,d} = r_{ic} - \frac{1}{\omega_0} \dot{r}_{ic,d} + k_{ic}(r_{ic} - r_{ic,d}) + \dot{r}_{ic} \int (r_{ic} - r_{ic,d}) dt, \quad (27)$$

where  $r_{\text{CMP,d}}$  is the desired location of the CMP,  $k_{\text{ic}} > 0$  is a proportional gain and  $i_{\text{ic}}$  is an integral gain. We find that the feedforward term involving  $\dot{r}_{\text{ic,d}}$  greatly improves tracking performance compared to the control law presented in Pratt *et al.*<sup>26</sup> The integral term helps counter modeling errors.

Given a desired CMP obtained from Eq. (27), the horizontal component of the desired linear momentum rate of change is computed based on Eq. (23):

$$P\dot{l}_{\text{d}} = Pm\dot{r} = P \frac{r - r_{\text{CMP,d}}}{z} mg_z. \quad (28)$$

This serves as the horizontal linear momentum rate of change objective in the QP. The desired vertical linear momentum rate of change is provided by a CoM height control law.

The assumption that the ground is flat and that the CoM remains at a constant height turns out to be practicable even on rough terrain and stairs.

### 5.3. Transition from single support to transfer

The transition from single support to transfer is only possible when half of the planned foot swing time has passed in order to prevent premature state transitions due to noisy ground contact sensing. After this time limit, the transition occurs when ground contact is detected using the force sensors in the feet.

### 5.4. Selection of weighting matrices

In the weighting matrix  $C_h$  in the QP Eq. (21), only the diagonal elements that correspond to the horizontal component of linear momentum rate of change are set to nonzero values. We used a value of 1 for these elements, chosen arbitrarily to fix the scaling of the objective function.

The weighting matrices  $C_v$  and  $C_\rho$ , which respectively regularize joint accelerations and contact force basis wrench multipliers, were set to  $C_v = 0.05 \cdot I$  and  $C_\rho = 0.01 \cdot I$ . These values were tuned rather roughly by increasing them until linear momentum rate of change tracking started to deteriorate in simulation. They do not affect tracking of motion tasks, as these appear as constraints in the QP.

## 6. Application to Physical Atlas Robot

Several limitations of the sensing and actuation of the Atlas robot had a significant impact on the performance of the control framework. In this section, we discuss how the control framework was implemented on Atlas and adapted to deal with its limitations. We do this for two main reasons. First, these limitations are not only relevant to Atlas but are found on many other robotic platforms. Second, dealing with the limitations is an essential part of the control framework as it sometimes meant the difference between the robot performing well or not at all.

Some implementation details were previously published in our paper on the Trials phase of the DRC.<sup>40</sup> We refer to this paper for more information.

### 6.1. Control frequency

The controller is set up to have separate, dedicated threads for real-time state estimation and control. The state estimation thread runs at 1000 Hz, the rate at which Atlas publishes sensor data. The control thread runs at a lower rate of 250 Hz to allow more computation time. See Smith *et al.*<sup>41</sup> for implementation details.

### 6.2. Communication latency and actuator bandwidth

We have identified communication round trip latency as one of the main sources of instability when running our control framework on the physical robot. The worst case estimate of this round trip latency is 8 ms. See Table 1 for a breakdown of this estimate. Controlled damping, both in joint space and in task space, is most affected by latency and can cause instability when the gains are set too high. High natural damping in Atlas' joints allowed us to tune down most of the gains for controlled damping, reducing high frequency instabilities without generating low frequency instabilities.

Similar to latency, actuator bandwidth limitations also required us to use lower feedback gains (both proportional and derivative) than those used in simulation, at an acceptable cost to tracking performance.

### 6.3. Sensor noise

The physical Atlas robot has sensor noise, especially for the measurements of the joint velocities. This noise is inherent to the fact that joint velocities are obtained by numerical differentiation of analog position measurements, amplifying high frequency noise. We use low-pass filters to process the measurements with a break frequency of 16 Hz, which appears to be a good trade-off between smooth signals and low latency.

### 6.4. Backlash

Low level joint control, as well as high level control, also suffers from backlash in the transmission between the actuator and the link. When the backlash gap is open, a

Table 1. Latency associated with the various part of the Atlas control system. Latencies for components marked with a <sup>†</sup> could not be measured due to the proprietary nature of the Atlas API and are worst case estimates.

Component	Maximum latency
State estimator	1 ms
Controller	4 ms
Communication between control computer and Atlas <sup>†</sup>	1 ms
Atlas internal communication network <sup>†</sup>	1 ms
Atlas actuators <sup>†</sup>	1 ms
<b>Total</b>	<b>8 ms</b>



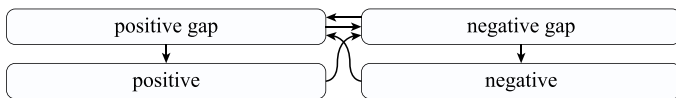


Fig. 5. State machine used to determine joint velocities given joint position measurements in the presence of backlash. Boxes represent states and arrows represent possible state transitions. The transition from *positive gap* (*negative gap*, respectively) to *positive* (*negative*) occurs after a specified time is spent in *positive gap* (*negative gap*), i.e.,  $t_{\text{gap}} > t_{\text{gap,max}}$  where  $t_{\text{gap}}$  is the time since the transition into the gap state, and  $t_{\text{gap,max}}$  is a constant, tuned to 0.03 s. Transitions to *positive gap* (*negative gap*) occur when the velocity obtained by numerically differentiating joint position changes from negative to positive (positive to negative). In the *positive* and *negative* states, the state machine directly outputs the joint velocity obtained from numerical differentiation, which is then low-pass filtered as described in Sec. 6.3. In the *gap* states, the numerically differentiated positions are first scaled by  $\frac{t_{\text{gap}}}{t_{\text{gap,max}}}$  before being low-pass filtered.

force generated by the actuator drives only the actuator itself and not the load. Since joint velocity feedback is based on sensing at the actuator side (as opposed to after the backlash gap at the load side), it is perceived that the load has moved rapidly. To counteract this measured peak in velocity, the controller generates a large counter-acting force. This causes the actuator to oscillate aggressively back and forth over the backlash gap at frequencies between 15 Hz to 100 Hz, leading to instabilities. To compensate for this backlash, we modify the control framework in three ways.

First, the input joint velocity signals are processed using a simple state machine, which attempts to disregard velocity measurements when the backlash gap is open. See Fig. 5 for an explanation of this processing step.

