Stabilizing Series-Elastic Point-Foot Bipeds Using Whole-Body Operational Space Control

Donghyun Kim, Ye Zhao, *Member, IEEE*, Gray Thomas, Benito R. Fernandez, and Luis Sentis, *Member, IEEE*

***Abstract*—Whole-body operational space controllers (WBOSCs) are versatile and well suited for simultaneously controlling motion and force behaviors, which can enable sophisticated modes of locomotion and balance. In this paper, we formulate a WBOSC for point-foot bipeds with series-elastic actuators (SEA) and experiment with it using a teen-size SEA biped robot. Our main contributions are on devising a WBOSC strategy for point-foot bipedal robots, 2) formulating planning algorithms for achieving unsupported dynamic balancing on our point-foot biped robot and testing them using a WBOSC, and 3) formulating force feedback control of the internal forces—corresponding to the subset of contact forces that do not generate robot motions—to regulate contact interactions with the complex environment. We experimentally validate the efficacy of our new whole-body control and planning strategies via balancing over a disjointed terrain and attaining dynamic balance through continuous stepping without a mechanical support.**

***Index Terms*—Dynamic locomotion, point-foot bipedal robots, series-elastic actuator, whole body operational space control.**

I. Introduction

During DARPA's Robotics Challenge held, in 2013 and 2015, various humanoid robots were required to execute many practical tasks [1]. Most of these robots implemented a certain class of controllers called whole-body controllers (WBC), which are easily reconfigurable to accomplish multiple control objectives under environmental constraints. A perceived limitation of these robots was that their motions were much slower than their human counterparts. To remedy this issue, future WBC methods must significantly improve the speed and accuracy of the executed motions without losing their natural versatility. Additionally, humanoid robots with series elastic actuators (SEAs) are increasingly being sought for safe whole-body physical interactions among people and to move in cluttered environments. However, controlling fast motion behaviors with high precision on robots with SEAs is quite challenging [2].

To demonstrate that WBCs can achieve highly dynamic and versatile behaviors on biped robots with SEAs, here we devise the following:

1) software and hardware control methods to achieve whole-body operational space behaviors that are stable and accurate;
2) feedback control strategies to accurately regulate internal force behavior that yield pulling or pushing forces on complex terrains;
3) a pose estimator that can overcome the limitations of low-cost inertial measurement units (IMU) and be well suited for highly dynamic behaviors;
4) a locomotion planner and controller that can stabilize a point-foot robot on flat and rough terrains.

We consider WBCs to be very important for all types of humanoid robots because of their versatility and intuitive usage. Their versatility implies that WBCs designed for one type of robotic structure could be transferred to a different type of robot with minimum supervision. Additionally, WBCs have the advantage of decoupling task dynamics. In the case of whole-body operational space controllers (WBOSCs), it is possible to simultaneously control motion tracking behaviors and feedback-based internal force behaviors. Internal forces correspond to the subset of contact forces that do not generate robot motions. In a previous paper, we discussed the need to reduce joint torque controller gains to achieve better task motion tracking performance [2]. By combining task motion control with internal force feedback control, the WBOSC achieves precise tracking of the internal forces regardless of the tracking set point of the joint torque controllers. This capability is unique to WBOSCs and especially important when moving in sophisticated types of terrains that require the feet to apply tensions or compressions to the terrain by precisely controlling the contact forces.

Without a doubt, WBCs in general offer many advantages for versatility and contact interactions. And to get their highest performance in terms of whole-body motion and force interactions, they benefit by being implemented in torque-controlled robots such as those using SEAs. Previously, WBOSCs had not been formulated for and experimented with point-foot bipedal robots with SEAs. Therefore, the main contribution of this paper is to design a WBOSC for highly dynamic point-foot bipedal robots and assess its performance capabilities for various types of locomotion and balance behaviors.

Manuscript received October 22, 2015; revised May 9, 2016; accepted July 3, 2016. Date of publication November 4, 2016; date of current version December 2, 2016. This paper was recommended for publication by Associate Editor C. Ott and Editor A. Kheddar upon evaluation of the reviewers' comments. This work was supported by the Office of Naval Research and NASA's Johnson Space Center.

The authors are with the Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712-0292 USA (e-mail: dk6587@utexas.edu; zhaoye8810@gmail.com; gray.c.thomas@gmail.com; benito@mail.utexas.edu; lsentis@gmail.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material consists of a video, showing three experiments of unsupported dynamic balancing using a point-foot series-elastic biped robot called Hume. In the first and second experiments, Hume accomplishes 16 steps without falling. In the third experiment, Hume accomplishes 18 steps. The second part of the video shows Hume supported by a planarizer stage and balancing over a disjointed terrain. A student pushes Hume backwards and vertically. Hume is able to stay on the terrain without falling down thanks to its use of an internal force feedback controller. It also shows a stiffer behavior on the vertical direction as it has been programmed to do. The last part of the video shows a dynamic multibody simulation of Hume walking over a rough terrain. Our phase space planner enables Hume to walk toward a desired goal while negotiating the obstacle. In all the instance in this video, whole-body operational space control has been used to coordinate task motion behaviors and internal force control.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2016.2597314

In most cases, robots with unstable dynamics, such as point-foot bipeds, use control structures that are customized to the particular robot structure in hand. This customization allows them to perform optimally for the designed task, but makes it difficult to improvise controllers on demand or transfer them to other types of robotics structures. One of the most successful point-foot biped robots is ATRIAS [3], which can continuously walk unsupported using small passive feet. However, its locomotion controller highly relies on the characteristics of the hardware, whose natural dynamics is similar to the spring–mass model. The reliance on hardware characteristics makes the controller less versatile than WBCs. Additionally, ATRIAS has not been shown to regulate internal forces on disjointed terrains for gaining balance. Compared with ATRIAS that uses small passive feet to stabilize the yaw motion, our biped robot Hume does not have any feet. MABEL [4] and MARLO [5] also have clever controllers designed to follow the hybrid zero dynamics, the latter robot being able to achieve unsupported locomotion for several steps. However, identifying the hybrid zero dynamics is a nontrivial process and difficult to adapt to other locomotion patterns. MARLO achieves 20 steps of unsupported walk [5]. In comparison, our biped robot Hume achieves 18 steps of unsupported balance. Some quadruped robots, such as StarlETH [6] and HyQ [7], use WBCs for locomotion and multicontact tasks. They are also able to balance on disjointed terrains. Compared with WBOSCs, they do not use feedback control of the internal forces. With respect to balancing capabilities, quadrupedal robots are fundamentally different compared with point-foot bipeds, since the additional two legs provide multiple support points to regain and maintain balance. For instance, when a quadruped robot receives a kick, it can reposition two or three of its legs to create a large support polygon for balance adaptation and center-of-mass (CoM) control. On the other hand, point-foot bipeds do not have a supporting polygon and, therefore, need to rely on the robustness of their locomotion planners and the precision of the foot placement controller.

Other inverted-pendulum-based algorithms such as the zero-moment point [8], the capture point [9], or the divergent component of motion [10] have not been experimentally demonstrated in unconstrained point-foot robots to date. Previously, we introduced a new foot placement planner [11], which we extend for this paper for correctness and physical implementation via formulating a stability analysis, extending it to rough terrains, extending it to unsupported point-foot dynamic locomotion, and explaining the effect of ground impacts on the robot's trajectory. We use this locomotion planner and controller on the physical biped system for flat terrain locomotion, and we also present a walking demonstration on a rough terrain in a simulation.

This paper is structured as follows. In Section II, we present a background of the WBOSC and point-foot biped locomotion. Section III presents details of our experimental system including hardware and software. We explain gain scheduling and internal force feedback control in Section IV. In Section V, we present our foot placement planner. Section VI presents experimental results. Finally, Section VII discusses interesting issues and future issues to be addressed. Additionally, the Appendixes briefly summarize a WBOSC, establish the equations that we use for the prismatic inverted pendulum (PIP), and explain our state estimation approach for estimating the robot's body orientation.

II. Related Work

A. Whole-Body Controllers

WBOSC is a torque-level, prioritized, projection-based, multitask, operational space, feedback controller for floating-based robots originally outlined in [12] and extended to include internal force control in [13]. It uses null-space projections to accomplish task prioritization and computes contact and constraint consistent reduced Jacobian matrices to project multiple task impedance controllers into joint torques. The WBOSC also controls internal forces using the internal force matrix that resides in the orthogonal space to joint accelerations and by extension to operational space motion tasks.

Many experimental approaches for WBCs rely on impedance-based techniques or inverse dynamics and contain optimization algorithms to solve for constraints and contacts. One of them is the pioneering work by Hyon *et al.* [14], which formulates task-space impedance controllers to fulfill contact and task constraints; the concept of impedance control was first presented in [15]. Stephens [16] represents the first implementation that we know of to achieve full dynamic model-based task control with contact constraints on a humanoid robot. In [17], the implementation of hierarchical inverse dynamic algorithms using a quadratic program is presented and demonstrated on a Sarcos biped robot. Experiments include balancing while withstanding external forces, balancing on a moving platform, and standing on a single foot. In [18], a torque-based WBC is presented for controlling the Atlas and Valkyrie humanoid robots. A quadratic program solver is used to minimize the error with respect to the desired momentum rate, contact forces, and task accelerations. In [19], the WBC with inequality constraints via inverse dynamics and a quadratic program solver are proposed for the humanoid robot HRP-2. The algorithm is used offline to generate trajectories that are then tracked by a real-time controller. In [20], a torque-based WBC focused on optimizing multicontact wrenches derived from desired CoM movements is presented. These studies aim at controlling humanoid robots with actuated ankles and, thus, do not consider the underactuation nature and fast locomotion dynamics of point-foot biped robots.

Many of the previous WBCs formulate structures for controlling contact forces. There are many styles for regulating those forces. In [14], desired CoM accelerations and task wrenches are projected and summed up yielding a total CoM wrench. When there are multiple contacts, an optimization algorithm is applied to optimally distribute the contact wrenches before projecting them to the CoM. A CoM Jacobian transpose projection is used to achieve the desired global wrench. One of the problems with this method is that the forward kinematics model used for controlling the CoM does not account for contact constraints, which could result in less accurate motion tracking and force regulation. Nonetheless, the method allows the Sarcos humanoid robot to balance fairly well but with visibly slow CoM motions.

One great controller is the full-body controller by Stephens and Atkeson [21], focusing on two stages consisting of first

solving for the contact forces and then followed by solving the full-body constrained inverse dynamics problem. To deal with multiple contacts, a quadratic program is posed based on a simplified relation between CoM and task wrenches to solve for the contact forces that comply with friction constraints. Although this work was one of the first to produce locomotion on a physical humanoid robot using WBCs, its algorithm suffers from two problems: First, task wrenches are projected into the CoM using a heuristic model. Second, it solves the inverse dynamics problem by artificially incorporating fictitious task wrenches yielding a physically incorrect whole-body dynamic equation.

Another successful WBC is the momentum-based controller by Bertrand *et al.* [22]. Variations of this controller were used to obtain the second place for the DARPA Robotics Challenge. The controller has a similar two-stage nature as the previous full-body controller, but it has multiple advantages. First, for the contact solver, it uses the centroidal momentum matrix [23], [24], which correctly describes the relationship between external wrenches, i.e., contact wrenches and gravitation forces, and joint accelerations. Then, it solves a quadratic problem that includes motion task constraints. In turn, the output reaction forces are dynamically correct with respect to CoM behavior and motion tasks. Second, using the centroidal momentum matrix, it solves for joint accelerations, which is a more generic output than reaction forces alone. In its second stage, it solves for the constrained inverse dynamics based on the output joint accelerations and contact wrenches. In terms of being dynamically consistent, we cannot think of advantages or disadvantages between the momentum-based controller and our WBOSC. The biggest difference is that the WBOSC uses internal forces as the control input for regulating contact forces. Because internal forces are fully controllable, the WBOSC uniquely formulates a sensor-based feedback controller to precisely regulate internal forces. In comparison, previous controllers rely on feedforward control of the internal forces that require high-bandwidth joint torque control. However, the high-bandwidth joint torque control impedes achieving high-bandwidth task control, as discussed in Section IV-A.

It is necessary for us to convert desired contact forces to internal forces. Fortunately, this is realizable by using the simple projection operators from contact forces to internal forces described in (41). As for determining the desired contact forces that comply with frictional constraints, we described in [13] a search-based method to do so. In the current paper, we describe a simple multicontact model to solve for the contact forces that comply with frictional constraints. It would also be possible to use the contact solvers mentioned in [21] and [22] as a first stage and then proceed to solve the inverse task dynamics problem using WBOSC as the second stage of the controller. The takeaway message is that we can use any contact solver of our choice, convert contact forces to internal forces using simple projections as described in (41), and then proceed to solve for the whole-body torques using WBOSC. Note that controlling task accelerations plus internal forces is equivalent to controlling reaction forces. For instance, the momentum-based controller previously reviewed solves for the joint accelerations and reaction forces that comply with task accelerations and friction constraints. In our case, we directly solve for the task accelerations and compute the internal forces that comply with friction constraints. The two cases are functionally equivalent. However, the WBOSC allows for feedback control of internal forces, which other controllers do not.

