

Walking on Partial Footholds Including Line Contacts with the Humanoid Robot Atlas*

Georg Wiedebach¹, Sylvain Bertrand¹, Tingfan Wu¹, Luca Fiorio², Stephen McCrory¹, Robert Griffin¹,
Francesco Nori², and Jerry Pratt¹

Abstract—We present a method for humanoid robot walking on partial footholds such as small stepping stones and rocks with sharp surfaces. Our algorithm does not rely on prior knowledge of the foothold, but information about an expected foothold can be used to improve the stepping performance. After a step is taken, the robot explores the new contact surface by attempting to shift the center of pressure around the foot. The available foothold is inferred by the way in which the foot rotates about contact edges and/or by the achieved center of pressure locations on the foot during exploration. This estimated contact area is then used by a whole body momentum-based control algorithm. To walk and balance on partial footholds, we combine fast, dynamic stepping with the use of upper body angular momentum to regain balance. We applied this method to the Atlas humanoid designed by Boston Dynamics to walk over small contact surfaces, such as line and point contacts. We present experimental results and discuss performance limitations.

I. INTRODUCTION

Humans are highly mobile and can walk over pointy rocks with the ground only partially supporting their feet. They walk on edges and point contacts without much trouble balancing. To achieve this they use a combination of various balancing mechanisms such as fast and dynamic stepping, shifting their Center of Pressure (CoP) within the available foothold (often referred to as the “ankle strategy”), and angular momentum-based methods such as “lunging” their upper body (often referred to as the “hip strategy”) [1] or moving their arms [2].

Humanoid robots can use these same strategies as humans, to varying effect [3]. Humanoid robots that take steps can maintain balance through foot placement. Those with finite-sized feet and controllability of their CoP can shift the CoP around in their base of support. Those with good control of their upper body can use angular momentum to help maintain balance. There has been theoretical work on using a combination of multiple recovery strategies [4], [5] in simulation to recover from disturbances such as pushes. When walking on unknown and potentially small footholds, the choice of balance strategy is essential. Taking a step to recover balance might not be possible if the potential footholds are sparse. CoP strategies might be limited by

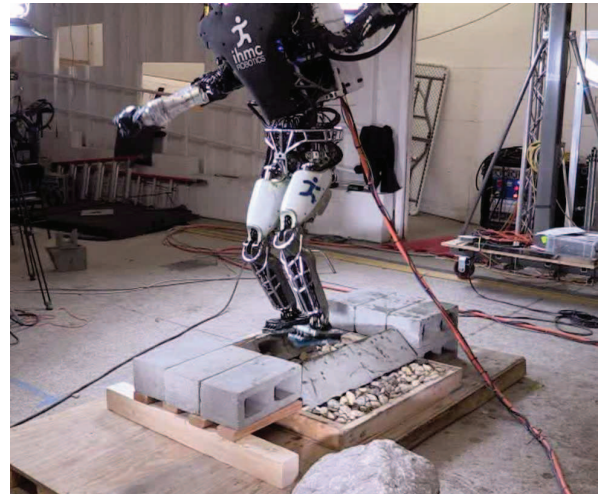


Fig. 1. Atlas walking on straight line contacts. Both cinder block rows are tilted and provide only line support for the feet of the robot. We present the approach taken to achieve the walking behavior seen here.

small support polygons. Angular momentum-based balancing as seen in Fig. 1 is generally available, but should be kept minimal since its effectiveness is limited by the robot’s kinematics. In addition, any lunging maneuver needs to be reverted when the robot brings its torso back into an upright position.

A second challenge is the estimation of the current foothold. Elevation maps of the environment built from 3d sensors, such as Lidars, can be used to find possible stepping regions [6] or to obtain a guess about the upcoming foothold. However, when walking on footholds with a small size this method might not offer the required precision. In some cases, such as walking on terrain that is covered in leaves or mud, visual foothold estimation can be misleading. To be useful in real world applications, a robot should be able to deal with unexpected partial footholds. Much like a human with closed eyes, it should carefully step and feel around with its foot before shifting its weight fully to the newly gained foothold. Humans have dense pressure sensing on the bottom of their feet, which allows them to instantly feel an “image” of the ground in fairly high resolution. Most state of the art humanoid robots, however, only have force sensors that allow them to detect the ground wrench, allowing for estimation of the immediate CoP, but not the full contact area. Instead, the available contacts must be inferred by the dynamic interaction of the foot with the ground, for example by observing rotation about contact edges and/or by

*This work was funded through the National Robotics Initiative by the National Aeronautics and Space Administration (grant number NNX12AP97G).

¹The author is with the Florida Institute for Human and Machine Cognition, 40 S Alcaniz St, Pensacola, FL 32502, United States

²The author is with the iCub Facility Department, Istituto Italiano di Tecnologia, Via Morego 30, Genova, Italy.