Second, the output desired joint accelerations are limited in their maximum amplitude and maximum rate of change (i.e., jerk). These limits are set to physically feasible values, which are typically chosen to be higher than the measured accelerations and jerk when the backlash gap is closed, but much lower than the measured accelerations and jerk when the actuator is oscillating over the backlash gap.

Finally, the QP is modified by adding a term to the cost function that penalizes the difference between the current solution of the ground contact forces and the previous solution. This effectively makes quick changes in the CoP location costly to the optimizer, which prevents the occurrence of rapid changes in the desired joint accelerations.

### 6.5. Low-level joint control

The control framework relies on accurate low-level joint control. While in simulation it is possible to directly apply the joint torques computed by the whole-body control framework, the physical robot can only be controlled through actuator valve current commands. For a joint with index  $j$ , changing the valve current  $i_j$  changes the oil flow into both chambers of the actuator, which changes the hydraulic pressure and hence the joint torque applied by the actuator.

Actuator output force is controlled through proportional control, with the actual force measured by pressure sensors in the hydraulic piston chambers. This piston

force is fairly accurately controllable. However, there is a significant amount of static friction between the hydraulic piston and cylinder. This stiction translates into joint torque stiction of up to 10 Nm. Consequently, hydraulic pressure inside the actuator is not a good measure of the total torque applied to the joint. While it is possible to accurately track a desired hydraulic force inside the actuator, the joint motion resulting from pure joint torque control is jerky and can cause CoP tracking errors.

The joints in the arms are more affected by static friction than those in the legs and back. This fact led us to use two different approaches to solving the problem of static friction.

For the legs and back, we chose to control the desired valve current  $i_j$  for a joint with index  $j$  not only based on the desired joint torque  $\tau_{d,j}$ , but also on a desired joint velocity obtained by integrating the joint accelerations that were found by solving the QP, Eq. (21). The control law for a leg/back joint can be written as

$$i_j = i_{\tau,j} + i_{\dot{q},j} \quad (29)$$

with

$$i_{\tau,j} = k_{\text{ff},\tau} \tau_{d,j} + k_{\tau,p} (\tau_{d,j} - \tau_j), \quad (30)$$

$$i_{\dot{q},j} = k_{\text{ff},\dot{q}} \dot{q}_j + k_{\dot{q}} \left( \int \dot{v}_{d,j} dt - \dot{q}_j \right), \quad (31)$$

where:

- $k_{\text{ff},\tau} \tau_{d,j}$  is a feedforward term based on desired torque;
- $k_{\tau,p}$  is a proportional gain acting on the difference between the desired torque  $\tau_{d,j}$  and the measured torque  $\tau_j$  (computed from the measured hydraulic pressure);
- $k_{\text{ff},\dot{q}} \dot{q}_j$  is a feedforward term that compensates for the oil flow in the cylinder as the actuator is moving;
- $k_{\dot{q}}$  is a gain acting on the difference between the desired joint velocity (obtained through numerical integration of desired joint acceleration  $\dot{v}_{d,j}$ ) and the measured joint velocity  $\dot{q}_j$ .

This control law strikes a compromise between torque control and joint velocity control. Torque control results in compliant behavior but is sensitive to unmodeled forces from the environment or from joint friction, whereas velocity control results in a high impedance behavior that is insensitive to unmodeled forces. By experimenting with the force control and velocity control gains, we were able to find a trade-off between smoothly tracking trajectories and being reasonably compliant to outside disturbances that allowed the robot to walk on a variety of terrains.

At the ankle joints,  $k_{\dot{q}}$  was kept at 0.0 since good ankle force control is critical for achieving desired CoP control.

For the arm joints, additional static friction required a control scheme with even higher associated impedance to achieve acceptable position tracking behavior. High impedance control is allowed by the fact that the arms are not used to interact with

the environment for balance. We chose to use a control law of the form

$$\dot{i}_j = \dot{i}_{\tau,j} + \dot{i}_{q,j}$$

with  $i_{\tau,j}$  as described earlier, and

$$\dot{i}_{q,j} = k_{\text{ff},q} \dot{q}_j + k_{q,p}(q_{d,j} - q_j) + k_{q,d}(\dot{q}_{d,j} - \dot{q}_j) + k_{q,i} \int (q_{d,j} - q_j) dt, \quad (32)$$

where  $q_{d,j}$  and  $\dot{q}_{d,j}$  are desired joint positions and velocities obtained directly from desired arm joint trajectories. The same arm joint trajectories are also used to formulate motion tasks for the QP as in Sec. 4.1.2. The parameters  $k_{q,p}$ ,  $k_{q,d}$  and  $k_{q,i}$  are proportional, derivative, and integral gains respectively.

### 6.6. Joint and link elasticity

Both the state estimator and the control framework assume the robot to be a rigid body system. However, the linkages between the actuators and the links of the robot are elastic, and joint angles are determined based on measured actuator deflections. This elasticity results in non-negligible errors when estimating the position of the CoM with respect to the feet. If not considered, such an error in the CoM position can result in the robot falling, especially in single support. To address this issue, we implemented an elasticity compensator in the sensor processing on the joint angle measurements. With this compensator, we assume that the elasticity of the links and joints can be represented as a deflection at the joint that is linear with the joint torque

$$q_{\text{proc},j} = q_{\text{raw},j} - \tau_j / K_j,$$

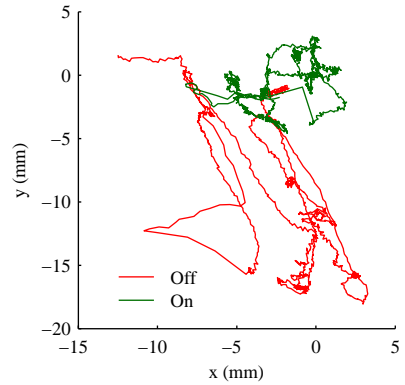
where  $q_{\text{proc},j}$  and  $q_{\text{raw},j}$  are the processed and raw angles for joint  $j$ ,  $\tau_j$  is the joint torque, and  $K_j$  is the estimated equivalent stiffness at the joint.

To tune the stiffness of the leg joints, we shift the robot weight rapidly in double support while ensuring that the feet are not moving at all. Because of the physical elasticity in the joints and links, the state estimator perceives a relative motion between the feet. A correlation can be identified between the erroneous motion of the feet and the leg joint torques. The joint stiffnesses  $K_j$  are tuned to reduce this correlation. We use a joint stiffness  $K_j = 10$  kNm/rad for all the joints except for hip yaw joints for which  $K_j = 7$  kNm/rad.