There are several advantages of using WBOSC as the inverse task dynamics solver. First, we can formulate a sensor-based feedback controller to regulate internal forces. Using feedback control for the internal forces is beneficial, since we intentionally reduce torque controller feedback gains in favor of position tracking performance, as discussed in Section IV-A. Employing internal force feedback control leads to more accurate tracking of the desired internal forces without reducing the task position tracking accuracy. Another advantage is that the WBOSC exposes the task's effective inertia matrices, which can be used to study dynamic properties such as the effect of collisions on end-effectors. Another difference is that the WBOSC provides a framework for task prioritization, which allows behaviors to automatically repress conflicting tasks that are less critical for safety during runtime. Finally, the WBOSC supports the execution of overdetermined tasks, i.e., tasks that attempt to control more degrees of freedom (DOFs) than that are available, given desired priorities. For instance, in [25], we control multiple tasks associated with a cloud of marker points. There are more markers than DOFs in the robot.

The quadruped robot described in [6] uses a reaction force method to balance a quadruped on a ramp with 40° inclination. Another quadruped described in [7] balances on a split terrain by regulating the contact forces. These types of controllers are of a similar nature as WBOSC. One difference is that they do not implement feedback control of the internal forces.

Overall, there are many great WBCs out there supporting the control of multiple tasks under contact constraints that we have compared to WBOSCs. The main point of this paper is to demonstrate and analyze the performance of the WBOSC on a bipedal point-foot robot that is characterized by faster locomotion dynamics than humanoid robots with actuated ankles and has no supporting polygon compared to quadruped robots.

B. Locomotion of Point-Foot Biped Robots

Point-foot biped robots similar to ours have often been studied [26]–[31] due to their mechanical simplicity and fast locomotion capabilities. Only a few have achieved dynamical balancing without a constraint mechanism, the three most notable ones being the hydraulically actuated hopper from [32] and the bipeds from [3] and [28]. These last two biped robots are based on the same mechanical architecture but differ in the type of controller.

One of the most successful approaches to point-foot locomotion comes from [33], which stabilizes the hybrid zero dynamics of a high-dimensional multibody model of the robot. However, the algorithm is designed for robots supported by a planarizing mechanism. Recently, the same group [28] has combined this full-body locomotion algorithm with the SIMBICON method [34] to achieve unsupported point-foot locomotion. Like ours, the robot accomplishes numerous steps before falling.

In works [3] and [35], an unsupported point-foot robot accomplishes continuous walking by using only simple rule-based algorithms to stabilize the walk. Our locomotion algorithm is

different in that it searches adequate foot positions to stabilize the robot around a balance point. The balance point can stay in place or track a desired trajectory for locomotion. In terms of walking, we cannot think of advantages or disadvantages compared with that line of work. The higher performance of their experimental locomotion results is due to a number of factors that we believe are not related to the choice of locomotion planner. First, the mechanical structure of their robot is more rigid than ours, providing significantly higher foot positioning accuracy. Second, they use a high-performance IMU, which contributes to precision and tracking stability. Third, their robot uses small passive feet that prevent it from drifting on the yaw direction. In contrast, our robot does not have passive feet. One key difference of the WBOSC is that it allows biped robots to use internal forces for balancing on highly irregular terrains, such as our example on balancing Hume on a steep disjointed terrain.

Another successful locomotion approach also based on hybrid zero dynamics utilizes human-inspired trajectories to generate stable periodic locomotion and even handle some roughness on the terrain [31], [36]. However, the locomotion experiments use a mechanical planarizing stage to constrain the robot to a plane.

Frameworks such as the capture point [9] and the divergent component of motion [10] based on the linear point-mass pendulum model are very practical and widely used for controlling humanoid robots with actuated feet. However, they have not been used to control point-foot robots to date.

III. System Characterization

A. Hardware Setup

Our robot, Hume, is a biped robot that is 1.5 m tall and 20 kg in weight. The leg kinematics resemble simplified human kinematics and contain an adduction/abduction hip joint followed by a flexion/extension hip joint followed by flexion/extension knee joint, as shown in Fig. 1. The lack of ankle joints allows the shank to be a lightweight carbon fiber tube. The tip of the shank contains contact sensors based on limit switches. The SEAs on all six joints are based on a sliding spring carriage connected to the output by steel ropes. The deformation in the springs is directly measured within the carriage assembly. The concept, kinematics, and specifications of the robot were proposed by our team at UT Austin, and the manufacturing was done mostly by Meka Robotics with some of our help. For fall safety, the robot is attached to a trolley system with a block and tackle system allowing easy lifting and locking at a particular height.

Hume is controlled with distributed digital signal processors connected by an EtherCAT network to a centralized PC running a real-time RTAI Linux kernel. This communication system introduces a 0.4-ms delay from the Linux machine to the actuator DSPs and back. Each DSP controls a single actuator, and they do not communicate directly with each other. Power is delivered through a tether.

Similar to [37], Hume is electrically powered and has six actuated joints. Our pose estimation setup combines an overhead motion capture system with a low-cost IMU. This sensing modality contrasts the setup found in [3], which takes advantage of its highly accurate IMU sensor. A relatively inexpensive

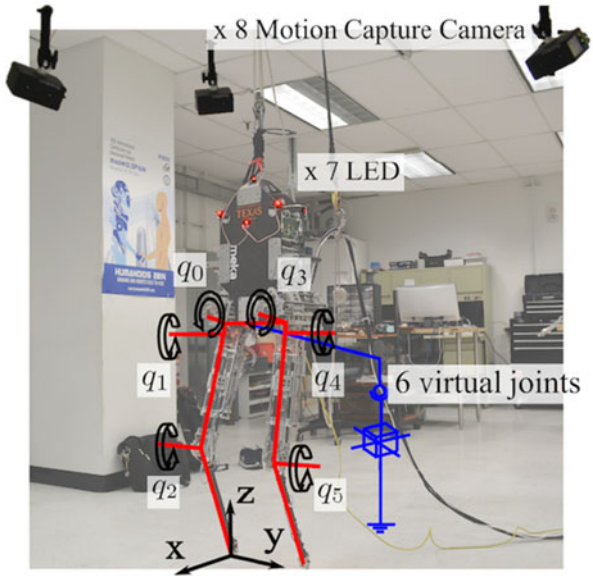


Fig. 1. Hume robot kinematics. Blue schematics describe the floating base joints, while black describes the kinematics of the six leg joints. The locations of the LED tracking markers identified by the PhaseSpace Impulse Motion Capture system are shown as red dots.

Microstrain 3DM-GX3-25-OEM IMU on our robot's torso measures angular velocity and linear acceleration, which is used in the state estimator. Additionally, the robot has a full overhead PhaseSpace Impulse motion capture system that gives it global coordinate information about seven uniquely identifiable light-emitting diode (LED) tracking devices mounted rigidly to the torso. The PhaseSpace system produces a data stream at 480 Hz and communicates to the Linux Control PC via a custom user datagram protocol. There is approximately 5 ms of delay in the feedback data. It accomplishes this using a system of eight high-speed sensors mounted on the ceiling above the robot and a proprietary software package to fuse their readings into a single estimate for the 3-D position of each marker. On each update, the system reports the location of as many of the uniquely identifiable LED markers as it can see in Fig. 1.

B. End-to-End Controller Architecture

The feedback control system is split into six joint-level controllers and a centralized high-level controller (see Fig. 2). This forms a distributed control system, where the joint controllers focus on high-speed actuator dynamics, while the centralized controller focuses on overall system dynamics. Yet, the high-level feedback is necessary to create the coupling between joints implied by operational space impedance tasks as well as regulating the internal forces between multicontact supports. In SEA control, we kept Meka's joint torque controller, which is based on passivity as described in [38] (shown in the lower right corner of Fig. 2).

C. Whole-Body Operational Space Control Algorithm

WBOSC [13] is a feedback control strategy based on operational space control [39], which extends it to floating base robots in contact with the environment. It allows the user to

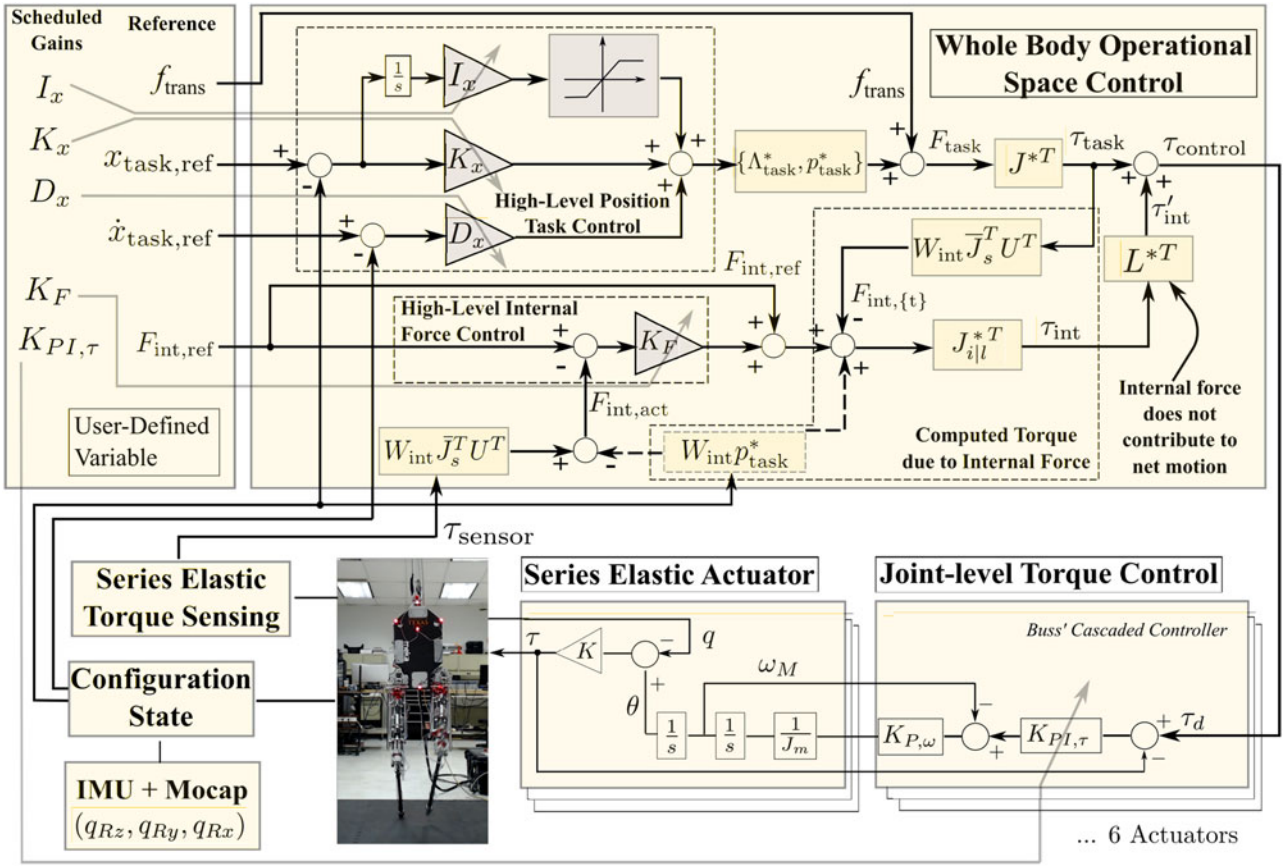


Fig. 2. Overall control diagram. This figure illustrates WBOSC and the joint-level torque controllers. One of our main contributions comes from the feedback control of internal forces. Note that the gains for the controllers are treated as additional input parameters to represent the gain scheduling for achieving the best possible performance of each task.

specify multiple task objectives and their impedance in operational space. It also subdivides the torques applied to the robot into orthogonal spaces that affect either robot motion or internal forces. When the user specifies these internal forces, the WBOSC achieves them using feedback. The full details of the WBOSC are explained in Appendix A.

At the implementation level, the WBOSC works well provided that communication latencies are sufficiently small. Achieving a 0.667-ms latency required significant software optimization. We modified Meka's firmware to ensure our controller operated within a real-time context, and to incorporate it into a hierarchical chain structure that ensures minimum latency for stacked control systems. We also reduced the basic computational cost of our WBOSC implementation by bypassing recursive dynamics software and instead using a closed-form expression to calculate the mass, gravity, and Coriolis matrices. To reduce the tracking error, we added an integral term to all position tasks.