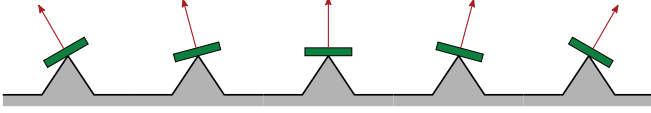


Fig. 2. When walking on sharp surfaces, we keep the foot in a close to flat orientation instead of having it conform to the ground. When balancing on a line, the foot orientation parallel to the line is held using position control while the torque perpendicular to the line is used to shift the CoP along the line contact. By rotating the foot, the effective surface normal can be altered to prevent slipping. If the foot surface is compliant, it can grip the sharp surface, further preventing slip.

examining achieved CoP locations during active exploration of the contact surface.

Most humanoid walking approaches in the literature assume full footholds and focus on footstep planning on potentially slanted but locally flat surfaces with large contact areas [7], [8], [9]. An example of this is the recent DARPA Robotics Challenge, where robots had to traverse a field of slanted cinder blocks. Other references, such as [10], [11] demonstrate the ability to traverse jagged rocks using robots with point feet. To our knowledge there has been little work on the task of walking over pointy rocks using bipeds with feet. When walking over rubble and small debris, torque controlled robots can rely on ankle compliance to let their foot conform to the ground. Here we focus on situations where this approach is not feasible when the available contact is severely limited. In this case, the foot should not conform to the ground but rather stay close to a flat orientation to avoid reaching ankle joint limits. Fig. 2 illustrates this situation with a front view of walking over a line or point contact. One advantage of walking on such surfaces is that the effective friction cone of the ground can be made quite large since the humanoid can angle its foot to achieve a large range of effective surface normals. Orienting the foot so that the surface normal is favorable to the desired ground reaction forces can significantly reduce the probability of slipping as compared to walking on flat but tilted surfaces.

To generate whole body motions for balancing and foothold exploration, we use the momentum-based control framework described in [12], which was first introduced in [13]. Since then, optimization based control frameworks became increasingly popular [14], [15]. We demonstrate the flexibility and the capabilities of the control algorithm. We show that it can handle non-trivial foothold geometries, foot exploration, online changes of the footholds, and complex balancing motions.

In our approach we combine fast swing times with upper body angular momentum to walk and balance on small footholds. After each step, we pause and explore the newly gained foothold before continuing to walk. First, we give an overview of the control framework used throughout this work in Section II. The estimation of the support area is described in Section III. The stepping and balancing is the subject of Section IV. Finally, we present experiments done with the Atlas robot in Section V that show the feasibility and limitations of the proposed method.

II. CONTROL FRAMEWORK

A momentum-based control framework is used to control the humanoid Atlas and is introduced in [12]. It uses a Quadratic Program (QP) to optimize a cost function at every controller time step. The core of the controller has been improved to enable the robot to explore footholds and balance with a limited support area. The QP formulation has been extended as follows:

$$\begin{aligned} \min_{\dot{\mathbf{v}}_d, \rho} \quad & c_{\dot{\mathbf{h}}_d} + c_{\mathbf{J}} + c_{\mathbf{P}} + c_{\rho} + c_{\dot{\mathbf{v}}_d} \\ \text{s.t.} \quad & \mathbf{A}\dot{\mathbf{v}}_d + \dot{\mathbf{A}}\mathbf{v} = \mathbf{W}_g + \mathbf{Q}_{CoM}\rho + \sum_i \mathbf{W}_{ext,i} \\ & \rho_{min} \leq \rho \\ & \dot{\mathbf{v}}_{min} \leq \dot{\mathbf{v}}_d \leq \dot{\mathbf{v}}_{max} \end{aligned} \quad (1)$$

The terms of the objective function are defined as

$$\text{Momentum Objective:} \quad c_{\dot{\mathbf{h}}_d} = (\mathbf{A}\dot{\mathbf{v}}_d - \mathbf{b})^T \cdot \mathbf{C}_{\dot{\mathbf{h}}} \cdot (\mathbf{A}\dot{\mathbf{v}}_d - \mathbf{b})$$

$$\text{Motion Objective:} \quad c_{\mathbf{J}} = (\mathbf{J}\dot{\mathbf{v}}_d - \mathbf{p})^T \cdot \mathbf{C}_{\mathbf{J}} \cdot (\mathbf{J}\dot{\mathbf{v}}_d - \mathbf{p})$$

$$\text{Contact Force Objective:} \quad c_{\mathbf{P}} = (\mathbf{P}\rho - \mathbf{r})^T \cdot \mathbf{C}_{\mathbf{P}} \cdot (\mathbf{P}\rho - \mathbf{r})$$

$$\text{Contact Force Cost:} \quad c_{\rho} = \rho^T \cdot \mathbf{C}_{\rho} \cdot \rho$$

$$\text{Joint Acceleration Cost:} \quad c_{\dot{\mathbf{v}}_d} = \dot{\mathbf{v}}_d^T \cdot \mathbf{C}_{\dot{\mathbf{v}}_d} \cdot \dot{\mathbf{v}}_d$$

where:

- $\dot{\mathbf{v}}_d$ are the desired joint accelerations and ρ consists of the contact point force magnitudes as introduced in [12].
- \mathbf{A} is the centroidal momentum matrix and $\mathbf{b} = \dot{\mathbf{h}}_d - \dot{\mathbf{A}}\mathbf{v}$ with $\dot{\mathbf{h}}_d$ being the desired rate of change of linear momentum and \mathbf{v} denoting the joint velocities. The momentum objective only incorporates the linear part of the centroidal momentum. This leaves the angular momentum free to be determined by the optimization.
- $\mathbf{J} = [\mathbf{J}_1^T \dots \mathbf{J}_k^T]^T$ and $\mathbf{p} = [\mathbf{p}_1^T \dots \mathbf{p}_k^T]^T$ are the concatenated Jacobian matrices and objectives for k desired motions.
- \mathbf{P} and \mathbf{r} can be used to define objectives on the ground reaction forces and will be defined later in this section.
- \mathbf{Q}_{CoM} is the Jacobian matrix from the contact force space ρ to the centroidal frame.
- $\mathbf{W}_{ext,i}$ are i external wrenches on the robot and \mathbf{W}_g is the gravitational wrench.
- ρ_{min} is the lower bound on ρ , used to enforce contact unilaterality.
- $\dot{\mathbf{v}}_{min}$ and $\dot{\mathbf{v}}_{max}$ are bounds on the joint acceleration, used to enforce joint angle limits. We compute the acceleration bounds every tick based on the current joint position and velocity to avoid hitting position limits.
- $\mathbf{C}_{\dot{\mathbf{h}}}$, $\mathbf{C}_{\mathbf{J}}$, $\mathbf{C}_{\mathbf{P}}$, \mathbf{C}_{ρ} , and $\mathbf{C}_{\dot{\mathbf{v}}_d}$ are positive definite cost function weighting matrices.

The first main contribution of this work is that all the desired motions are now part of the objective function, whereas they were previously formulated as equality constraints. The biggest difference we observe is that the momentum objective used to maintain balance can take priority over the motion objectives. We have introduced and tuned weights \mathbf{C} for all terms of the objective function. The weight magnitudes can be adjusted by the walking control algorithm

and may be changed online according to the current state of the robot. This enables the optimization to prioritize some objectives over others. For example, during stable stance on both feet, a high weight can be assigned to manipulation tasks to achieve good end effector trajectory tracking. On the other hand, during difficult balancing maneuvers, the weight on the momentum objective is increased, sacrificing precise upper body motions to help regain balance.

The second contribution is the introduction of an objective for the ground reaction force with the c_P cost term. It can be used to control the desired CoP of an end effector in contact with the environment by favoring a certain distribution in the ground reaction force magnitudes. For a single foot f in contact with the environment, the position of the local CoP on the sole can be written as

$$\mathbf{x}_{CoP,f} = \mathbf{P}_f \cdot \boldsymbol{\rho}_f, \quad (2)$$

$$\mathbf{P}_f \equiv \frac{1}{F_f^Z} \cdot \mathbf{S} \cdot \mathbf{Q}_f, \quad (3)$$

$$\mathbf{S} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

Here, \mathbf{Q}_f transforms all the force magnitudes $\boldsymbol{\rho}_f$ of the foot contact points to a single wrench at the contact plane origin. The matrix \mathbf{S} is a selection matrix selecting only the horizontal torques induced by the ground contact forces at the contact plane frame which is aligned with the bottom of the foot. The value of F_f^Z is the vertical foot force magnitude computed during the previous control step. The full \mathbf{P} matrix in the objective can then be assembled from all m flat end effectors with a CoP position objective. The vector \mathbf{r} contains all desired CoP positions $\mathbf{x}_{CoP,f,d}$:

$$\mathbf{P} = \text{diag}(\mathbf{P}_1 \dots \mathbf{P}_m), \quad (5)$$

$$\mathbf{r} = [\mathbf{x}_{CoP,1,d}^T \dots \mathbf{x}_{CoP,m,d}^T]^T. \quad (6)$$

III. FOOTHOLD DETECTION

To balance and walk it is helpful to have an estimate of the current support polygon as the area where the CoP can be placed. While walking over uncertain terrain, the new foothold is explored after each step. If there is no additional information about the foothold available we start by assuming the full foot to be in contact with the ground. Then the robot starts to shift its local foot CoP around within the foothold. If the position of the desired CoP is in an area of the foot that is supported by the ground, the measured CoP will closely follow the desired one. However, if the desired CoP is moving outside the base of support, the foot will start to rotate about the edge of the support (Fig. 3). We use this rotation of the foot to refine the estimate of our support polygon by cutting out the part of the foothold that was not able to carry weight.