The performance of the elasticity compensation method was evaluated by clamping down the feet as shown in Fig. 6(a). The robot was then commanded to achieve scripted tension forces and torques between the feet using the technique described in Sec. 4.2.3. The tension forces ranged between  $-200$  N and  $200$  N in the forward ( $x$ ) direction and left ( $y$ ) direction. A tension yaw torque between the feet ranged between  $-100$  Nm to  $100$  Nm. The relative displacement of the left foot with respect to the right foot was then determined using forward kinematics, and is shown in Fig. 6(b). The elasticity compensator is seen to greatly reduce the relative



(a)



(b)

Fig. 6. Elasticity compensation evaluation. (a) Experimental setup with feet clamped. (b) Relative displacements between feet during experiment, as perceived by forward kinematics, with elasticity compensation on and off.

displacements. The elasticity compensation method also reduces the range of yaw displacements from 26.8 rad to 12.9 rad.

### 6.7. Foot slip

The walking controller was initially not robust to foot slip, which would occur frequently. In particular, the desired ICP trajectory was not updated based on sensor measurements. As a result, the desired ICP location at the end of a single support state could be outside of the upcoming support foot polygon due to a difference between planned and actual foot locations. This issue was resolved by including a feedback loop on the robot state to update the ICP plan. To reduce the amount of foot slip, we also added a Cartesian space PD control motion goal on the supporting foot to hold its position when barely loaded.

### 6.8. State estimation

As mentioned in earlier work,<sup>1</sup> we wrote a state estimator based on an extended Kalman filter for the virtual robotics challenge (VRC). The performance of this state estimator was excellent in the various simulated environments of the VRC. However, due to several non-Gaussian disturbances that were unmodeled in simulation, we observed unexpected behaviors on the physical robot. In the worst case, the estimated and actual CoM velocities were of opposite directions, leading to major instabilities. We implemented a new state estimator, starting from a simple estimation of the pelvis state using the leg kinematics and strongly trusting the measurements from the robot's inertial measurement unit (IMU).

The state estimator relies only on joint encoders to estimate the joint positions and velocities, and only on the IMU to estimate the pelvis orientation and angular

velocity. For the pelvis position, we fuse the measurements from the accelerometer and leg kinematics. At high frequencies, the measurement from the accelerometer is predominant, so as to reduce disturbances coming from the joint backlash, whereas at low frequencies leg kinematics is trusted more so as to reduce cumulative error due to the integration of the measured pelvis linear acceleration. The state estimator has a drift of about 2 cm to 3 cm per step for the pelvis horizontal position, and about 5 millimeters per step for the pelvis vertical position. Despite the drift in pelvis position, the performance is good enough for walking on uneven terrain. The state estimator is also robust to unexpected rotations of the support foot.

The drift can be further reduced with the help of nonkinematic sensors, such as the head-mounted LIDAR and stereo cameras on Atlas. Empirically, we were able to eliminate the drift using LIDAR-based 3D point cloud registration<sup>42</sup> to continuously measure the absolute position of the robot in the surrounding 3D environment. The two sensors complement each other in spatial and temporal resolution. In particular, kinematics odometry provides millisecond-millimeter-accurate incremental measurement while LIDAR localization provides second-centimeter-accurate absolute measurement. Lidar-based localization was not used in the DRC because there was not enough time available to test and verify performance and the operator found that kinematics-based localization was sufficient for completing the DRC tasks.

## 7. Results

This section presents experimental results for walking (Sec. 7.1), manipulation (Sec. 7.2), and balancing in single support (Sec. 7.3), as well as simulation results for multicontact balancing (Sec. 7.4). Experiments were done using the Boston Dynamics Atlas robot; see e.g., Kuindersma *et al.*<sup>19</sup> for some hardware details. See the attached video for footage of the experiments and simulations.

### 7.1. Walking

Here we present results for walking on flat ground (Sec. 7.1.1), over slanted cinder blocks (Sec. 7.1.2), and across unknown rough terrain (Sec. 7.1.3).

#### 7.1.1. Flat ground

Figure 7 shows frames from a video of a 10-step walking sequence on flat ground, starting and ending in a balanced configuration with the feet together. The data spans 14 s. While walking steadily, an average speed of 0.43 m/s is achieved, with a step length of 0.35 m, single support state times between 0.63 s and 0.68 s, and transfer (double support) state times between 0.17 s and 0.19 s. This walking gait is the fastest we have been able to produce on Atlas at this point and is limited by the hydraulic pump power, at least for the currently chosen gait.

Figure 8 shows the active state in the walking state machine and various tracking error signals during this run. Note that while foot position tracking errors during mid-swing are significant, they are reduced to within 2 cm at touchdown.



Fig. 7. Atlas walking on flat ground at a steady-state average velocity of 0.43 m/s. Frames are taken at 1.2 s intervals and are ordered left to right, then top to bottom.

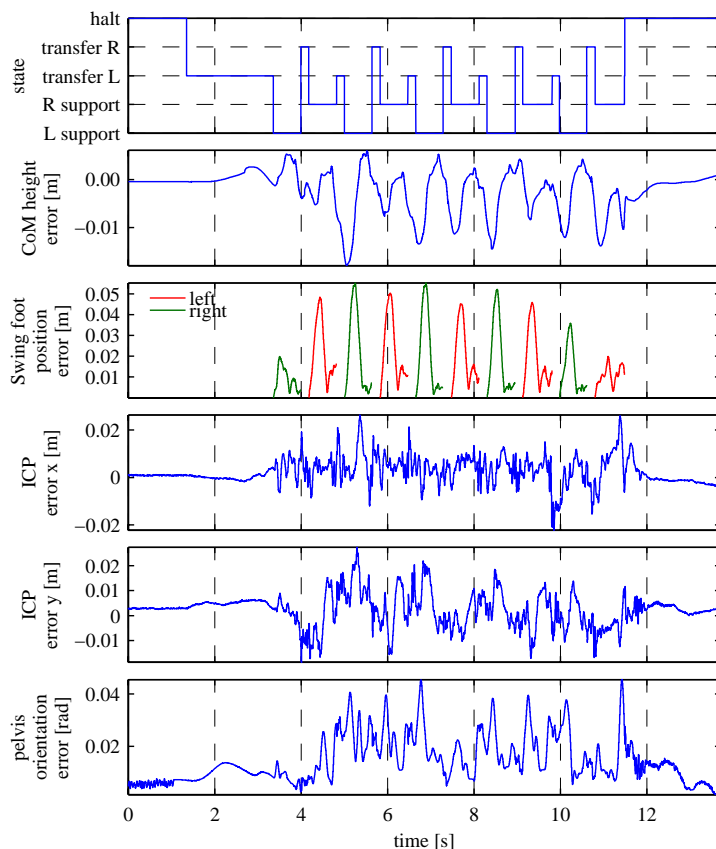


Fig. 8. State evolution and tracking errors during walking on flat ground at a steady-state average velocity of 0.43 m/s. Swing position errors are shown as magnitudes and are only plotted during single support states. In the pelvis orientation error graph, the angle of an angle-axis representation of orientation error is displayed.