D. Contact Switching Transitions

To reduce the high-speed behavior caused by a sudden change in joint torques, we devise a strategy that smoothens out the torque commands when the robot transitions between single and dual support. The sudden change in torque command is due to the instantaneous switch between constraint sets within the WBOSC. When the controller uses a single-contact constraint, it returns τ_{single}. When it uses a dual-contact constraint, it returns τ_{dual}. To ensure a smooth torque trajectory, we employ τ_{trans}, which is a special instance of the WBOSC that uses a single-contact constraint but adds an artificial transition force associated with the contact behavior of the transitioning foot. The transitioning foot corresponds to the foot that just landed on the ground or the foot that is about to lift up from the ground. In either case, we do not instantaneously switch the controller between single- and dual-contact phases, but instead transition smoothly from single contact to dual contact or vice versa. And the manner that we transition is by adding the artificial transition force to the single-contact controller and slowly increasing or decreasing its value until it matches the final torques of the next phase. That is, for the single-contact transitioning controller, we adjust (36) to include an artificial transitional force f_{trans} as

$$F_{\text{task}} = \Lambda^*_{\text{task}} u_{\text{task}} + \mu^*_{\text{task}} + p^*_{\text{task}} + f_{\text{trans}}. \tag{1}$$

Based on the dual-contact hypothesis τ_{dual}, we find the reaction force that would be generated on the foot that just landed on the ground or that is generated on the foot about to take off. Using the second row of the matrix (30), and based on the forces corresponding to the task set in single-contact phase resulting from the dual-contact phase reaction forces, i.e.,

$f_{\text{dual}\to\text{single}} \triangleq S_{\text{dual}\to\text{single}} F_r$, we obtain

$$f_{\text{dual}\to\text{single}} = S_{\text{dual}\to\text{single}} \left(\overline{J}_s^T \left[U^T \tau_{\text{dual}} - b - g \right] + \Lambda_s \dot{J}_s \dot{q} \right) \quad (2)$$

where $S_{\text{dual}\to\text{single}}$ is a projection operator that transforms the dual-contact reaction forces to equivalent task forces of the single-contact controller. Let us take the case of the swinging foot that just landed on the ground. Rather than switching instantaneously to the dual-contact controller, we continue using the single-contact controller with the artificial transition force. That force is smoothly increased until it matches the dual-contact reaction forces projected to the single-contact controller. Therefore, as the controller transition approaches to dual support, we increase the artificial force from $(f_{\text{trans}} = 0) \to (f_{\text{trans}} = f_{\text{dual}\to\text{single}})$. Conversely, after the controller switches from the dual-support phase to the single-support phase prior to lifting the foot from the ground, we smoothen the transition by gradually removing the artificial force $(f_{\text{trans}} = f_{\text{dual}\to\text{single}}) \to (f_{\text{trans}} = 0)$. To achieve this gradual transition, we define $f_{\text{trans}} \triangleq w \cdot f_{\text{dual}\to\text{single}}$, $w \in [0, 1]$, where w varies linearly with time from one to zero or from zero to one over the course of the transition as appropriate.

To avoid running the controller twice every servo cycle during transitions, we reuse previous values for τ_{dual} and $f_{\text{dual}\to\text{single}}$. When a foot lifts, the old value of τ_{dual} is simply the last controller action before the start of the transition, and $f_{\text{dual}\to\text{single}}$ is calculated once based on that controller state. When a foot lands, we run the dual-support controller once at the start of the transition for the sole purpose of acquiring $f_{\text{dual}\to\text{single}}$. This process is similar to [6] except that they apply it to a quadratic-programming (QP)-based controller, whereas we apply it to a WBOSC-based controller.

IV. Enhancement of Whole-Body Operational Space Control

A. Online Feedback Gain Adjustment

To enhance the bandwidth of the proposed WBOSC controller (shown in the upper portion of Fig. 2), we lowered the gains of the torque feedback controllers (shown in the lower portion of the same figure). A detailed study on stability and bandwidth tradeoff between high-level position control and low-level torque control is presented in [2]. In particular, reducing the low-level torque controller bandwidth allows us to increase the high-level position controller bandwidth without compromising stability. To achieve a higher bandwidth with position control, we increase the PID feedback gains in the WBOSC, reduce the joint-level torque feedback gains, and set the integral gains on the low-level torque controllers. Detuning torque gains would reduce the accuracy of internal force tracking performance if we would solely rely on feedforward models—see discussion in Section IV-B. For this reason, we design internal force feedback controllers to precisely track that subset of the contact forces—this technique was also discussed in Section IV-B.

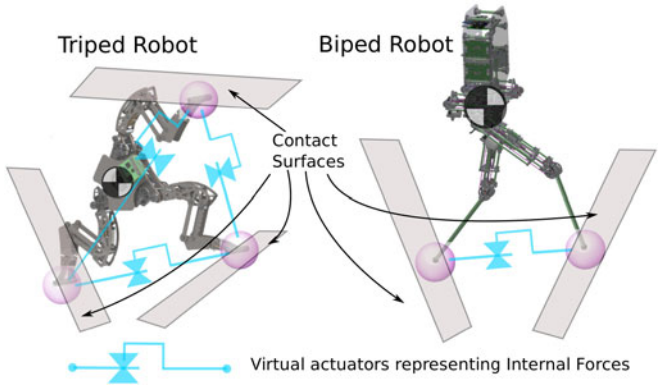


Fig. 3. Internal forces for various robots. Internal forces in point-foot robots correspond to tensions or compressions between pairs of supporting contacts.

One drawback of detuning torque controllers is that friction and stiction cause torque errors. This effect results in joints not moving when torque commands are small compared with the stiction threshold. To tackle this problem, we simultaneously adjust the torque controller gains and the WBOSC feedback controller gains based on each robot joint's effective load: In joints belonging to the stance leg, we turn OFF the integral gain of the torque controller. In joints belonging to the swing leg, we turn ON the integral gain of the same controller to reduce the effects of friction in the actuators.

This can be counterintuitive since larger effective mass implies larger feedback gain. Intuitively, the natural frequency of joint force output in SEA actuators decreases as the joint's effective load increases [40]. When one of the robot's legs is in contact with the ground, its effective mass increases, and as a result, its natural force frequency decreases. Applying integral torque gain in a controller with a small natural frequency can reduce the phase margin due to an increase in bandwidth and a drop in phase by -90°. After various trials, we concluded that a proportional feedback control with motor velocity feedback in the torque controller and a PID control in WBOSC gives the best performance for the stance leg.

B. Sensor-Based Internal Force Feedback Control

Internal forces are associated with joint torques that produce no net motion. As such, internal forces correspond to mutually canceling forces and moments between pairs or groups of contact points, i.e., tensions, compressions, and reaction moments. For instance, a triped point-foot robot has three internal force dimensions, while a biped point-foot robot has a single internal force dimension, as shown in Fig. 3.

Internal forces are fully controllable, since they are orthogonal to the robot's motion. As such, both the robot's movements and its internal forces can be simultaneously controlled. Moreover, in many types of contact poses, internal forces are easily identifiable using some physical intuition. For instance, in the triped pose of Fig. 3, the three feet can generate three virtual tensions between the points of contact. The physics of tension

forces were analyzed in greater detail using a virtual linkage model in [13].

Internal forces are part of the core WBOSC. In Appendix A, we describe the model-based control structures enabling direct control of internal forces. In particular, the basic torque structure derived in (43) is

$$\tau_{\text{int}} = J_{i|l}^{*T}\left(F_{\text{int,ref}} - F_{\text{int},\{t\}} + \mu_i^* + p_i^*\right) \tag{3}$$

where $F_{\text{int,ref}}$ is the vector of desired internal forces, and $F_{\text{int},\{t\}}$ corresponds to the mapping of task torques into the internal force manifold. The above equation would be sufficient for feedforward internal force control if the commanded torques were equal to the actual torques, and if the kinematic and dynamic models were exact. However, as we mentioned in Section I and will further explore in detail in Section IV-A, we intentionally lower the bandwidth of joint torque controllers in order to increase the bandwidth of WBOSC's task motion controllers. As a result, we cannot solely rely on feedforward control of internal forces. Formulating internal force feedback control is a significant advantage over increasing the bandwidth of joint torque controller. It enables good force tracking accuracy but only in the dimensions that are rendered of interest to the behaviors, which in our case correspond to internal forces. Because internal forces do not affect task motion, the internal force feedback controller does not negatively affect the accuracy of the motion tasks as the joint torque controllers do. Overall, it is best to formulate feedback control only in the dimensions that are practical, and in our case, they are the motion tasks in one hand and the internal force task on the other. To our knowledge, this is the first successful use of sensor-based feedback control of internal forces in a real robot.

Since internal forces are fully controllable, we can achieve better internal force tracking accuracy by incorporating a simple proportional controller of the measured internal force error via (43)

$$\tau_{\text{int}} = J_{i|l}^{*T}\Big(F_{\text{int,ref}} - F_{\text{int},\{t\}} + \mu_i^* + p_i^* \tag{4}$$

$$+ K_F\left(F_{\text{int,ref}} - F_{\text{int,act}}\right)\Big) \tag{5}$$

where K_F is a proportional control gain, and $F_{\text{int,act}}$ are the actual sensor-based internal forces. To obtain these sensor-based forces, we use the torque sensors on the SEAs to find the reaction forces as per (30) and apply a projection W_{int} to find internal forces

$$F_{\text{int,act}} \triangleq W_{\text{int}}\left[\overline{J}_s^T\left(U^T\tau_{\text{sensor}} - b - g\right) + \Lambda_s\dot{J}\dot{q}\right] \tag{6}$$

where τ_{sensors} corresponds to the vector of torques sensed by the spring element in each SEA (see Fig. 2).

The above internal force mapping is distinguished from previous work due to its sensor-based force feedback nature, and its mapping is valid due to the physical fact of robot redundancy in the multicontact case. The induced contact closed loop causes the number of controlled motion tasks to be smaller than that of actuated joints. Correspondingly, additional DOFs are available to be controlled for more force tasks, such as internal forces in (5). This mapped internal forces are consistent with contact constraints and cancellation of accelerations on the robot's base or on the actuated joints [13]. More details can be found in Appendix A.

To calculate internal forces for Hume, we need to define the mapping given in (41) in Appendix A, where W_{int} is the matrix representing the map from reaction forces to internal forces. In our case, Hume controls the internal forces between the two feet during dual-contact phases. In the dual-support mode, the reaction forces are $(f_{Rx},\ f_{Ry},\ f_{Rz},\ f_{Lx},\ f_{Ly},\ f_{Lz})^T$, where R and L mean the robot's right and left foot, respectively. According to [13], W_{int} consists of S_t, a selection matrix of tensions, R_t, a from global frame to the direction parallel to the line between two contact points, and Δ_t, a differential operator matrix, i.e.,

$$W_{\text{int}} = S_t\ R_t\ \Delta_t \tag{7}$$

with

$$S_t = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \tag{8}$$

$$R_t = \begin{pmatrix} \hat{x}^T \\ \hat{y}^T \\ \hat{z}^T \end{pmatrix}, \quad \begin{cases} \hat{x} = \frac{P_R - P_L}{\|P_R - P_L\|} \\ \hat{y} = (-\hat{x}(2), \hat{x}(1), 0)^T \\ \hat{z} = \hat{x} \times \hat{y} \end{cases} \tag{9}$$

$$\Delta_t = \begin{pmatrix} I_{3\times3} & -I_{3\times3} \end{pmatrix} \tag{10}$$

where P_R and P_L are the position of the right and left feet, respectively.

There are two ways to compute desired internal forces: 1) by approximating them via a simple force statics problem and choosing values that comply with friction cones; and 2) by first solving an optimization problem including task accelerations and friction cones to obtain desired reaction forces, $F_{r,\text{ref}}$, and then projecting them into internal forces using (41), i.e.,

$$F_{\text{int,ref}} = W_{\text{int}}\ F_{r,\text{ref}}. \tag{11}$$

In this paper, we use method 1. If one wants to use method 2, she/he can use existing methods such as the QP-based contact solver defined in [16] or the centroidal-momentum-based contact solver defined in [22]. Then, the equation above can be used to solve for the desired internal forces to be subsequently used as inputs to WBOSC.

V. Foot Placement Planner

The foot placement planner determines footstep locations based on a PIP model that is representative of the robot in single-contact phases. The footstep locations are selected such that the PIP model is stabilized. This should result in a stable balance behavior, assuming the trajectory generators and WBOSC 1) successfully place the feet at the desired locations, 2) achieve the desired height of the CoM, and 3) fix the orientation of the robot's body.