To detect foot rotation, we implemented two methods: one based on the measured rotational velocity of the foot and one based on the geometry of the foot and ground planes. The first method relies on accurate joint velocity measurements from the state estimator to compute the rotational speed and

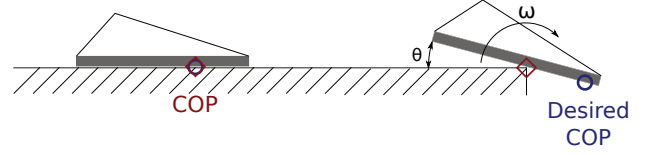


Fig. 3. Foothold Exploration. Side view of two feet with one foot rotating about the edge of the supporting ground. If foot rotation is detected, the control algorithm adjusts the foothold and removes the part that cannot support weight. The rotation speed ω and the angle between foot and ground θ are estimated and serve as indicators for foot rotation.

axis of rotation of the foot. A threshold on the rotational speed ω determines if the foothold should be adjusted. To determine the foot rotation geometrically, the foot plane is intersected with the ground plane. The intersection corresponds to the axis of the rotation and the angle θ between the planes equals the foot angle with respect to the ground. This requires an estimate of the ground plane normal and is not as fast as the velocity based detection, but is more robust to noise. Once a foot rotation is detected, the foothold is cropped at the line of rotation and the part of the foothold that would not support the desired CoP is removed. After exploration, we regulate the foot orientation to stay flat by adding an angular acceleration objective for each foot. This objective becomes active when the CoP of that foot comes close to an edge of the foothold. The acceleration driving the foot to a flat orientation is computed using a PD control law.

These foothold detection methods provide a quick estimate of the actual foothold. To obtain a more conservative or precise estimate of the support area we can maintain the set of measured CoP positions during exploration. Since the CoP cannot leave the area of support, the convex hull of the CoP measurements is a conservative bound on the foothold. If the geometry of the foothold is given by an elevation map or by the operator, the CoP positions can be used to fit the foothold to this map of known ground contact points (see Fig. 4). For example, it might be known from the elevation map that the foot contact is a line. To find the exact foothold we then fit a line to the measured CoP positions. This will account for inaccurate stepping due to swing foot tracking errors and provide the control algorithm with a precise location of the ground contact on the foot. To account for foot slippage, new measurements are trusted more heavily than previous ones.

Each foothold is represented as a collection of contact points at the corners of the support area [12]. Although the exploration of the foothold can result in a varying number of corner points, we found that it is sufficient in practice to approximate it with a constant number of four corner points. This simplifies the optimization problem by keeping its size constant.

IV. WALKING AND BALANCE CONTROL

The Instantaneous Capture Point (ICP) [16] is our main indicator of the balancing state of the robot. It is based on the Linear Inverted Pendulum Model (LIPM) [17] and is defined

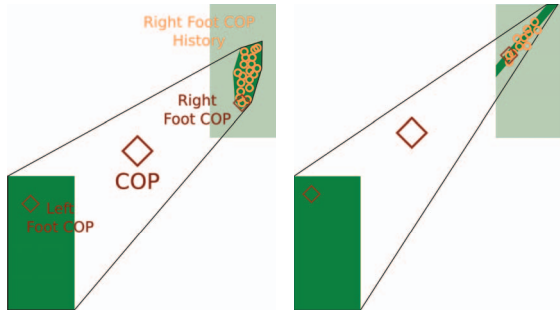


Fig. 4. We can use the CoP history to get a conservative estimate on the foothold by computing the convex hull of all measured CoP positions, or to fit a known terrain surface (such as a line) to the foot.

as

$$\mathbf{x}_{ICP} = \mathbf{x}_{CoM} + \frac{1}{\omega_0} \cdot \dot{\mathbf{x}}_{CoM} , \quad (7)$$

where $\omega_0 = \sqrt{\frac{g}{z}}$ is the pendulum constant. The equations presented in this section are reduced to two dimensions, since the ground reference points are defined to lie in the ground plane. The LIPM can be extended with a flywheel at the top to simulate upper body momentum and can be used to control the walking gait of a humanoid robot [18], [19]. It can be seen from this model that the Centroidal Moment Pivot (CMP) [20] reference point controls the ICP dynamics according to

$$\dot{\mathbf{x}}_{ICP} = \omega_0 \cdot (\mathbf{x}_{ICP} - \mathbf{x}_{CMP}) . \quad (8)$$

This equation states that the ICP will diverge from the CMP with increasing speed as the distance between the two gets bigger. During regular walking the torso of the robot is controlled to be upright with near zero upper body angular momentum. If there are no torques acting on the CoM of the robot the CoP coincides with the CMP [21]. Therefore, the CoP strategy is sufficient to keep the robot balanced as long as the ICP lies inside the support area. In cases where the ICP is outside the support area, the CoP strategy is not able to keep the robot balanced. In such a context, stepping or using angular momentum is necessary. When a step can be taken to where the ICP will be at the end of the swing, the use of angular momentum is generally avoided. When stepping is not available or is insufficient, angular momentum can be used to make the CMP leave the support area temporarily, depending on the available joint torques and range of motion. This way the ICP can be driven back towards the support to regain balance. This corresponds to an upper body lunging maneuver.