See Fig. 10 for an overhead view showing the ICP and center of mass (CoM) trajectories and Fig. 9 for plots of these quantities versus time. Note that during steady-state walking, the CoM is never inside either foot polygon, demonstrating the dynamic nature of the walking gait. However, the ICP does enter the foot each step, signifying that the gait is one-step capturable,<sup>3</sup> i.e., it is possible to come to a stop in a single step at any time. Note also that the ICP significantly leads the CoM due to the substantial CoM velocity. While the ICP “ping pongs” between the feet in relatively straight lines, the CoM exhibits a smooth trajectory.

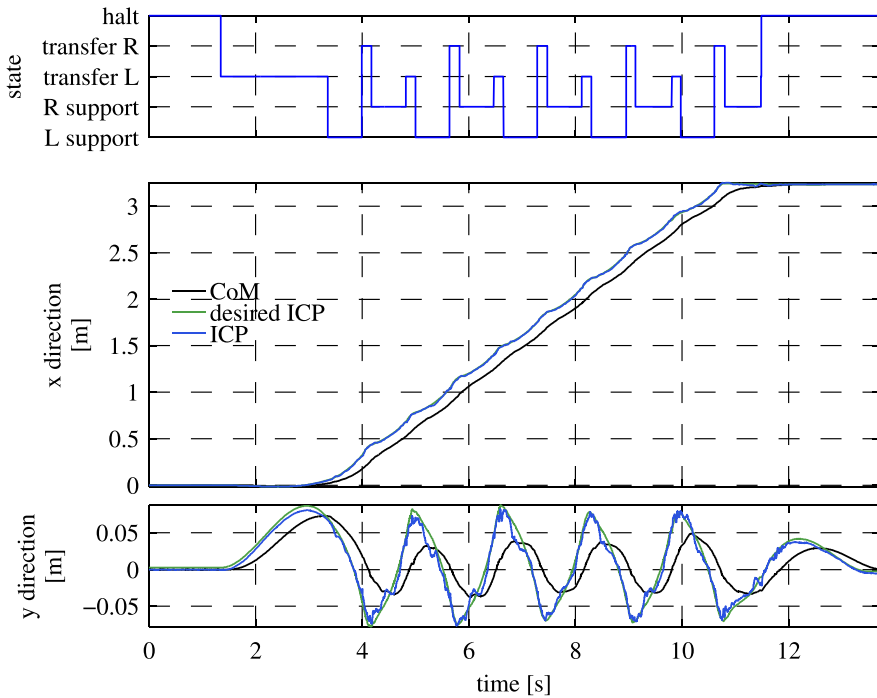


Fig. 9. Trajectory of CoM as well as of the desired and actual ICP while walking on flat ground at a steady-state average velocity of 0.43 m/s.

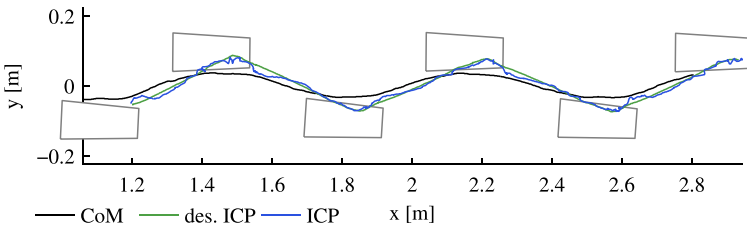


Fig. 10. Overhead view of CoM and ICP trajectories during steady state walking on flat ground ( $t \in [6, 10]$ ). To scale.



During the DRC, a slower, more conservative gait was used, obtained by down-tuning parameters such as step length and swing time, resulting in slower trajectories to track.

### 7.1.2. Cinder blocks

Figure 11 shows frames from a video of Atlas walking across a set of slanted cinder blocks using the same algorithm. During this run, the step locations are specified by the user. The robot traverses the cinder blocks in 20 s, taking 12 steps. See Fig. 12 for tracking performance.

### 7.1.3. Rubble

See Fig. 13 for video frames showing Atlas traversing rubble in the form of wooden boards. The wooden boards are not fixed to the ground. The terrain is unknown to the footstep planner: footsteps are simply planned at a fixed height equal to the average of the initial height of the feet, step length and step width are fixed, and foot pitch and roll are set to zero. This experiment demonstrates robustness of the walking controller to unknown terrain.

See Fig. 14 for tracking results. Note the foot placement errors of up to 5 cm and the altered state transition times, mostly a result of partial and unstable footholds deteriorating the control accuracy in the whole body.

## 7.2. Manipulation

Figure 15 shows Atlas manipulating a piece of wood during practice for the DARPA Robotics Challenge debris task.

## 7.3. Balancing in single support

Figure 16 shows Atlas balancing on one foot. Atlas is capable of standing on one foot while quickly moving the arms through a choreographed set of trajectories. Humans who attempt this task have trouble balancing without required taking a balance recovery step. Figure 17 shows tracking results for this task.



Fig. 11. Atlas walking over tilted cinder blocks. Frames are taken at 1.3 s intervals.