We previously devised a footstep location algorithm called the "phase space constant time to velocity reversal planner" [11]. The swing motion is separated into two phases: lifting and landing the foot. The landing location is computed before the landing phase starts. In every step, when the lifting phase reaches 70% completion, the planner computes the next footstep location.

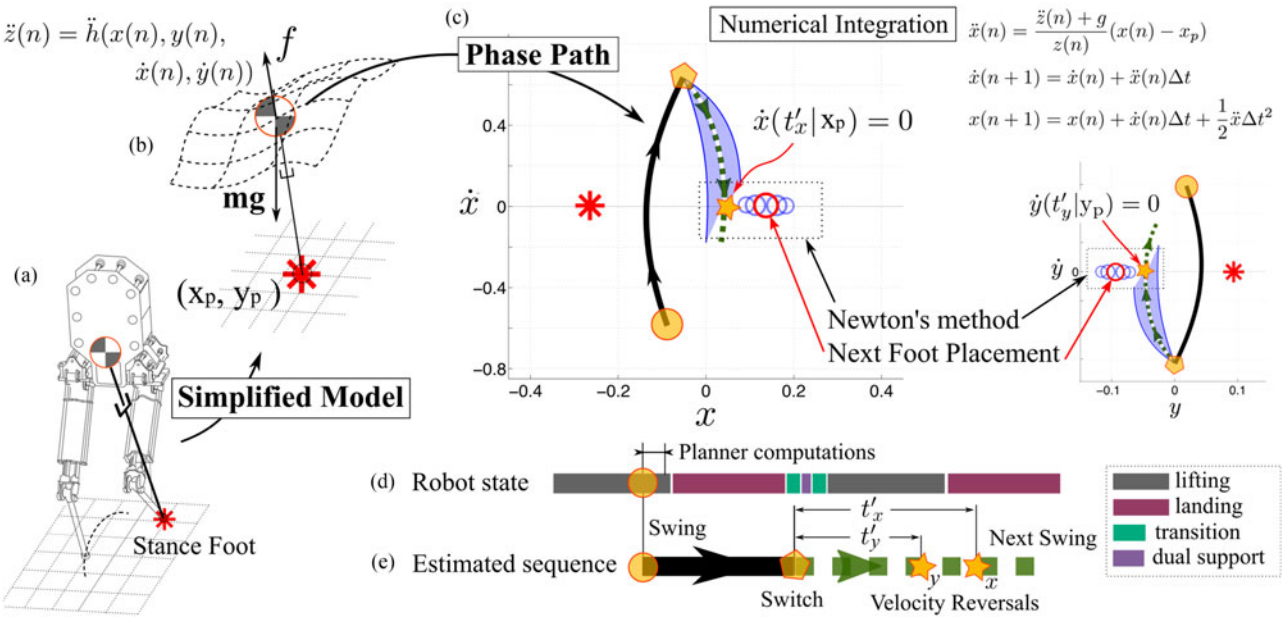


Fig. 4. Constant time to velocity reversal planner. As shown in (a), we approximate the dynamics of the robot with the PIP model, shown in (b). This model predicts the dynamics of the horizontal CoM position x, y given the stance foot location $(x_p,\ y_p) =$ ● and the height surface $z = h(x,\ y)$. This can be integrated forward in time via the numerical integration procedure shown in (c). When the planner starts operating it records the initial state ✱ and integrates this state forward to determine the switching state ⬟. As shown in timelines (d) and (e), the "Estimated sequence" of the planner has an analogue in the "Robot states" of the state machine. In particular, the switching state ⬟ roughly corresponds to the dual-support phase of the walking state machine. This state ⬟ represents the planner's guess at the time and state $(x, \dot{x}, y, \dot{y})$ values immediately after the switch. The goal of the planner is to stabilize the robot, and this is achieved by choosing the next footstep ○ such that x and y velocity equal zero t'_x and t'_y seconds, respectively, after the foot switch every step. For sufficiently smooth height surfaces, the relationship between the next footstep location and the velocity is monotonic, so only a single solution exists. We use Newton's method with numerical differentiation to identify this solution. There exist two velocity reversal states: ★$_x$ and ★$_y$, as shown in (c).

The value of 70% was empirically determined to ensure that the planner completes before the landing phase starts. It is a processor intensive task that must be run outside of the real-time thread. The operational space set-point trajectory for the swing foot is then defined based on a polynomial function and the desired landing position, with the trajectory ending once ground contact is sensed. If the ground is at the expected height and the position tracking is ideal, the footstep will land after the nominal swing time. If the planned step is outside the mechanical limits of the robot, the planner chooses the closest reachable step.

A. *Velocity Reversal*

Our planner attempts to stabilize the robot by making its CoM reverse direction every step. In its simplified model, the feet instantly change between swing and support modes, and the CoM reverses its velocity a time t' after the previous contact switch.

In contrast with approaches based on linear models with analytically solvable dynamics such as the capture point or the divergent component of motion methods, our approach is different because the PIP model is not analytically solvable and numerical search is used to plan the steps. This leads to the primary computational element in the planning procedure: a shooting method over the possible footstep locations to find a CoM trajectory that accomplishes the CoM velocity reversal goal. In a previous paper [11], which only considered constant elevation of the CoM, we used bisection search to find footstep locations. In this new work, which considers variable elevations, we use Newton's method based on numerical derivative approximations to find the footsteps.

As illustrated in Fig. 4, the planner begins calculating the landing location when 70% of the lifting phase is reached. As such, the planner continuously replans to correct for trajectory deviations. Using the current estimate of the CoM velocity and position, it numerically integrates forward until the predefined swing time ends to predict its CoM position and velocity when its stance foot and swing foot will switch roles, ⬟ in Fig. 4. The equation used in the numerical integration is presented in Appendix B.

The implementation of the planner enforces the choice of t'. This time value remains constant for every step; thus, the user only needs to specify a single parameter. The method for selecting t' is explained in Section V-B. The planner then finds and returns the footstep location that causes the robot's CoM velocity to reach zero t' seconds after the foot switch. For each potential footstep location considered, the planner integrates forward in time starting from the postimpact state, as suggested in Fig. 4, ensuring the velocity is zero after t' seconds. This integration can be viewed as a function mapping footstep location to a future velocity. Newton's method uses this function to find a foot placement that results in zero CoM velocity. The number of integrations performed is very low, since the process relies on the monotonicity of the relationship between footstep

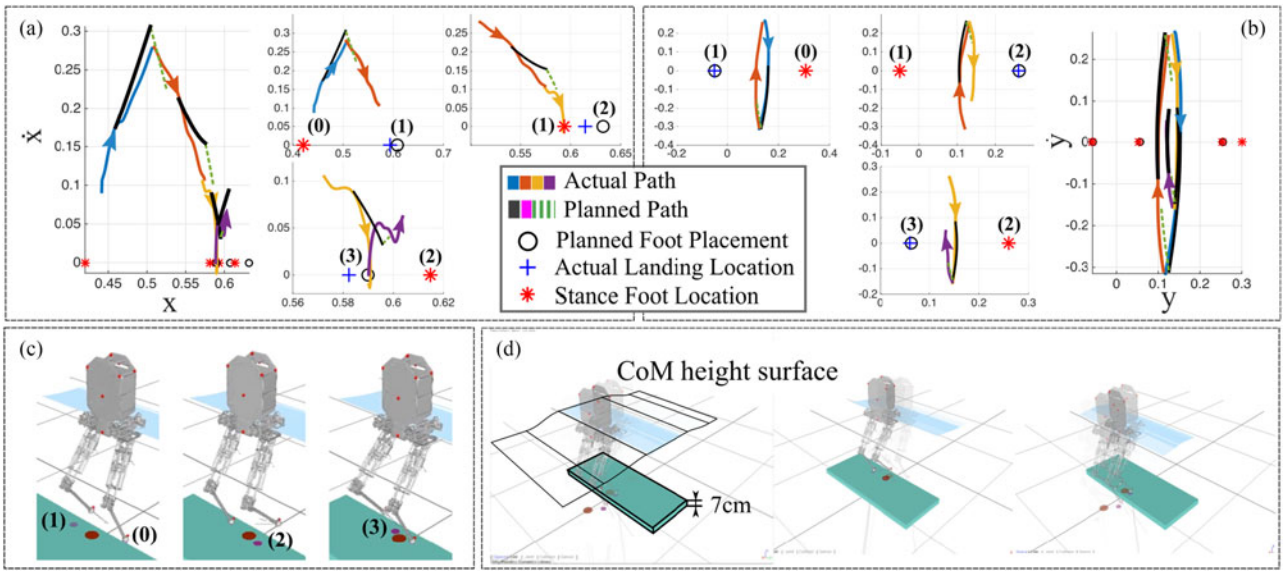


Fig. 5. Simulation of unsupported point-foot walking over a rough terrain. (a) Sagittal CoM phase portrait of three steps of walking over a rough terrain. (b) Lateral CoM phase portrait. The smaller plots correspond to individual steps. The corresponding three steps are shown in (c) using the SrLib multibody dynamic simulation environment. (d) More steps of the dynamic walking simulation over the rough terrain.

location, x_p and y_p, and the velocity after t' seconds, $\dot{x}(t'_x|x_p)$ and $\dot{y}(t'_y|y_p)$ [11].

As shown in Fig. 5 Velocity Reversal, our planner allows Hume to step over a 7-cm-tall platform and using a CoM height surface that conforms to the terrain. Specifically, this height surface is defined, piecewise, as a function of a global x coordinate, with three constant height pieces connected by two sinusoidal segments. The middle constant height piece is 7 cm above the others to account for the obstacle's height. The surface maintains first-order continuity. In this simulation, the robot's planner follows a moving goal location. As this goal location passes over the platform, the robot ascends and descends it while constantly stepping. Although the planner usually finds the proper foot placement, the robot shows variable forward and backward swinging motions. This variability arises when the robot performs large motions such as stepping down from a platform. Since the simplified PIP model used for planning does not account for the multilimb robot dynamics, predicting the CoM path becomes increasingly more difficult when estimating large motions. Despite this discrepancy, the planner and controller successfully achieve a stable walk over the challenging terrain.

In the simulation, we assume zero time delay, perfect sensor data, perfect torque tracking, and correct dynamic and kinematic models. These assumptions, which are difficult to obtain in the real world, make it easier to check the basic functionality of the planner.

B. Stability Analysis

It can be shown that t' is stable for a limited range of values assuming linear height surfaces. A natural property of our planner is that the footsteps converge toward the location of the CoM projected to the ground. In practice, this is not a desirable behavior, since the feet would move too close to each other making the robot more sensitive to small disturbances. To account for this potential problem, we create a hybrid behavior—if the planned velocity at the next transition will be too small, we artificially extend the swing time, which keeps the dynamics away from converging to the origin. In this section, we only consider the system's stability when this convergence effect is not present.

We begin by formulating the well-known linear inverted pendulum model

$$\ddot{x} = \frac{g}{h}(x - x_p) \tag{12}$$

where g, h, and x_p are gravitational acceleration, the constant CoM height, and the stance foot location, respectively. The solution of this ordinary differential equation is

$$x(t) = x_p + (x_0 - x_p)\cosh(\omega t) + \frac{\dot{x}_0}{\omega}\sinh(\omega t) \tag{13}$$

where $\omega = \sqrt{g/h}$, and can be expressed as a discrete time state-space system with a constant step duration T as

$$X((k+1)T) = AX(kT) + Bx_{p,k}, \quad k \in \mathbb{Z} \tag{14}$$

where

$$A = \begin{bmatrix} \cosh(\omega T) & \omega^{-1}\sinh(\omega T) \\ \omega\sinh(\omega T) & \cosh(\omega T) \end{bmatrix} \tag{15}$$

$$B = \begin{bmatrix} 1 - \cosh(\omega T) \\ -\omega\sinh(\omega T) \end{bmatrix}. \tag{16}$$

Here, $X(kT) = \begin{bmatrix} x_{kT} & \dot{x}_{kT} \end{bmatrix}^T$ represents the state at the instant when the input changes—the instant the stance foot and swing foot change roles, and a new foot position x_p is put into place.

As explained previously, the planner chooses x_p such that the CoM velocity becomes zero at time t'. This can be expressed as

a function of the state of the discrete system

$$0 = \dot{x}_{kT+t'} = \begin{bmatrix} \omega \sinh(\omega t') & \cosh(\omega t') \end{bmatrix} X(kT) - \omega \sinh(\omega t') x_{p,k} \quad (17)$$

$$x_{p,k} = \begin{bmatrix} 1 & \omega^{-1} \coth(\omega t') \end{bmatrix} X(kT). \quad (18)$$

However, we include an additional linear bias term in the control law above

$$x_{p,k} = x_d \kappa_p + \begin{bmatrix} (1-\kappa_p) & \omega^{-1} \coth(\omega t') \end{bmatrix} X(kT) \quad (19)$$

to move the robot toward a goal location x_d, effectively creating a locomotion behavior.