In our control framework the desired CMP position is transformed to an optimization objective: the desired rate of change of the linear, horizontal momentum of the CoM determines the position of the CMP. Therefore, we can compute

$$\dot{\mathbf{h}}_{d,linear} = \frac{m \cdot g}{z} (\mathbf{x}_{CoM} - \mathbf{x}_{d,CMP}) \quad (9)$$

from (7) and (8). The subscript d is short for desired and $\dot{\mathbf{h}}_{d,linear}$ refers to the horizontal component of the desired rate of change in the linear centroidal momentum. This is

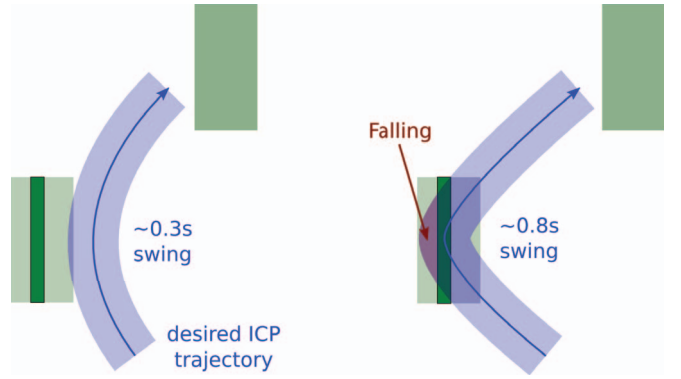


Fig. 5. ICP trajectories (blue) for different swing times during a forward step with the right foot. Here the stance foot has only a line contact with the ground (dark green). The light blue area accounts for possible disturbances, as well as sensor noise and modeling uncertainty, which will move the ICP away from its nominal path. It can be seen that a fast swing time is more robust to disturbances since the ICP is less likely to exit the support area in such a way that it cannot be caught by the upcoming step (red area). When this occurs, if the footstep location cannot be significantly modified, angular momentum must be used in order to prevent a fall.

the input to the optimization introduced in Section II. The desired position of the CMP is computed according to the control law

$$\mathbf{x}_{d,CMP} = \mathbf{x}_{ICP} - \frac{1}{\omega_0} \cdot \dot{\mathbf{x}}_{d,ICP} + k_p \cdot (\mathbf{x}_{ICP} - \mathbf{x}_{d,ICP}) \quad (10)$$

similar to the approach in [12]. Desired values are defined by a reference trajectory for the ICP that is computed based on the next footstep positions [22].

When walking on small footholds it becomes difficult to balance on the stance foot during a step. Reducing the step time can be helpful to increase the robustness of steps on small footholds. Fig. 5 conceptually shows a planned ICP trajectory with a bound that accounts for disturbances, sensor noise and modeling uncertainties. If the swing is fast enough, the ICP does not enter the stance foothold. Instead it gets directed towards the upcoming foothold and the robot needs to finish the step in order to regain balance. Generally it is desirable to “fall” towards the upcoming foothold in this way. If the stance support area is small, CoP control cannot be used to redirect the direction of the fall. The ICP control capabilities are limited by the achievable locations of the CMP. This means that a situation where the ICP leaves the support polygon, such that the robot is not falling towards the next foothold, is not recoverable unless the CMP leaves the support polygon as well.

The difference between CoP and CMP is directly related to the change in angular momentum [20]

$$\dot{\mathbf{h}}_{angular} = \tau_{CoM} = m \cdot g \cdot (\mathbf{x}_{CMP} - \mathbf{x}_{CoP}) . \quad (11)$$

In other words, torques on the CoM of the robot can be used to temporarily move the CMP outside of the support polygon of the robot and drive the ICP back inside the support area (Fig. 6). This is necessary when the ICP leaves the support and cannot be caught by an upcoming step. Applying torques on the CoM of the robot causes lunging of the upper body and produces motions that are similar to humans recovering