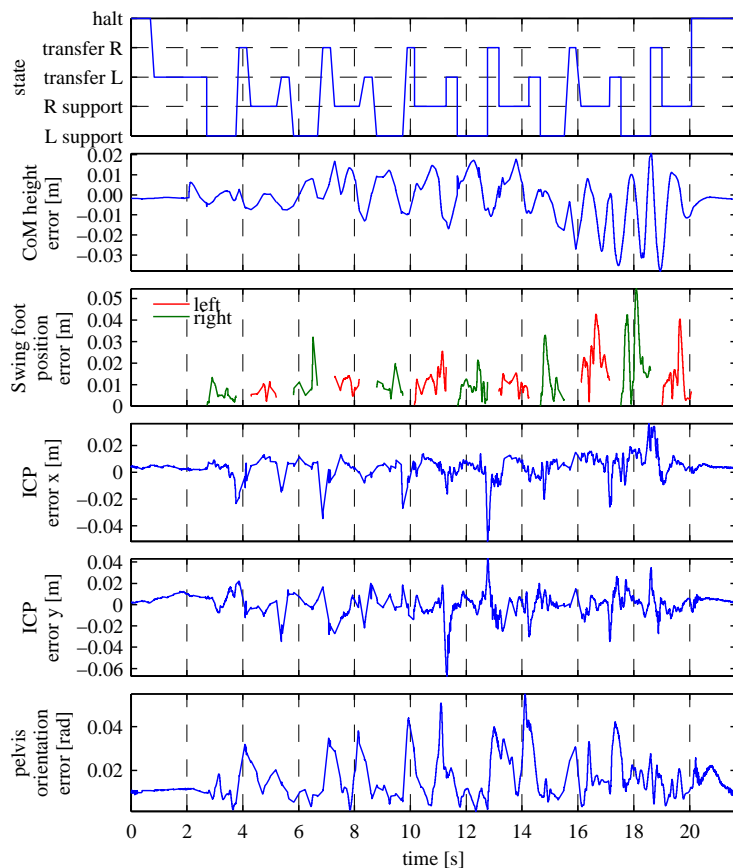


Fig. 12. State evolution and tracking errors while walking over tilted cinder blocks. Swing position errors are shown as magnitudes and are only plotted during single support states. In the pelvis orientation error graph, the angle of an angle-axis representation of orientation error is displayed.



Fig. 13. Atlas walking over rubble. Frames are taken at 1.6 s intervals.

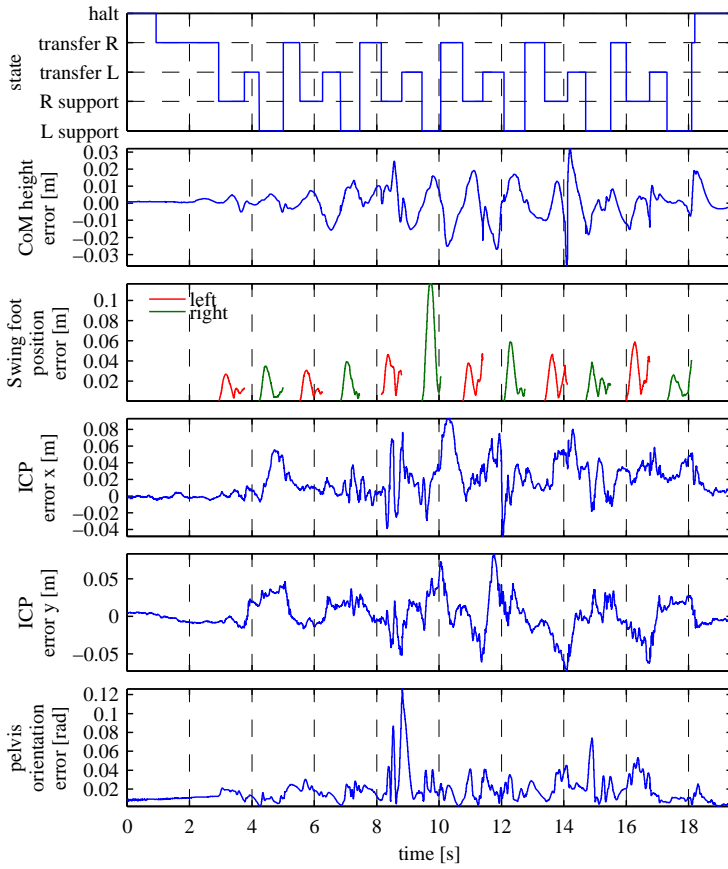


Fig. 14. State evolution and tracking errors while traversing rubble. Swing position errors are shown as magnitudes and are only plotted during single support states. In the pelvis orientation error graph, the angle of an angle-axis representation of orientation error is displayed.

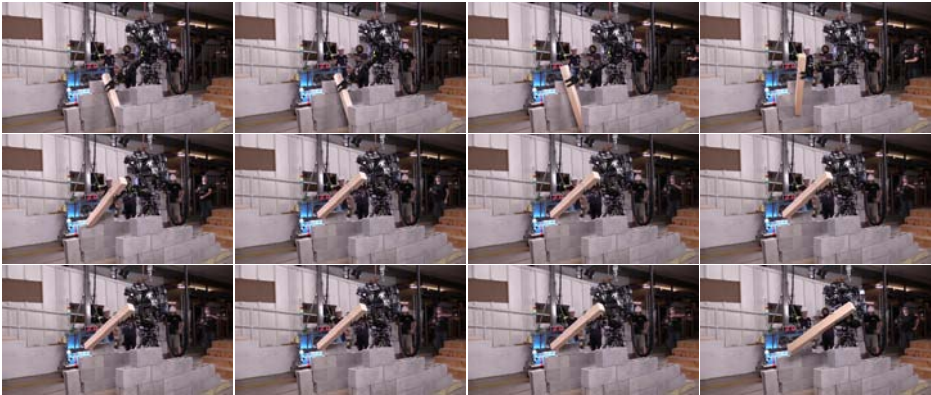


Fig. 15. Atlas moving a piece of debris. Frames are taken at 0.8 s intervals.



Fig. 16. Atlas balancing on one foot and executing a series of postures. Frames are taken at 1.9 s intervals.