Without loss of generality, we assume that the goal is the origin for the stability analysis. The closed-loop system under the proposed feedback law is

$$X((k+1)T) = (A + BK)X(kT) \quad (20)$$

$$K = \begin{bmatrix} (1-\kappa_p) & \omega^{-1} \coth(\omega t') \end{bmatrix} \quad (21)$$

$$A + BK = A' \quad (22)$$

$$A'_{11} = 1 - \kappa_p + \kappa_p \cosh(\omega T) \quad (23)$$

$$A'_{12} = \omega^{-1}(\sinh(\omega T) + (1 - \cosh(\omega T)) \coth(\omega t')) \quad (24)$$

$$A'_{21} = \kappa_p \omega \sinh(\omega T) \quad (25)$$

$$A'_{22} = \cosh(\omega T) - \sinh(\omega T) \coth(\omega t'). \quad (26)$$

Using the eigenvalues of the matrix A', we determine the stability of the closed-loop system. To be robust to model uncertainties, we chose parameters κ_p and t' such that they produce eigenvalues with magnitudes close to 0.8 for the desired h and T values. One interesting fact of t' is that an infinitely large t' stops the robot's CoM exactly at the stance foot position, which is equivalent to the capture point [9]. However, this is not a desired behavior because when the CoM is near the stance leg, crossing of the legs might occur. Thus, we design the planner parameters to gradually decrease the robot's CoM velocity at the moment when the stance leg switches to the other leg. We achieve this decrease in velocity by setting the magnitudes of the eigenvalues of A' close to 0.8.

C. Impact Model

In many cases, the hybrid dynamics of a robot impacting the ground are significant. This warrants a model that includes a discrete map to represent the sudden velocity changes that occur during impact [41]. However, since our planning algorithm focuses exclusively on the CoM behavior, and since the body mass of our robot is much larger than the leg mass, we used a model that predicts no change of the CoM velocity during impact. Here, we use the approximations that the SEAs of our robot decouple the reflected rotor inertias, that the actuators are frictionless, that the robot's upper body mass is fairly decoupled from the foot due to the leg's kinematic chain, and that the knee of the landing leg is not entirely stretched.

VI. Experimental Results and Assessment

A. Unsupported Dynamic Balancing

To start the experiment, Hume is briefly supported, while it rises to the desired height for balancing. Once it reaches its starting height, the experimenter balances the robot carefully and lets it go as it takes its first step. Once free, the robot continuously steps until it falls over. There is a harness rope, slack when the robot is at its starting height, which catches it if it falls to prevent major damage. The power and Ethernet tether hang slack from another rope.

The motion follows a time-scripted state machine, shown in Fig. 6(a). Since the states are symmetric with respect to the supporting leg being either right or left, states are categorized in two different compound tasks with left and right single support having symmetric structures. The WBOSC compound task, x_{task}, differs between dual-support and single-support phases of the stepping state machine. In dual support, the compound task coordinates are $[\text{CoM}_z,\ q_{Rz},\ q_{Ry},\ q_{Rx}]^T$, where CoM_z represents the height of the CoM. q_{Rz}, q_{Ry}, and q_{Rx} are body yaw, pitch, and roll angles, respectively. Those coordinates are controlled via the acceleration input of WBOSC, u_{task}, shown in (36) and via PD or PID control laws. In single support, the compound task is $[\text{CoM}_z,\ q_{Ry},\ q_{Rx},\ \text{foot}_x,\ \text{foot}_y,\ \text{foot}_z]^T$. The desired height is set to the initial height when Hume begins to step, and the body orientation is set to be straight up. The desired foot trajectory for the lifting phase consists on first reaching a predefined z height while keeping x and y constant. Then, a third-order polynomial is used to generate the desired (x, y, z) landing trajectory. Since Hume's feet are points, in the single-support phase, it is not possible to control the yaw motion, q_{Rz}. To compensate for this deficiency, we use the brief time that the robot spends in dual support to correct for it. In the balancing experiment, we do not use internal force feedback control to reduce complexity of the sequencing process. Since the standing surfaces are flat, internal force regulation is not needed to keep the contacts stable. All control parameters for the single- and dual-support phases, and for the stance and swing legs, are shown in Table I. The parameters of the planner used in this experiment are shown in Table II.

In Fig. 6(b) and (c), the x- and y-directional CoM trajectories (red) are superimposed on the one-step predicted path by our planner (blue). Although predicting CoM path for multiple steps is difficult, computing the CoM path for a single step using the PIP model closely approximates the actual CoM motion. The orientation error is bounded by 0.05 rad [see Fig. 6(d)]. Given the model disturbances and the impacts, this error is reasonable small to validate the controller's performance. Fig. 6(e) shows snapshots and phase paths for this experiment. The phase space data are corrupted by high-frequency noise from the IMU sensor signal and the joint encoders that combined to compute CoM velocity.

Transition and gain scheduling techniques also play an important role by smoothing the motion and tracking the tasks. In Fig. 7, the commanded torque (blue) changes from 0 to -60 N$\cdot$ m without significant jerk due to our contact transition technique. When the right leg switches to a stance leg (green

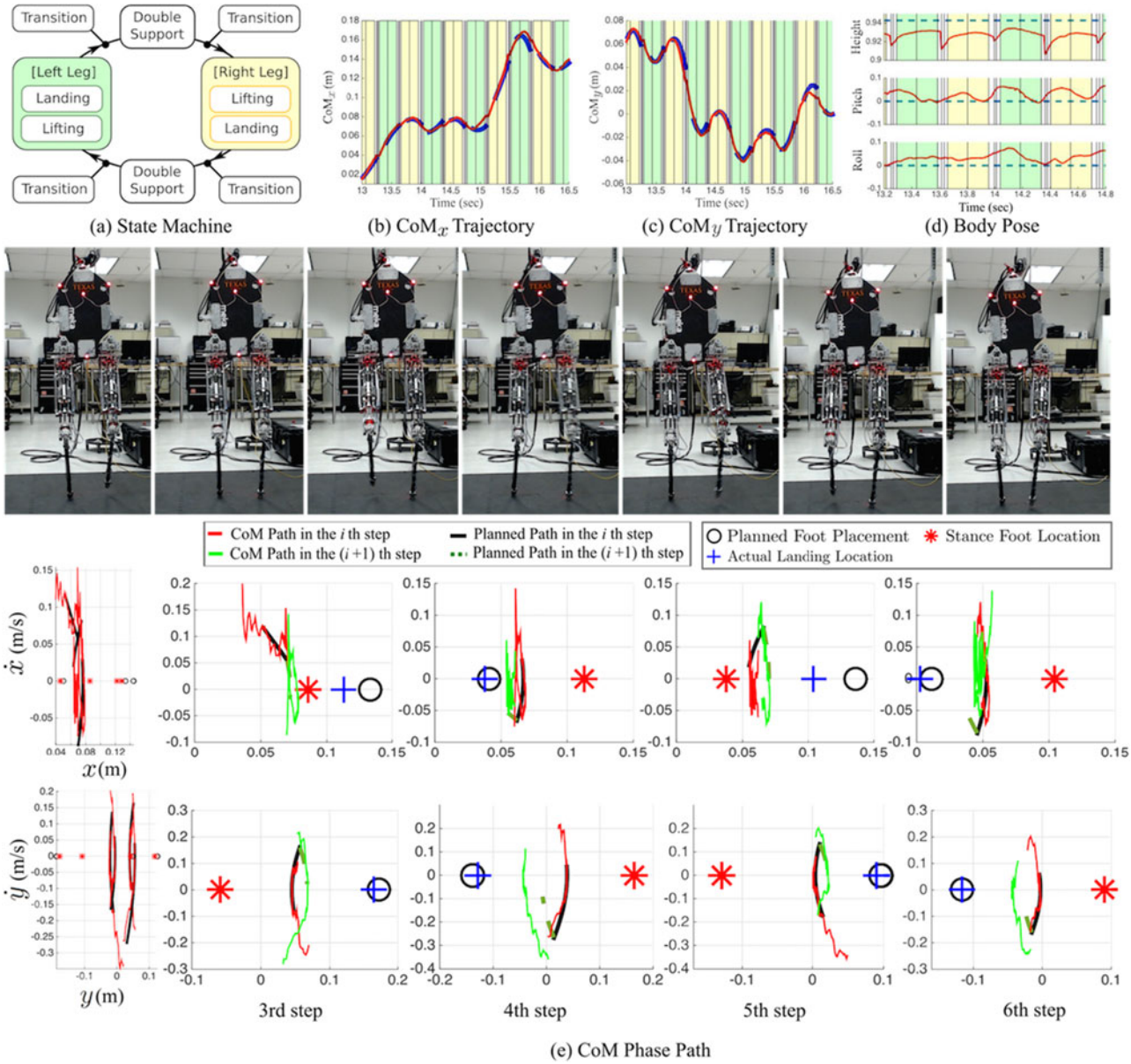


Fig. 6. Unsupported dynamic balancing experiment. This figure shows a portion of an experiment in which our point-foot biped robot, Hume, accomplishes 18 steps of unsupported dynamic balancing. (a) Motion within the balancing experiment is divided into three states: dual support, swing leg lift, and swing leg landing. Additionally, a transition state exists at the beginning of the lifting phase and the end of the landing phase to avoid the jerk caused by a sudden change in controller constraints. (b) and (c) x- and y-directional trajectory of the CoM. The blue line represents actual data, while the red lines indicate the trajectories estimated by the planner. (d) Position and orientation task tracking is plotted over a representative portion of the stepping experiment, with sensor data in red and desired values, dotted, in blue. Height refers to CoM height. (e) Snapshots and data of steps 3–6 are shown. The phase paths of steps 3–6 are expanded into individual step planning plots. For each step, a red line marks the actual CoM path up to the switching state, and a green line continues the trajectory after the switch. The robot's initial stance foot in the step planning plot is denoted with ●, the planned second footstep with a black circle, and the achieved second stance foot location with a blue cross. Therefore, a green line in the ith step plot is the same path as the red line in the $(i+1)$th step plot. A black line and a dark green dotted line are, respectively, the PIP model's predicted paths before and after the switching state.

background), around 20 N· m of torque tracking error appears. This is expected because we detune the low-level torque controller to achieve stiffer position control by the WBOSC. The controller corrects yaw error during the dual-support phase. Additionally, Hume incurs a significant bending of the stance leg, which results on uncertainty of the position of the CoM with respect to the stance foot. Despite all these sources of error, our robot was able to dynamically balance unsupported for 18 steps using its point contacts.

B. *Balance on a Disjointed Terrain*

In this experiment, Hume balances on a high-pitch terrain composed of two 45° wedges angled in toward the robot to create a convex floor profile. Because there is no way to control lateral motion, the planarizer is used to constrain the motion of Hume to the sagittal plane.

Note that we do not include the 5-kg sliding linkage in the dynamic model and regard it as unmodeled disturbance.

TABLE I
GAIN SET FOR THE UNSUPPORTED DYNAMIC BALANCING EXPERIMENT

Position Gain										
	Dual Support					Single Support				
	CoM_z	q_{Rz}	q_{Ry}	q_{Rx}	CoM_z	q_{Ry}	q_{Rx}	$foot_x$	$foot_y$	$foot_z$
$K_x\,(s^2)$	270.0	100.0	200.0	210	270.0	200.0	210.0	220.0	237.0	400.0
$I_x\,(s^3)$	20.0	0.0	20.0	35.0	20.0	20.0	35.0	35.0	40.0	35.0
$D_x\,(s^1)$	10.0	0.0	15.0	10.0	10.0	15.0	10.0	35.0	40.0	60.0

Torque Gain								
	Stance Leg			Swing Leg				
	Both Abduction	Both Hip	Both Knee	Both Abduction	Right Hip	Left Hip	Right Knee	Left Knee
$K_{P,\tau}$ (N· m· rad· s^1)	2.84	43	22	3.09	58.57	32.70	49.75	26.1
$K_{I,\tau}$ (N· m· rad· s^1)	0.15	0	0	0.19	5.86	3.48	6.85	4.15

TABLE II
PLANNER PARAMETERS FOR THE UNSUPPORTED DYNAMIC BALANCING EXPERIMENT

Transition	Lifting	Landing	Dual Support	κ_x , κ_y	t'_x , t'_y	λ_x	λ_y
0.024 (s)	0.145 (s)	0.15 (s)	0.016 (s)	0.4, 0.4	0.185, 0.18	0.7, −0.88	0.69, −0.81

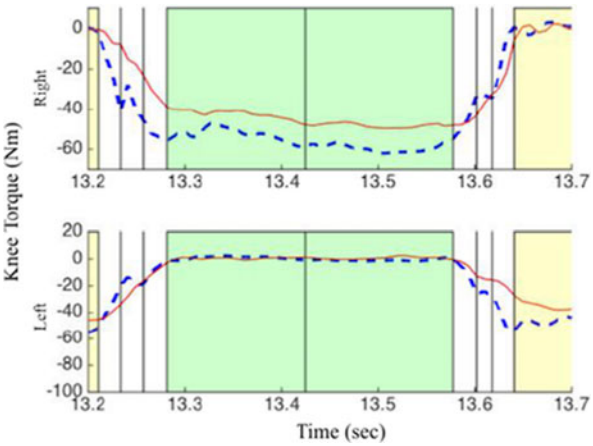


Fig. 7. Knee joint torque. The plot shows torque trajectories of right and left knee joints. Blue dotted lines are commanded torque and red solid lines are measured torque. Color indication of the state machine in Fig. 6 is used.