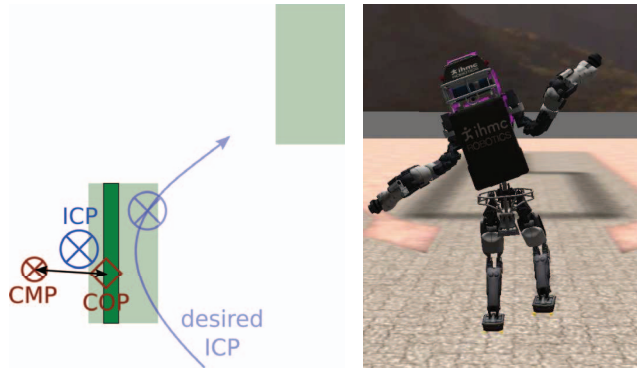


Fig. 6. When the ICP leaves the support area the robot can use a lunging maneuver to recover. The first picture is a top view of the ground reference points in that situation. The difference in CMP and CoP positions corresponds to a change in the angular momentum. The CMP “pushes” the ICP back to its desired trajectory. In the second image the simulated Atlas robot can be seen recovering from a push to the left during a step by using this lunging technique.

from strong pushes or balancing on small footholds. In our case the rate of change of angular momentum and the CoP are output of the QP and are not computed explicitly. During normal gait, the chest is kept upright limiting the generation of angular momentum. This results in the CoP staying close to the CMP. If this is not possible because the CoP can not leave the support area, the QP will be forced to use torque on the CoM to achieve its momentum objective, thereby creating a large distance between CMP and CoP. In addition we use weight scheduling on the momentum objective, increasing it when the ICP comes close to the edge of the support and decreasing it after recovering.

The lunging maneuver is limited by maximum actuator torques and joint limits of the robot, and can only be used for a short amount of time. In addition the angular momentum of the torso needs to be brought back to zero before joint limits are reached. The strategy of lunging the torso increases the robustness of walking on small footholds, particularly when swing speeds are slow, such that the robot must balance for an extended time over the support foot while waiting for the swing foot to reach the next step (Fig. 5 right).

To exploit knowledge of the current foothold of the robot we adjust the ICP and CoM trajectories of the robot during the step. If the current foothold is large we move the desired CoM at the end of the step closer to the current foothold. This prevents the robot from shifting a lot of weight to the new foothold. On the other hand, if the stance support area is small, we actively fall towards the upcoming foothold. In this case the desired ICP at the end of the swing is closer to the new foothold. This behavior for different foothold configurations can be seen in Fig. 7 and Fig. 8.

V. RESULTS & DISCUSSION

We employed the method described above on the Atlas robot¹. Using a combination of the foothold detection methods and balancing strategies, the robot was able to walk over

¹For a summary of the results presented here refer to the video at www.youtube.com/watch?v=_5PtXHsr038

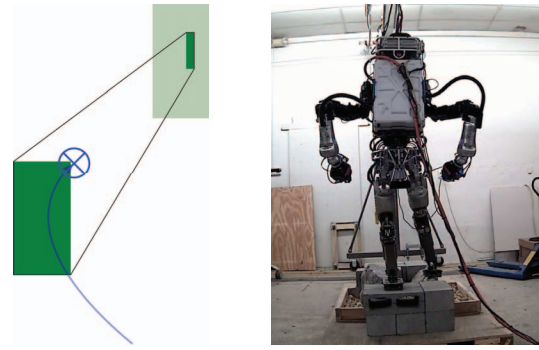


Fig. 7. If the expected foothold is small, the robot keeps its weight close to the stance foot and explores the upcoming foothold before shifting its weight to it. The left figures show the ICP trajectories with the final ICP at the end of the step, on the right side the Atlas robot can be seen right after taking a step that corresponds to the contact situation depicted on the left.

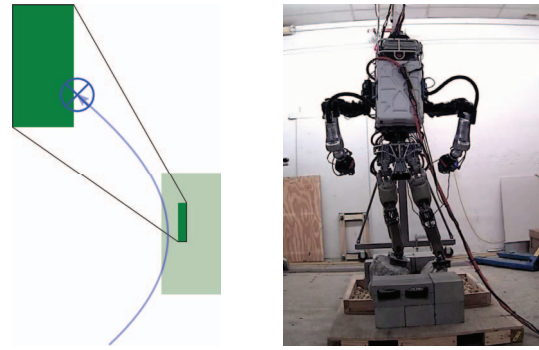


Fig. 8. When stepping to a good foothold while the stance foot is supported partially the robot will end the step with more of its weight resting on the good foothold.

different sets of limited footholds. For the results presented here a swing time of approximately 0.6 s was used. Faster walking causes the steps to become increasingly imprecise due to larger swing foot tracking errors. The hydraulic pump of the robot additionally limits the stepping speed, as the flow rate necessary to achieve the fast leg joint motions cannot be maintained. In comparison, human step times are approximately 0.3 s during fast walking. Therefore, Atlas must balance longer in single support and hence resort to lunging in order to prevent falling more than a human does (see Fig. 5).