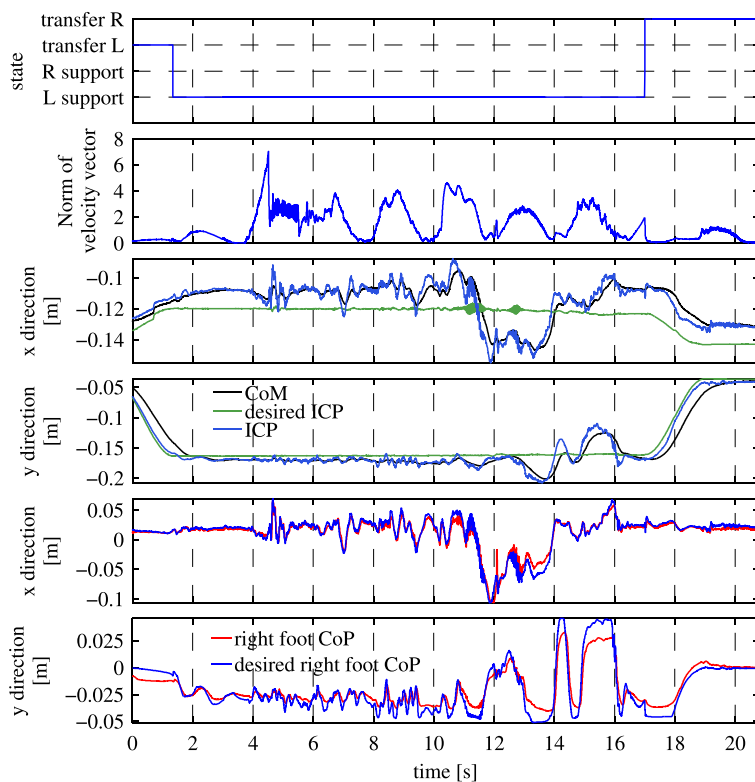


Fig. 17. CoM, ICP, and right foot CoP trajectories while balancing on the right foot. The desired right foot CoP was computed from the basis vector multiplier vector  $\rho$  (see Sec. 4.2.1) coming out of QP Eq. (21). Also plotted is the Euclidean norm of the robot velocity vector  $v$ , despite the dimension mismatch between the components of  $v$ . This subplot is only intended to show when the robot is moving to a different posture.

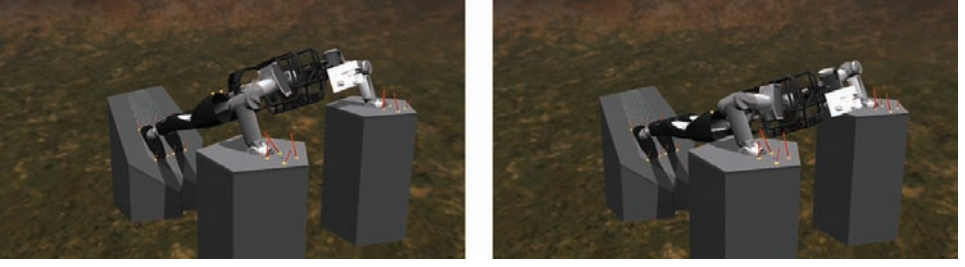


Fig. 18. Simulation model of Atlas performing push-ups while balancing on slanted surfaces. Ground reaction force vectors shown in red. For this simulation, the complex hand collision geometry was replaced by four contact points. Contact points are shown as yellow balls. Only these contact points can collide with the environment.

#### 7.4. Multi-contact balancing

Figure 18 shows a simulation model of Atlas performing push-ups while balancing on slanted surfaces. The formulation of the unilateral ground contacts with friction presented in Sec. 4.2.1 allows the robot to balance without slipping. Note that a simplified contact model using four contact points was used at the hands. The simulation was performed using Simulation Construction Set, a simulation environment originally developed by Yobotics Inc. and currently maintained by IHMC. Rigid body dynamics were computed using the Articulated Rigid Body Algorithm<sup>37</sup> and simulated using a fourth-order Runge–Kutta integrator. Ground contact forces were determined using spring-damper ground models.

### 8. Discussion

Our whole-body algorithm computes joint accelerations based on desired Cartesian accelerations. This requires effectively inverting a matrix, which can be singular when there is a subspace of Cartesian accelerations that are not achievable. Such a singularity occurs for example when the knee is straight and the ankle, knee, and hip are on a line, such that accelerations along this line are unachievable. Singularities such as this have been a major issue in robotics and one of the main reasons why so many humanoid robots walk with significantly bent knees even though real humans walk with fairly straight knees. We have experimented with several different methods for compensating for singularities, such as using damped least squares, but have not yet found a solution with which we are satisfied. In general, we believe that our techniques overconstrain degrees of freedom that need not be constrained. For example, to achieve walking, motion along the line from the ankle to the hip is not critical. Instead of precisely controlling Cartesian position in this direction, perhaps it would be more suitable to just control the knee angles with low gain feedback. In future work we hope to investigate techniques in which we reduce the level of control of those things which need not be controlled.

To date, most of our work with Atlas has been focused on producing reliable, robust walking with an emphasis on not falling down. This focus has been driven by the rules of the DARPA Robotics Challenge, in which falls lead to time penalties and damage to the robot. However, focusing on reliable walking has come at the price of ignoring energy efficiency and producing natural looking gaits. The flat ground walking results presented in this paper were obtained by pushing parameters of a reliable but inefficient walking algorithm to the limits of the hardware. In prior work, we have produced more efficient and natural-looking walking that exploits the natural dynamics of the robot better.<sup>43</sup> However, in that work the walking was not very reliable or adaptable to new situations, such as going over stairs. We hope to someday develop algorithms that are both reliable and that exploit natural dynamics.

Convex optimization is proving to be a very useful tool for robot control. Many robotics sub-problems can be written as a set of linear equality constraints and convex inequality constraints. Due to improvements in algorithms and increased speed of computers, as of this writing various algorithms are fast enough to solve problems with dozens of constraints on the order of a couple of milliseconds, allowing these algorithms to be used in real time control loops. A key requirement for walking robots is to prevent slipping by keeping interaction forces between the robot and the world inside friction cones. Friction cone constraints are conic constraints and thus convex. Second-order cone programs (SOCP) can be used to solve problems with such constraints, but current SOCP solvers are not fast enough. This led us to approximate friction cones with polyhedra, allowing the use of a quadratic program (QP) solver. It is likely that further improvements in convex optimization algorithms and speed increases in computers will in the future allow us to use more general convex constraints, as well as solve larger problems in real time loops.

Our whole body control algorithm is intended to be a compliant controller, in which we avoid rigidly controlling joint angles or any subspace that need not be stiff. We believe that such compliant control is desirable for increasing robustness to disturbances, such as stepping on unknown ground, and can also result in a safer robot, particularly for robots that interact physically with people. Many humanoid robots which utilize rigid position control at each joint have achieved impressive results when walking, even over slightly unknown terrain. Most of these robots have good ground force sensing in the feet or ankles and adjust their joint angle reference trajectories to compensate for variations between their expected zero moment point (ZMP) and the measured ZMP. Therefore, these robots do achieve compliant control along the subspaces required. However, this type of control requires accurate force sensing everywhere that a part of the robot might interact with the world. Instead, we prefer techniques in which every actuator inherently has a low impedance and achieve compliance through control of these actuators. To achieve desired ground reaction wrenches we therefore do not need force sensing at each contact location. However, we do need to know which points on the robot are in contact with the environment. In animals and humans, there are both low impedance muscle and high

resolution force and contact sensing throughout the skin. We look forward to a day when actuators and skin sensing on robots match the impressive characteristics of biology.