The robot's tasks were to maintain a 100-N internal force pushing outwards against the two contact points, a desired impedance task for the CoM, and a desired impedance task for the body orientation. The reference of 100 N is roughly calculated with the assumption that the 200-N body weight is equally distributed between each foot and a 100-N horizontal force generates a reaction force normal to the 45° angled surfaces.

By controlling the internal force, Hume does not slip, while it tracks the reference CoM path within a 2-cm error. This experiment is divided into two subexperiments: Hume was controlled to track a time-varying CoM trajectory, which followed an elliptical path in the sagittal plane, as shown in Fig. 8(b), and Hume was controlled to hold a Cartesian impedance task on the CoM, which had low stiffness horizontally and high stiffness vertically, as shown in Fig. 8(c). In the test, $x_{\mathrm{task}} = [\mathrm{CoM}_x,\ \mathrm{CoM}_z,\ q_{Ry},\ q_{Rx}]^T$, $F_{\mathrm{task,ref}} = [0]_{4\times 1}$, and $F_{\mathrm{int,ref}} = 100\ N$, using the notation of Fig. 2. We do not control CoM_y and q_{Rz}, since the planarizer rigidly constrains those directions of motion. However, we do control the robot's roll, q_{Rx}, since the planarizer is slightly flexible in the roll direction. Thus, controlling the roll and pitch helps to sustain the body pose.

CoM tracking performance is shown in Fig. 8(b). Although tracking errors exist, they are bounded within 2 cm. We believe this error occurs because of various reasons.

1) The legs of Hume are flexible and that flexibility is not modeled.
2) The planarizer that holds Hume on the back is unmodeled.
3) The low-cost IMU that we use suffers from quick orientation drift.

The second subexperiment shows that feedback control of internal forces allows Hume to stay balanced despite external forces exerted by a human. Due to the WBOSC's internal force feedback controller, the disturbances we exert on the robot do not produce large deviations in the internal force tracking performance.

Due to feedback control, the errors between desired (blue) and actual (red) internal forces are small enough to keep Hume balance on the difficult terrain. In contrast, in Fig. 9, we turned OFF the feedback control of internal forces. The experiment was conducted on a 20° pitch terrain due to the robot being unable to stay up on the 45° pitch terrain when turning OFF the feedback. We can see that the error to command ratio is much larger when using feedforward control only (more than 50%) relative to the experiment with the feedback enabled (less than 20%). These experiments ultimately show the effectiveness of

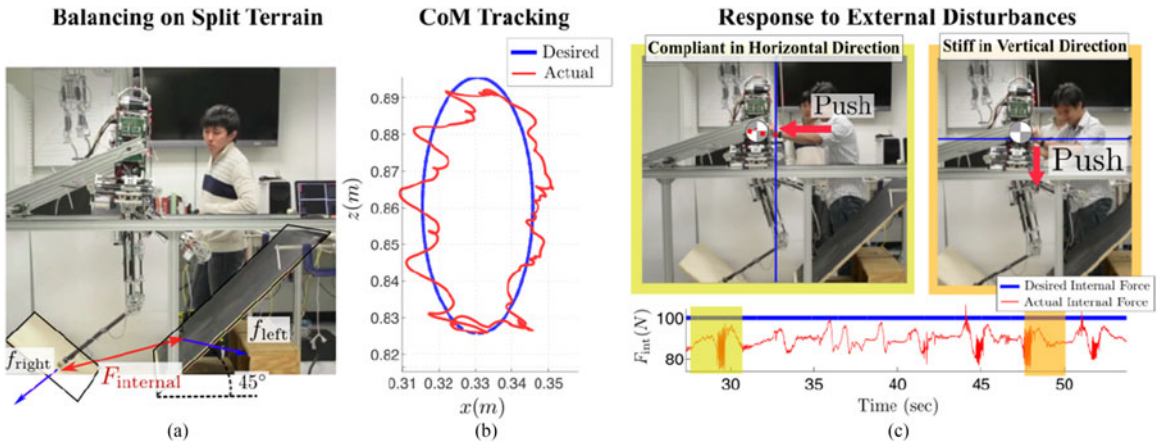


Fig. 8. Trajectory tracking and human disturbance rejection on a disjointed terrain. (a) Hume standing between two inclined wooden panels and tracking a position task with its CoM. This position set point follows a constant velocity trajectory along an elliptical path shown, along with the measured CoM path, in (b). In (c), the CoM has different impedances in the horizontal and vertical directions. When the robot is pushed backwards, it moves as though the CoM were connected to a low-spring-constant spring, whereas when the robot is pushed downwards, it reacts as though connected to its set point by a far stiffer spring. Due to the feedback regulation of internal forces, the biped does not fall down when disturbed with large external forces.

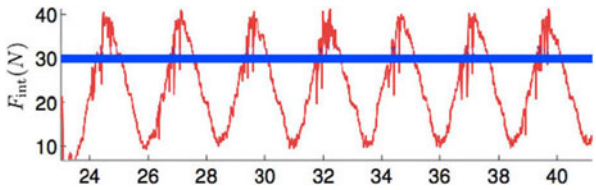


Fig. 9. Internal force control without feedback. Without feedback loop, internal force control shows larger than 50% error.

closed-loop internal force control as applied to maintaining a frictional contact.

VII. Discussion and Conclusion

The central focus of our paper has been on formulating a WBOSC and a new balance planner to achieve unsupported dynamic balancing of a point-foot biped robot without passive feet. The underactuated nature of point-foot bipeds forces us to "give up" on the x and y components of CoM motion, leaving those aspects to evolve naturally, while the WBOSC controls the remaining DOFs. These liberated DOFs are still controlled, but must be controlled indirectly, on a step-by-step basis, by choosing the footstep locations. By formulating a WBOSC and a phase-space-based foot placement planner, we make Hume, our 6-DoF point-foot biped robot with SEAs, balance on the rough terrain when constrained to move in the robot's sagittal plane, and take 18 steps on the flat terrain when unsupported. In addition, our framework also demonstrates its ability to step over a 7-cm obstacle in simulation. Achieving these capabilities required advancements in the WBOSC that push its performance boundaries to the next level by stabilizing highly dynamic biped robots and showing the benefits of employing feedback control of internal forces.

The WBOSC uses the robot's dynamic model to simultaneously control motion and force behaviors in the task dimensions that are rendered practical for the operations at hand. For example, the WBOSC on a torque-controlled robot can precisely control the position of the robot's CoM, while also precisely controlling the internal forces—in the sense of the tensions and compressions performed with the entirety of the robot's body—between disjointed surfaces. And only when joint torque sensing is available, these types of sophisticated behaviors can be achieved without reliance on force sensors located on the end-effectors. More importantly, using the same idea, the WBOSC on torque-controlled robots could control the internal forces between any two points in the robot's structure, provided that they are distant enough from each other. In our case, we employed this unique force/motion feedback control capability, to balance our biped robot on a high-pitch split surface. Beyond contacts on the feet or end-effectors, the WBOSC aims at achieving contact awareness and command over the entire robot's body structure. And as such, it may one day be the ideal type of controller for walking or moving in direct contact with objects and people.

An important issue of this paper has been the integration of WBOSC with joint-level torque controllers. To obtain good performance, torque control at the joint level has been sought. The reason is that joint torque control reduces the effect of the natural dynamics of the actuator, i.e., stiction, dynamic friction, nonlinearities, thus yielding excellent force behaviors for the actuator. However, using torque-feedback control nested within a position feedback controller needs careful understanding of its stability as a function of loop latencies [2]. Torque controllers substantially decrease the effect of natural friction of the actuators, forcing the task position controllers to provide damping via velocity feedback. However, velocity feedback is very sensitive to loop latencies as was analyzed in detail in [42]. In order to achieve stiff task position control, which is desirable for some tasks to reject modeling disturbances, system latencies both due to the WBC and to communications need to be minimized. In the case of Hume, we have greatly optimized all control computations and embedded communications to decrease the servo round trip, including communications, to 0.667 ms.

Additionally, when there is a need to deliver stiff position tracking control, it is desirable to detune or reduce torque gains. This effect was reported in [43] and more recently analyzed in depth in our paper [2]. In essence, we prioritize position accuracy over torque accuracy when position tracking is essential. To compensate for the detuned torque controllers, when our biped balanced on the split terrain, we added a force feedback loop that regulated the internal force from joint torque sensor. This feedback loop mitigated the effects of trading off motion accuracy versus torque accuracy. The key advantage is that increasing internal force control gains does not affect the task position control performance, whereas increasing torque gains reduces task position performance.

Finally, exact state measurement has been crucial to maneuvering with the highest possible accuracy given the hardware limitations of our low-cost IMU. However, many successful legged robots, such as Atlas, ATRIAS, and the MIT Cheetah 2 robot, use an optical IMU called KVH Industries Model 1750, which has a minuscule bias error, 0.00055 °/s. This is not the case in our experimental setup, as we used a low-cost and much less accurate MEMS IMU, i.e., the Microstrain 3DM-GX3-25-OEM, with a bias error 500 times worse than the previous one, i.e., 0.25 °/s. In other words, our IMU's orientation estimate drifts away quickly. In practice, the drift appears to be much faster than the specification. To compensate for this discrepancy, we devised a sensor fusion approach by combining the IMU with visual data from our motion capture system. Nonetheless, the overall pose sensing system is far less accurate and speedy than using the high-end IMU. In the future, we plan on replacing our current IMU with the high-end version KVH IMU and also redesigning part of the leg joints to increase mechanical rigidity. With those improvements, we believe that we will substantially increase the accuracy of the foot positioning, enabling faster and more robust locomotion behaviors.

APPENDIX A
WHOLE-BODY OPERATIONAL SPACE CONTROL

The WBOSC was first laid out in [12] with a further exploration of internal forces that was published later in [13]. Interested readers should refer to these sources for a description of the theoretical background complete with proofs for the concepts below, as space limits us to a cursory overview of WBOSC as applied to bipedal robots.

Modeling biped robots entail representing not only the state of each joint $q_1, \ldots, q_{n_{\text{joints}}}$, but also the states of the robot as an object in space. We choose to parameterize the robot's base as a 6-D floating joint between the world and the robot's hip coordinate system with state vector $q_b \in \mathbb{R}^6$. Combining the robot and base states into a single vector, we arrive at $q \in \mathbb{R}^6 \oplus \mathbb{R}^{n_{\text{joints}}} = \mathbb{R}^{n_{\text{dofs}}}$, the generalized joint vector. The joint torques can only directly affect the joints themselves, and not the floating base dynamics; therefore, we define an underactuation matrix $U \in \mathbb{R}^{\left(n_{\text{dofs}}-6\right)\times n_{\text{dofs}}}$ that maps the global joint vector to the subspace of actuated joints

$$q_{\text{act}} = U\ q \tag{27}$$

with $q_{\text{act}} \in \mathbb{R}^{n_{\text{joints}}}$ being the robot's actuated joints. The dynamics of the robot's generalized joints can be described by a single, linear, second-order differential equation

$$A\ddot{q} + b + g = U^T \tau_{\text{control}} \tag{28}$$

where $\{A, b, g\}$ represent the inertia matrix, Coriolis/centrifugal forces, and gravitational forces, respectively, while τ_{control} is the desired control command applied to the output joints of the robot. Without considering contacts, or the subtle nonholonomic effect of angular momentum, joint torques would have no effect on the geometrically uncontrollable 6-D subspace of the generalized joints: $\{z \in \mathbb{R}^{n_{\text{dofs}}} : z^T A^{-1} U^T = 0\}$. However, we can sometimes gain the ability to control more of this space due to contact constraints.