Fig. 9 shows a simulated testbed with line shaped stepping stones. After each step is taken, the robot returns to a statically stable double support stance, with the CoM inside the support polygon. The robot then explores the new foothold before continuing to walk. The exploration phase takes about 1 to 3 s depending on the foothold.

The real robot was able to step over single, randomly oriented lines (Fig. 10). We created line contacts by tilting cinder blocks to a 45° angle. In the case of straight, forward oriented lines the real robot was able to continuously walk on the line contacts for several steps in a row (Fig. 1). The incorporation of angular momentum to maintain balance produces human like balancing motions. In Fig. 11, a top view of the ground reference point trajectories is shown. They were recorded during a run with the real Atlas robot



Fig. 9. The simulated Atlas robot walking over randomly oriented, line shaped stepping stones and a point foothold. After each step the new foothold is explored and the control algorithm adjusts the stepping accordingly.

walking over line contacts. It can be seen that for the line footholds the CMP leaves the support area towards the outside to prevent the robot from falling and producing angular momentum. Finally, Fig. 12 shows a sequence of pictures taken during a step with the stance foot on a angled line. In this sequence a sideways lunging motion can be seen that helps the robot maintain balance and finish the step.

Our algorithm requires adequate control of the CoP and CMP. On the Atlas robot, the CoP is controllable with an accuracy of approximately 2 cm due to good force control in the ankle joints. This is verified using the foot force sensors. To improve CMP tracking we add a joint velocity control term based on integrated desired accelerations to the leg and back joints [12]. Since there is no sensor for measuring the CMP directly, it is difficult to quantify how well the robot is able to control it.

As long as the combination of left and right footholds provides a sufficiently large support area we observed that the robot was able to regain balance after each step. If the footholds are alternating between full and partial, the simulated robot is able to explore and walk over a point stepping stone (actual size 2 cm \times 2 cm). In reality the performance of the robot was limited by multiple factors:

- The torque and velocity limits of the actuators on the robot bound the stepping speed and the change in angular momentum achievable. As hardware evolves, we are expecting to see robots perform better and better at tasks like balancing that reach the physical limits of the robot.
- Maximum joint angles of the hip and spine joints cap the amount of lunging motion. This is expected and similar limits are found in humans. The ICP algorithm we use for balancing relies on the LIPM with flywheel as model. It does not incorporate knowledge of the joint limits. To improve our algorithms we are working on extending our controller to incorporate knowledge of range of motion and other limitations.
- Finally, noisy sensors prevent precise ICP calculations and hinder the foothold estimation. Our simulations include sensor noise but disturbances, elasticities, and inaccuracies on the real robot make balancing on small

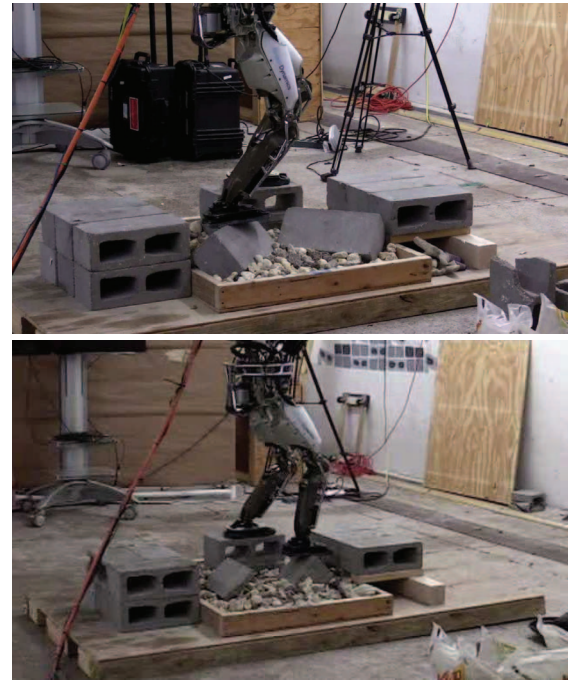


Fig. 10. Atlas walking over variously oriented line contacts. We use tilted cinder blocks to produce line contacts. It can be seen that the feet of the robot stay flat and do not conform to the cinder block surfaces.

footholds a hard problem. Velocity estimations are usually particularly noisy or, if filtered, delayed. We aim to improve our state estimator to provide fast and precise robot state measurements. A promising approach on improving joint velocity measurements using link mounted IMUs has been proposed in [23].