In this work a lot of effort was put into state estimation. Critically important for achieving balance with the presented algorithm is accurate knowledge of the CoM position and velocity, as well as the direction of the gravitation vector. Atlas has a high performance IMU with a precision fiber optic gyro, as well as a fairly accurate mass model of the robot. Because of this we were able to achieve impressive balancing results when doing upper body motions while balancing on one leg. From simple experiments of attempting to copy these motions ourselves, we believe that Atlas achieved better than human performance at balancing on one leg. However, our Atlas is still poor at recovering from disturbances by stepping. This is partially due to the slow swing speeds we can currently achieve (0.6 s minimum swing time with Atlas compared to approximately 0.3 s for a human). Improvements in swing time and accuracy will lead to significant improvements in disturbance recovery through stepping.

The presented algorithm works on flat ground, tilted cinder blocks, stairs, and level ground with moderate debris. In each of these cases the foot of the robot has a large contact area of support with the ground. For steep stairs, only the front half of the feet touch the stair, with the heel hanging over the edge of the stair. For cases such as this, the operator indicates the available foot contact area using the operator interface. The presented algorithms should extend to cases where the feet have a very small surface of contact with the ground as long as the available support polygon is sensed or deduced from sensory information so that it can be provided to the QP solver. We have done some preliminary work on deducing foot contact location by sensing an edge of rotation when walking over obstacles with partial foot support. Future work will include stepping on the tops of large pointed rocks, where only partial foot contact is possible.

## Acknowledgments

This work was funded through the National Aeronautics and Space Administration (Grant No. NNX12AP97G) and the Defense Advanced Research Projects Agency (Grant No. FA8750-12-1-0316). The Atlas robot was built and supported by Boston Dynamics. LIDAR-based localization algorithms were provided by Péter Fankhauser from ETH.

## References

1. T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Engelsberger, S. Mccrory, J. V. Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-I. H. Hoang, B. Layton, P. D. Neuhaus, M. J. Johnson and J. E. Pratt, Summary of team IHMC's virtual robotics challenge entry, in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Atlanta, GA (IEEE, 2013).

2. D. E. Orin and A. Goswami, Centroidal momentum matrix of a humanoid robot: Structure and properties, in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems* (2008), pp. 653–659.
3. 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, *Int. J. Robot. Res.* **31**(9) (2012) 1094–1113.
4. O. Khatib, A unified approach for motion and force control of robot manipulators: The operational space formulation, *IEEE J. Robot. Automat.* **3**(1) (1987) 43–53.
5. O. Khatib, L. Sentis, J. Park and J. Warren, Whole-body dynamic behavior and control of human-like robots, *Int. J. Human. Robot.* **1**(1) (2004) 29–43.
6. L. Sentis and O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, *Int. J. Human. Robot.* **2**(4) (2005) 505–518.
7. L. Sentis, Synthesis and control of whole-body behaviors in humanoid systems, Doctor of Philosophy, Stanford University (2007).
8. S.-H. Hyon, J. Hale and G. Cheng, Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces, *IEEE Trans. Robot.* **23**(5) (2007) 884–898.
9. S.-H. Hyon, R. Osu and Y. Otaka, Integration of multi-level postural balancing on humanoid robots, in *Proc. 2009 IEEE Int. Conf. Robotics and Automation* (IEEE, 2009), pp. 1549–1556.
10. L. Righetti, J. Buchli, M. Mistry and S. Schaal, Inverse dynamics with optimal distribution of ground reaction forces for legged robots, in *Emerging Trends in Mobile Robotics: Proc. 13th Int. Conf. Climbing and Walking Robots and the Support Technologies for Mobile Machines* (World Scientific Publishing Co. Pte. Ltd., Singapore, 2010), pp. 580–587.
11. M. Mistry, J. Buchli and S. Schaal, Inverse dynamics control of floating-base systems using orthogonal decomposition, in *2010 IEEE Int. Conf. Robotics and Automation*, No. 3, Anchorage, Alaska (2010), pp. 3406–3412.
12. L. Di Gaspero and E. Moyer, QuadProg++ (2015), <http://quadprog.sourceforge.net/>.
13. Gurobi Optimization, Inc., Gurobi Optimizer Reference Manual (2015), <http://www.gurobi.com>.
14. J. Mattingley and S. Boyd, CVXGEN: Code generation for convex optimization (2015), <http://cvxgen.com/>.
15. S. Kudoh, T. Komura and K. Ikeuchi, The dynamic postural adjustment with the quadratic programming method, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2002*, Vol. 3 (2002), pp. 2563–2568.
16. B. J. Stephens and C. G. Atkeson, Dynamic balance force control for compliant humanoid robots, in *RSJ Int. Conf. Intell. Robots* (IEEE, 2010), pp. 1248–1255.
17. L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres and J. Fourquet, Dynamic whole-body motion generation under rigid contacts and other unilateral constraints, *IEEE Trans. Robot.* **29**(2) (2013) 346–362.
18. L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan and S. Schaal, Optimal distribution of contact forces with inverse dynamics control, *Int. J. Robot. Res.* (2013), available at: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364912469821>.
19. S. Kuindersma, F. Permenter and R. Tedrake, An efficiently solvable quadratic program for stabilizing dynamic locomotion, in *Proc. Int. Conf. Robotics and Automation (ICRA)* (Hong Kong, China, 2014).
20. A. Herzog, L. Righetti, F. Grimminger, P. Pastor and S. Schaal, Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics, in *Proc. IEEE Int. Conf. Intelligent Robotics Systems* (2014).