We consider two contact cases for point-foot biped robots: single support in which one foot is in contact, and dual support where the robot is supported by both feet. In the single-support phase, we describe the contact via a support Jacobian $J_s \in \mathbb{R}^{3\times n_{\text{dofs}}}$, which maps from generalized joint velocity to the velocity of the constrained foot point in Cartesian space. When considering dual contact, our support Jacobian represents twice as many constraints. Since this generalized point, either the single foot point in $\mathbb{R}^3$ or the dual foot point in $\mathbb{R}^6$, is constrained, we know its acceleration must be zero. Substituting the constraint $J_s\ddot{q} + \dot{J}_s\dot{q} = \ddot{x}_{\text{foot(or feet)}} = 0$ and adding the associated costate of constraint space reaction forces F_r, the dynamics become

$$A\ddot{q} + b + g + J_s^T F_r = U^T \tau_{\text{control}} \tag{29}$$

and we can find $\ddot{q}$ and F_r by solving the matrix equation

$$\begin{pmatrix} A & J_s^T \\ J_s & 0 \end{pmatrix}\begin{pmatrix} \ddot{q} \\ F_r \end{pmatrix} = \begin{pmatrix} U^T\tau_{\text{control}} - b - g \\ -\dot{J}_s\dot{q} \end{pmatrix}. \tag{30}$$

Converting to upper diagonal form via Gaussian elimination, we find

$$\begin{pmatrix} A & J_s^T \\ 0 & I \end{pmatrix}\begin{pmatrix} \ddot{q} \\ F_r \end{pmatrix} = \begin{pmatrix} U^T\tau_{\text{control}} - b - g \\ \overline{J}_s^T\left[U^T\tau_{\text{control}} - b - g\right] + \Lambda_s\dot{J}_s\dot{q} \end{pmatrix} \tag{31}$$

with $\Lambda_s \triangleq \left[J_s A^{-1} J_s^T\right]^{-1}$ and $\overline{J}_s \triangleq A^{-1}J_s^T\Lambda_s$. Substitute $N_s^T \triangleq I - J_s^T\Lambda_s J_s A^{-1}$ to more conveniently express the resulting constrained dynamic equation

$$A\,\ddot{q} + N_s^T\,(b+g) + J_s^T\Lambda_s\dot{J}_s\dot{q} = \left(UN_s\right)^T\tau_{\text{control}}. \tag{32}$$

This can be viewed as constraining the dynamics to the dynamically consistent null space of the constraint by defining the dynamically consistent pseudoinversion operator

$$\overline{X} \triangleq A^{-1}X^T\left[XA^{-1}X^T\right]^{-1} \tag{33}$$

and observing that

$$N_s = I - \overline{J}_s J_s \tag{34}$$

is the null-space projector of J_s under dynamically consistent inversion such that $J_s N_s = 0$, $N_s A^{-1} J_s^T = A^{-1}N_s^T J_s^T = 0$, and $N_s N_s = N_s$.

The WBOSC for an operational task representation, p_{task}, is defined by the differential kinematic equation

$$\dot{p}_{\text{task}} = J^*_{\text{task}}\, \dot{q}_{\text{act}} \tag{35}$$

where $J^*_{\text{task}} \triangleq J_{\text{task}}\overline{UN}_s \in \mathbb{R}^{n_{\text{task}} \times n_{\text{act}}}$ is the contact consistent task Jacobian, and $J_{\text{task}} \in \mathbb{R}^{n_{\text{task}} \times n_{\text{dofs}}}$ is the unconstrained task Jacobian. The basic control structure for the single-support phase of the biped is thus

$$\tau_{\text{control}} = J^{*T}_{\text{task}} F_{\text{task}} \tag{36}$$

with F_{task} being the entry point for feedback control laws to govern trajectories, applied forces, or combinations of the two in the operational space. For instance, when controlling an operational space position trajectory, we use the model-based control law

$$F_{\text{task}} = \Lambda^*_{\text{task}} u_{\text{task}} + \mu^*_{\text{task}} + p^*_{\text{task}} \tag{37}$$

with u_{task} being a desired acceleration for the operational reference. $\{\Lambda^*_{\text{task}}, \mu^*_{\text{task}}, p^*_{\text{task}}\}$ are inertia, velocity-based forces, and gravity-based forces in the operational space that can be found in the previous references.

In the case of dual support, there appear closed-loop effects between the legs of the robot in contact with the environment. Our previous work has thoroughly addressed this problem by creating structures to control the internal forces. Internal forces are defined as those that are orthogonal to joint accelerations. As such, internal forces do not produce any movement and only contribute to force generation within the closed loop formed by the contacts. Analyzing the right-hand side of (31), those forces correspond to the manifold

$$\left(U\, N_s\right)^T\, \tau_{\text{control}} = 0 \tag{38}$$

which reflect the cancellation of acceleration effects. Therefore, the torques that fulfill the above constraint belong to the null space of $(U\, N_s)$, which is defined by the projection

$$L^* \triangleq \left(I - U\, N_s\, \overline{U\, N_s}\right). \tag{39}$$

The torques associated with internal forces are those that do not contribute to net movement, i.e.,

$$\tau'_{\text{int}} = L^{*T} \tau_{\text{int}} \tag{40}$$

where τ_{int} is the set point for the internal forces. Thus, when simultaneously controlling operational space tasks and internal forces, we superimpose the orthogonal structures of (35) and (39) yielding the WBOSC command

$$\tau_{\text{control}} = J^{*T}_{\text{task}} F_{\text{task}} + L^{*T} \tau_{\text{int}}. \tag{41}$$

Internal forces can be physically defined as linear forces and mutually canceling reaction moments between pairs of supporting contacts. As explained in our previous works, such forces are expressed via the equation

$$F_{\text{int}} = W_{\text{int}}\, F_r \tag{42}$$

where F_r is the set of all reaction forces on the environment, and W_{int} is a matrix containing geometric transformations and selection criteria to extract the internal forces [see (6)]. Using this mapping, we demonstrated that the dynamics of internal forces correspond to

$$F_{\text{int}} = \left(\overline{J}^*_{i|l}\right)^T \tau_{\text{int}} + F_{\text{int},\{t\}} - \mu^*_i - p^*_i \tag{43}$$

where $\overline{J}^*_{i|l} \triangleq \left(L^* U \overline{J}_s W^T_{\text{int}}\right)$. Additionally, $F_{\text{int},\{t\}}$ are internal forces induced by task behavior, i.e., $F_{\text{int},\{t\}} \triangleq W_{\text{int}} \overline{J}^T_s\, J^{*T}_{\text{task}} F_{\text{task}}$, and μ^*_i and p^*_i are Coriolis/centrifugal and gravitational effects on the internal forces. Inverting the above equation, we derive the torques needed to accomplish a desired internal force

$$\tau_{\text{int}} = J^{*T}_{i|l} \left(F_{\text{int,ref}} - F_{\text{int},\{t\}} + \mu^*_i + p^*_i\right) \tag{44}$$

where $J^*_{i|l}$ is the Moore–Penrose left-pseudoinverse of $\overline{J}^*_{i|l}$, also referred to as the reduced Jacobian of internal forces acting on the contact closed loops, and $F_{\text{int,ref}}$ is the vector of desired internal forces that we use as an entry point to control internal forces.

Appendix B
Prismatic Inverted Pendulum Model

In the proposed planner, numerical integration starts with $\ddot{z}(x(n),\ , y(n))$, as described in Fig. 4. The PIP model can be expressed as the differential equation

$$\begin{aligned} \ddot{x} &= \frac{g + \ddot{z}}{z}(x - x_p) \\ \ddot{y} &= \frac{g + \ddot{z}}{z}(y - y_p). \end{aligned} \tag{45}$$

Accounting for z being a function of x, y, the height surface

$$\frac{dz}{dt} = \frac{\partial z}{\partial x}\dot{x} + \frac{\partial z}{\partial y}, \tag{46}$$

$$\frac{d^2 z}{dt^2} = \frac{d}{dt}\left(\frac{\partial z}{\partial x}\right)\dot{x} + \frac{\partial z}{\partial x}\ddot{x} + \frac{d}{dt}\left(\frac{\partial z}{\partial y}\right)\dot{y} \tag{47}$$

$$\ddot{z} = \frac{\partial^2 z}{\partial x^2}\dot{x}^2 + \frac{\partial z}{\partial x}\ddot{x} + \frac{\partial^2 z}{\partial y^2}\dot{y}^2 + \frac{\partial z}{\partial y}\ddot{y}. \tag{48}$$

By plugging (44) into (47), we obtain

$$\ddot{z} = \frac{\partial^2 z}{\partial x^2}\dot{x}^2 + \frac{\partial z}{\partial x}\frac{g + \ddot{z}}{z}(x - x_p) + \frac{\partial^2 z}{\partial y^2}\dot{y}^2 + \frac{\partial z}{\partial y}\frac{g + \ddot{z}}{z}(y - y_p) \tag{49}$$

$$\left(1 - \frac{g}{z}\left(\frac{\partial z}{\partial x}(x - x_p) + \frac{\partial z}{\partial y}(y - y_p)\right)\right)\ddot{z} \tag{50}$$

$$= \frac{\partial^2 z}{\partial x^2}\dot{x}^2 + \frac{\partial z}{\partial x}\frac{g}{z}(x - x_p) + \frac{\partial^2 z}{\partial y^2}\dot{y}^2 + \frac{\partial z}{\partial y}\frac{g}{z}(y - y_p)$$

$$\ddot{z} = \frac{\frac{\partial^2 z}{\partial x^2}\dot{x}^2 + \frac{\partial z}{\partial x}\frac{g}{z}(x - x_p) + \frac{\partial^2 z}{\partial y^2}\dot{y}^2 + \frac{\partial z}{\partial y}\frac{g}{z}(y - y_p)}{1 - \frac{g}{z}\left(\frac{\partial z}{\partial x}(x - x_p) + \frac{\partial z}{\partial y}(y - y_p)\right)}. \tag{51}$$

APPENDIX C
STATE ESTIMATION

The controller needs a body orientation estimate every servo cycle, 0.667 ms; yet, the motion capture system updates at only 480 Hz, occasionally fails to track a subset of the markers, and has a processing delay. An IMU is used to mitigate this problem. When the motion capture position update arrives, the new best estimate of the orientation at the instant in the past corresponding to the delayed sensor data is found. We maintain a list of recent IMU measurements and calculate a new estimate of the current robots orientation by integrating the angular velocities.

We use least squares to minimize the distance between motion capture LED positions $\hat{y}_i^k \in \mathbb{R}^3$ and predicted motion capture LED position $\tilde{y}_i^k \in \mathbb{R}^3$ for $i = 1, \ldots, n$, where n is typically 7, but decreases when LEDs are blocked, and where k represents the time in the past associated with the delayed motion capture data. Our model predicts motion capture LED locations based on an affine transformation of a default pattern $\tilde{y}_i^k = x^k + A^k z_i^k$, where $T^k = \{x^k \in \mathbb{R}^3, A^k \in \mathbb{R}^{3\times 3}\}$ is the affine transform at time k and the default pattern, $z_i \in \mathbb{R}^3$, represents the LEDs in a known frame. The pattern origin is the geometric center of the LED position, $\Sigma_{i=1}^7 e_j z_i = 0$ for $j = x, y, z$. Each affine transform includes both a linear translation and rotation term. Our estimation problem is linear in the individual components of this affine transform, which is why we use it. However, this linearity comes at a disadvantage: an affine transform can represent both physically realistic rotation and nonphysical skewing and scaling of the pattern. Since the physical reality will always bias the estimation problem toward valid rigid body transforms, we can safely assume that these transforms can be converted to a physical one later. We find the best fit affine transform as follows:

$$\theta^k = \begin{pmatrix} x^k \\ \mathrm{vec}(A^k) \end{pmatrix}, \qquad \tilde{\mathbf{y}}^k = \begin{pmatrix} \tilde{y}_1^k \\ \vdots \\ \tilde{y}_7^k \end{pmatrix} = R\theta^k \tag{52}$$

$$R = \begin{pmatrix} I_{3\times 3} & e_x z_1^\mathsf{T} & e_y z_1^\mathsf{T} & e_z z_1^\mathsf{T} \\ I_{3\times 3} & e_x z_2^\mathsf{T} & e_y z_2^\mathsf{T} & e_z z_2^\mathsf{T} \\ \vdots & \vdots & \vdots & \vdots \\ I_{3\times 3} & e_x z_7^\mathsf{T} & e_y z_7^\mathsf{T} & e_z z_7^\mathsf{T} \end{pmatrix} \tag{53}$$

$$K_o \in \mathbb{R}^{n\times 7} = \begin{pmatrix} e_0^\mathsf{T}, \text{ if LED 1 was found} \\ e_1^\mathsf{T}, \text{ if LED 2 was found} \\ \vdots \\ e_6^\mathsf{T}, \text{ if LED 7 was found} \end{pmatrix} \tag{54}$$

$$W = \begin{pmatrix} I_{3n\times 3n} & 0 \\ 0 & \lambda I_{9\times 9} \end{pmatrix} \tag{55}$$

$$R_r \triangleq \begin{pmatrix} (K_o \otimes I_{3\times 3})R \\ 0 \quad I_{9\times 9} \end{pmatrix} \tag{56}$$

$$\theta^k \triangleq (R_r^\mathsf{T} W R_r)^{-1} R_r^\mathsf{T} W \begin{pmatrix} (K_o \otimes I_{3\times 3})\hat{\mathbf{y}}^k \\ \tilde{\theta}(k|k-p) \end{pmatrix}. \tag{57}$$

Here, (52) describes affine transforms in vector form and demonstrates the linearity of prediction. The base regressor (53) incorporates the default pattern data, but assumes all the LEDs are visible. We use a knockout vector (54) of variable size to define which LEDs are to be used in each update. We also employ a regularization term and a weighting matrix (55) parameterized by a variable λ that controls the tradeoff between new rotational information and old data. This results in the full regularized regressor (56) and the best affine transform estimate (57).