VI. CONCLUSION

Using the presented walking framework we were able to walk over partial footholds such as line contacts with the Atlas humanoid. This is an important step in the effort of making legged robots useful in real world scenarios. The ability to walk on unexpected partial footholds greatly increases the robustness of a robot when employed in cluttered environments. In addition it vastly extends the set of environments a robot can traverse. We use angular

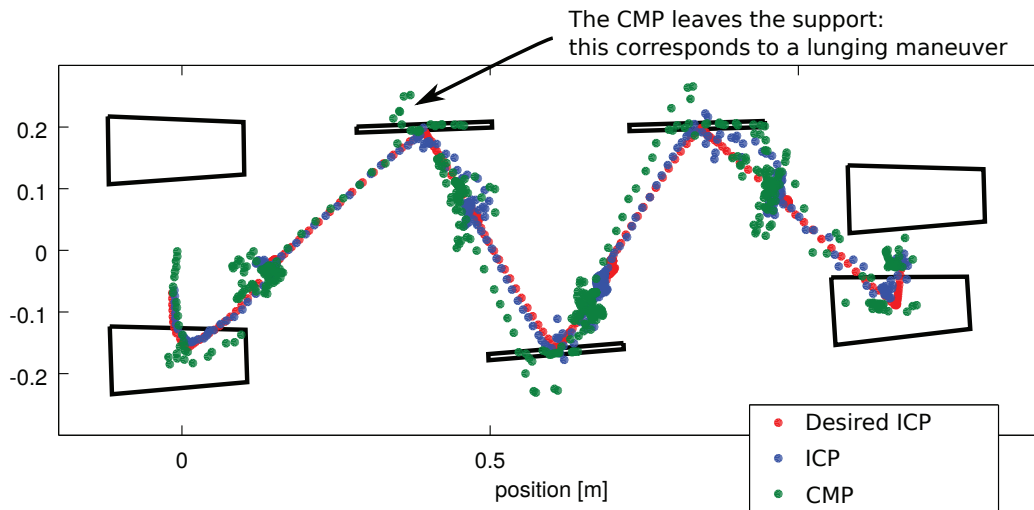


Fig. 11. This plot shows a top view of the measured ground reference points during a run with the real Atlas robot (walking from left to right). The robot walks over the straight cinder block field seen in Fig. 1. The black polygons show the detected footholds after line fitting. Every tenth data point is shown, each a duration of 10 ms seconds apart.

momentum to regain balance when other strategies cannot prevent a fall. In the future, we hope to improve the balancing capabilities of our robot by improving our control algorithms and state estimation and by applying the algorithms to robots with higher joint velocity limits.

REFERENCES

- [1] F. B. Horak and L. M. Nashner, "Central programming of postural movements: adaptation to altered support-surface configurations," *Journal of neurophysiology*, vol. 55, no. 6, pp. 1369–1381, 1986.
- [2] M. Pijnappels, I. Kingma, D. Wezenberg, G. Reurink, and J. H. van Dieën, "Armed against falls: the contribution of arm movements to balance recovery after tripping," *Experimental brain research*, vol. 201, no. 4, pp. 689–699, 2010.
- [3] B. Stephens, "Humanoid push recovery," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2007, pp. 589–595.
- [4] Z. Aftab, T. Robert, and P.-B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 159–164.
- [5] C. G. Atkeson and B. Stephens, "Multiple balance strategies from one optimization criterion," *IEEE Press*, 2007.
- [6] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1251–1262, 2012.
- [7] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 279–286.
- [8] A. Stumpf, S. Kohlbrecher, D. C. Conner, and O. von Stryk, "Supervised footstep planning for humanoid robots in rough terrain tasks using a black box walking controller," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 287–294.
- [9] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Planning and control of a humanoid robot for navigation on uneven multi-scale terrain," in *Experimental Robotics*. Springer, 2014, pp. 401–415.
- [10] I. Manchester, U. Mettin, F. Iida, and R. Tedrake, "Stable dynamic walking over rough terrain: Theory and experiment," in *ISRR 2009*. Springer-Verlag, 2009, pp. 1–16.
- [11] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 175–191, 2011.
- [12] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Engelsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot Atlas," *International Journal of Humanoid Robotics*, vol. 13, 1650007, 2016.
- [13] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.
- [14] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based Full Body Control for the DARPA Robotics Challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.
- [15] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2589–2594.
- [16] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.
- [17] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 239–246.
- [18] 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, 2012.
- [19] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhau, "Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower body humanoid," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.
- [20] M. B. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *The International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, 2005.
- [21] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3785–3790.
- [22] J. Engelsberger, T. Koolen, S. Bertrand, J. Pratt, C. Ott, and A. Albu-Schäffer, "Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4022–4029.
- [23] X. Xinjilefu, S. Feng, and C. G. Atkeson, "A distributed MEMS gyro network for joint velocity estimation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1879–1884.



Fig. 12. Picture sequence showing the Atlas robot taking a step with the stance foot only partially supported. The pictures in one row are taken at the same time. Each column shows the robot stepping from a different camera angle. During the step the robot lunges its upper body to the side to prevent a fall to the left. After the step is completed the upper body is brought back into a upright pose.