21. S. Feng, E. Whitman, X. Xinjilefu and C. G. Atkeson, Optimization-based full body control for the Atlas robot, in *14th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2014 (IEEE, 2014), pp. 120–127.
22. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, Resolved momentum control: Humanoid motion planning based on the linear and angular momentum, in *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol. 2 (IEEE, 2003), pp. 1644–1650.
23. 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 *Proc. 2010 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (2010), pp. 3157–3162.
24. S.-H. Lee and A. Goswami, A momentum-based balance controller for humanoid robots on non-level and non-stationary ground, *Autonomous Robots* **33**(4) (2012) 399–414.
25. D. E. Orin, A. Goswami and S.-H. Lee, Centroidal dynamics of a humanoid robot, *Autonomous Robots* **35**(2–3) (2013) 161–176.
26. J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson and P. Neuhaus, Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower body humanoid, *Int. J. Robot. Res.* (2012).
27. J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schaffer and G. Hirzinger, Bipedal walking control based on capture point dynamics, in *2011 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, September (2011), pp. 4420–4427.
28. V. Duindam, Port-based modeling and control for efficient bipedal walking robots, Ph.D. dissertation, University of Twente (2006).
29. M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control* (Wiley, New Jersey, 2006).
30. F. Bullo and R. Murray, Proportional derivative (PD) control on the Euclidean group, California Institute of Technology, Technical Report (1995).
31. N. Pollard and P. Reitsma, *Animation of Humanlike Characters: Dynamic Motion Filtering with a Physically Plausible Contact Model*, Yale Workshop on Adaptive and Learning Systems (2001).
32. R. M. Murray, S. S. Sastry and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation* (CRC Press Inc., 1994).
33. D. Stewart and J. C. Trinkle, An implicit time-stepping scheme for rigid body dynamics with Coulomb friction, in *IEEE Int. Conf. Robotics and Automation, 2000, Proc. ICRA'00*, Vol. 1 (IEEE, 2000), pp. 162–169.
34. A. Escande, N. Mansard and P.-B. Wieber, Hierarchical quadratic programming: Fast online humanoid-robot motion generation, *Int. J. Robot. Res.* (2014) 0278364914521306.
35. A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal and L. Righetti, Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid, available at: <http://arxiv.org/pdf/1410.7284v1>.
36. A. Herzog, L. Righetti, F. Grimminger and S. Schaal, Experiments with hierarchical inverse dynamics controllers on a torque controlled humanoid, in *Proc. Dynamic Walking* (2014), available at: <http://www-clmc.usc.edu/publications/H/herzog-DW2014.pdf>.
37. R. Featherstone, *Rigid Body Dynamics Algorithms* (Springer-Verlag, New York Inc., 2008).
38. M. B. Popovic, A. Goswami and H. Herr, Ground reference points in legged locomotion: Definitions, biological trajectories and control implications, *Int. J. Robot. Res.* **24**(12) (2005) 1013–1032.
39. J. Engelsberger, T. Koolen, S. Bertrand, J. E. 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 *IEEE Int. Conf. Intelligent Robots and Systems (IROS 2014)* (2014), pp. 4022–4029.
40. M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. De Boer, T. Koolen, P. Neuhaus and J. Pratt, Team IHMC's lessons learned from the DARPA robotics challenge trials, *J. Field Robot.* **32**(2) (2015) 192–208.
  41. J. Smith, D. Stephen, A. Lesman and J. Pratt, Real-time control of humanoid robots using openjdk, in *Proc. 12th Int. Workshop on Java Technologies for Real-Time and Embedded Systems* (ACM, 2014), p. 29.
  42. F. Pomerleau, F. Colas, R. Siegwart and S. Magnenat, Comparing ICP variants on real-world datasets, *Auton. Robots* **34**(3) (2013) 133–148.
  43. J. E. Pratt, Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots, Doctor of Philosophy, Massachusetts Institute of Technology (2000).



**Twan Koolen** is currently a Ph.D. student in the Computer Science and Artificial Intelligence Laboratory at MIT and a member of the Robot Locomotion Group. Twan received his M.Sc. degree in Mechanical Engineering *cum laude* from Delft University of Technology in 2011, and has worked as a research associate at the Institute for Human and Machine Cognition between 2010 and 2014. His main research interest is dynamic and robust legged locomotion.



**Sylvain Bertrand** is currently a Research Associate at the Florida Institute for Human and Machine Cognition (IHMC). Sylvain received his M.Sc. degree in Mechanical Engineering from the University of Poitiers in 2009 and received his Ph.D. in Mechanical Engineering from the University of Versailles in 2013. His Ph.D. was about implementing and validating a running controller for a bipedal robot in simulation.

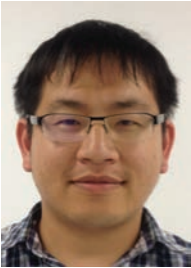


**Gray Thomas** is currently an M.Sc./Ph.D. student in the Mechanical Engineering Department at the University of Texas at Austin, and a member of the Human Centered Robotics Lab. Gray received his B.Sc. degree in Engineering with a self-designed concentration in robotics from Franklin W. Olin College of Engineering in 2012, and has worked as a Research Associate at the Florida Institute for Human and Machine Cognition between 2012 and 2013.

He has recently received the NASA Space Technology Research Fellowship to pursue robust identification and control of multi-contact dynamic locomotion with the series elastic NASA R5-Valkyrie Robot.



**Tomas de Boer** is currently developing autonomous robots for horticulture at Priva B.V. in the Netherlands. Tomas received his Ph.D. in Robotics in 2012 from Delft University of Technology. He has worked at PAL Robotics in 2012 for the development of the humanoid robot Reem-C, and was a Research Associate at the Florida Institute for Human and Machine Cognition (IHMC) in 2013.



**Tingfan Wu** received his Ph.D. from University of California, San Diego in 2013. From 2013–2015, he was a Postdoctoral Researcher at Institute of Human and Machine Cognition (IHMC). During his stay at IHMC, he worked on perception and control algorithms on multiple humanoid robots including Boston Dynamics Atlas, NASA Valkyrie, and Sandia Stepper. Tingfan's major research interests lie in the applications of machine learning techniques in computer vision and robotics problems. He is the author of multiple papers on facial expression analysis and robot learning.



**Jesper Smith** is currently a Research Associate at the Florida Institute for Human and Machine Cognition. Jesper received his M.Sc. in Mechanical Engineering from the Delft University of Technology in 2012. His interests include walking algorithms and system integration.



**Johannes Engelsberger** received the Dipl.-Ing. degree in Mechanical Engineering from the Technical University of Munich (TUM), Munich, Germany, in 2009. Since then, he has been working as Ph.D. student with the German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Wessling, Germany. His research interests include bipedal walking and running control, humanoid whole-body control and mechanical design.



**Jerry Pratt**, Ph.D., leads a research group at the Florida Institute for Human and Machine Cognition (IHMC) that concentrates on understanding and modeling bipedal and quadrupedal gait and applying that knowledge to the fields of robotics, human assistive devices, and man-machine interfaces. Current projects include humanoid avatar robots for co-exploration of hazardous environments, fast and efficient biologically-inspired running robots, and exoskeletons for restoration of gait in paralyzed individuals. Jerry led Team IHMC to a second place finish at the DARPA Robotics Challenge in 2015. He is an inductee in the Florida Inventors Hall of Fame.