Note that the regularization to $\tilde{\theta}(k|k-p)$ in (57) accounts for dynamics using the IMU data history, starting p steps before k at $k-p$—the index of the last LED position sensor update

$$\tilde{\theta}(k|k-p) = \hat{\theta}^{k-p} + \sum_{t=k-p}^{k} \begin{pmatrix} 0_3 \\ \mathrm{vec}(\hat{\omega}_{\mathrm{IMU}}^t \times) \end{pmatrix} \Delta t. \tag{58}$$

Finally, to attain a valid rigid body transform $\hat{\theta}^k$ from the general affine θ^k, we convert the direction cosine matrix A^k into the closest fit quaternion using the method of Bar-Itzhack [44] and then use that quaternion to generate a new direction cosine matrix $\hat{A}^k$, which is guaranteed to be a rigid body rotation. The original x^k and this new $\hat{A}^k$ form the rigid body transform $\hat{\theta}^k$. The algorithm returns $\tilde{\theta}(t, k)$ as the best estimate of the orientation until a new motion capture LED position message is received.

REFERENCES

[1] K. Iagnemma and J. Overholt, "Special issue: DARPA robotics challenge (DRC)," *J. Field Robot.*, vol. 32, no. 2, pp. 187–312, 2015.

[2] Y. Zhao, N. Paine, and L. Sentis, "Feedback parameter selection for impedance control of series elastic actuators," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 999–1006.

[3] S. Rezazadeh *et al.*, "Spring-mass walking with ATRIAS in 3D: Robust gait control spanning zero to 4.3 KPH on a heavily underactuated bipedal robot," in *Proc. ASME Dyn. Syst. Control Conf.*, Oct. 2015, p. V001T04A003.

[4] H. Park, A. Ramezani, and J. W. Grizzle, "A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 331–345, Mar. 2013.

[5] B. G. Buss, K. A. Hamed, B. A. Griffin, and J. W. Grizzle, "Experimental results for 3D bipedal robot walking based on systematic optimization of virtual constraints," in *Proc. Amer. Control Conf.*, Jul. 2016, pp. 4785–4792.

[6] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Quadrupedal locomotion using hierarchical operational space control," *Int. J. Robot. Res.*, vol. 33, no. 8, pp. 1047–1062, 2014.

[7] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "Ground reaction forces control for torque-controlled quadruped robots," in *Proc. IEEE Int. Conf. Intell. Robots Syst./Workshop Whole-Body Control Robots Real World*, 2014.

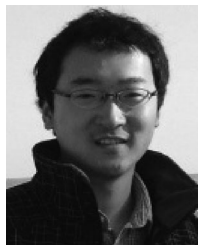
[8] M. Vukobratović and B. Borovac, "Zero-moment point: Thirty five years of its life," *Int. J. Humanoid Robots*, vol. 1, no. 1, pp. 157–173, 2004.

[9] J. Pratt *et al.*, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower body humanoid," *Int. J. Robot. Res.*, vol. 31, no. 10, pp. 1117–1133, 2012.

[10] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 355–368, Apr. 2015.
[11] D. Kim, G. Thomas, and L. Sentis, "Continuous cyclic stepping on 3D point-foot biped robots via constant time to velocity reversal," in *Proc. IEEE Int. Conf. Control Autom. Robot. Vision*, 2014, pp. 1637–1643.
[12] L. Sentis, "Synthesis and control of whole-body behaviors in humanoid systems," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2007.
[13] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 483–501, Jun. 2010.
[14] S. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: balancing in the presence of unknown external forces," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 884–898, Oct. 2007.
[15] N. Hogan, "Impedance control: An approach to manipulation," in *Proc. IEEE Amer. Control Conf.*, 1984, pp. 304–313.
[16] B. Stephens, "Push recovery control for force-controlled humanoid robots," Ph.D. dissertation, , Carnegie Mellon Univ., Pittsburgh, PA, USA, 2011.
[17] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *ArXiv e-prints*, Oct. 2014.
[18] S. Bertrand and J. Pratt, "Momentum-based control framework: Application to the humanoid robots atlas and valkyrie," in *Proc. IEEE Int. Conf. Intell. Robots Syst./Workshop Whole-Body Control Robots Real World*, 2014.
[19] L. Saab, O. Ramos, N. Mansard, P. Souères, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 346–362, Apr. 2013.
[20] B. Henze, C. Ott, and M. Roa, "Posture and balance control for humanoid robots in multi-contact scenarios based on model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 3253–3258.
[21] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1248–1255.
[22] S. Bertrand *et al.*, "Momentum-based control framework: Application to the humanoid robots atlas and valkyrie," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Workshop Slides*, 2014.
[23] H. Herr and M. Popovic, "Angular momentum in human walking," *J. Exp. Biol.*, vol. 211, no. 4, pp. 467–481, 2008.
[24] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonom. Robots*, vol. 35, nos. 2/3, pp. 161–176, 2013.
[25] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," *J. Physiol., Paris*, vol. 103, no. 3, pp. 211–219, 2009.
[26] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *Int. J. Robot. Res.*, vol. 20, no. 2, pp. 129–143, 2001.
[27] E. Westervelt, J. Grizzle, C. Chevallereau, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton, FL, USA: CRC Press, 2007.
[28] B. G. Buss *et al.*, "Preliminary walking experiments with underactuated 3d bipedal robot Marlo," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2529–2536.
[29] C. Chevallereau et al.,"RABBIT: A testbed for advanced control theory," *IEEE Control Syst. Mag.*, vol. 23, no. 5, pp. 57–79, Oct. 2003.
[30] T. Yang, E. Westervelt, A. Serrani, and J. P. Schmiedeler, "A framework for the control of stable aperiodic walking in underactuated planar bipeds," *Auton. Robots*, vol. 27, no. 3, pp. 277–290, 2009.
[31] A. Hereid *et al.*, "Dynamic multi-domain bipedal walking with ATRIAS through slip based human-inspired control," in *Proc. ACM Int. Conf. Hybrid Syst., Comput. Control*, 2014, pp. 263–272.
[32] M. Raibert, *Legged Robots that Balance*. Cambridge, MA, USA: MIT Press, 1986.
[33] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *Int. J. Robot. Res.*, vol. 30, no. 9, pp. 1170–1193, 2011.
[34] K. Yin, K. Loken, and M. van de Panne, "SIMBICON: Simple biped locomotion control," in *Proc. ACM Trans. Graph.*, vol. 26, no. 3, p. 105, 2007.
[35] S. Rezazadeh and J. W. Hurst, "Toward step-by-step synthesis of stable gaits for underactuated compliant legged robots," in *Proc. IEEE-RAS Int. Conf. Robot. Autom.*, 2015, pp. 4532–4538.
[36] H.-H. Zhao, W.-L. Ma, M. Zeagler, and A. Ames, "Human-inspired multi-contact locomotion with AMBER2," in *Proc. ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, Apr. 2014, pp. 199–210.
[37] J. Grizzle, A. Ramezani, B. Buss, B. Griffin, K. A. Hamed, and K. Galloway, "Progress on controlling MARLO, an ATRIAS-series 3D underactuated bipedal robot," *Dyn. Walking*, 2013.
[38] H. Vallery, R. Ekkelenkamp, H. Van Der Kooij, and M. Buss, "Passive and accurate torque control of series elastic actuators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 3534–3538.
[39] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Int. J. Robot. Res.*, vol. 3, no. 1, pp. 43–53, 1987.
[40] N. Paine *et al.*, "Actuator control for the NASA-JSC valkyrie humanoid Robot: A decoupled dynamics approach for torque control of series elastic robots," *J. Field Robot.*, vol. 32, no. 3, pp. 378–396, Jan. 2015.
[41] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet, "3D bipedal robotic walking: Models, feedback control, and open problems," in *Proc. IFAC Symp. Nonlinear Control Syst.*, vol. 2, no. 3, 2010, p. 8.
[42] Y. Zhao, N. Paine, K. S. Kim, and L. Sentis, "Stability and performance limits of latency-prone distributed feedback controllers," *IEEE Trans. Ind. Electron.*, vol. 62, no. 11, pp. 7151–7162, Nov. 2015.
[43] T. Boaventura, G. A. Medrano-Cerda, C. Semini, J. Buchli, and D. G. Caldwell, "Stability and performance of the compliance controller of the quadruped robot HyQ," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1458–1464.
[44] I. Y. Bar-Itzhack, "New method for extracting the quaternion from a rotation matrix," *J. Guid., Control, Dyn.*, vol. 23, no. 6, pp. 1085–1087, 2000.

Donghyun Kim received the B.S. degree in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2007. He received the M.S. degree in mechanical engineering from Seoul National University, Seoul, South Korea, in 2012. He is currently working toward the Ph.D. degree at the Human Centered Robotics Laboratory, University of Texas at Austin, Austin, TX, USA.

His research interests include agile locomotion, state estimation, and whole body control and optimal control of legged systems with real-time constraints. Mr. Kim received a National Science and Technology fellowship with an academic Distinguished Achievement Award from KAIST.



Ye Zhao (S'11–M'16) received the B.E. degree in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2011, and the M.S. and Ph.D. degrees in mechanical engineering from The University of Texas at Austin, Austin, TX, USA, in 2013 and 2016, respectively, where he also received the UT Robotics Portfolio graduate degree in 2016.

His research interests include robust motion planning and optimization, hybrid dynamics, temporal logic task planning of whole-body locomotion in unstructured environments, distributed impedance control of series elastic actuators, and robotic systems.

Dr. Zhao is a co-Chair of the IEEE Robotics and Automation Society (RAS) Student Activities Committee and a Committee Member of IEEE-RAS Member Activities Board. He is also a Review Editor of *Frontiers in Robotics* and *AI: Humanoid Robotics*, an Associate Editor of the 2017 IEEE Workshop on Advanced Robotics and its Social Impact and a Committee Member of the IEEE-RAS Technical Committee (TC) on Whole-Body Control and TC on Model-based Optimization for Robotics.

Gray Thomas received the B.S. degree from Olin College of Engineering, Needham, MA, USA, in 2012, where he designed his own major in robotics engineering. Since 2013, he has been working toward the Ph.D. degree in mechanical engineering from The University of Texas at Austin, Austin, TX, USA.

He joined Florida Institute for Human and Machine Cognition in 2012 and participated in the DARPA virtual robotics challenge. Since 2013 he has been with the Human Centered Robotics Laboratory, The University of Texas at Austin. Since 2015, he has been a NASA Space Technology Research Fellow, collaborating with Johnson Space Center. His research interests include robust system identification, reliable control for legged robots, and computationally efficient algorithms for robot control and planning.

Benito R. Fernandez received the M.S. and Ph.D. degrees in mechanical engineering from Massachusetts Institute of Technology, Cambridge, MA, USA, in 1985 and 1988, respectively, and he received degrees in Chemical and Materials Engineering from Simón Bolívar University, Sartenejas, Venezuela, in 1979 and 1981, respectively.

He joined the faculty of the UT Cockrell School of Engineering in 1990. He is part of the Advanced Mechatronics Laboratory and is Director of the Neuro-Engineering Research and Developing Laboratory. He has four issued patents and two pending patents, as well as numerous IP disclosures. He studies and develops engineering systems that have "brain-like" behavior. The mission of his laboratory's research is to advance the state-of-the-art in applied intelligence. His research interests include the areas of artificial neural systems, fuzzy logic systems, evolutionary systems, genetic algorithms, robust-optimal adaptive control, automated modeling, system identification, system diagnostics and prognosis, mobile agent systems, evolutionary design of mechatronic devices, hybrid eXtreme devices, evolutionary machines, adaptive robust-optimal control architecture, evolutionary neuro-fuzzy immune-based agent systems, remote robust diagnostics and prognosis of industrial equipment, heterogeneous unmanned mobile evolving robotic systems, and secure cyber-physical systems.

Luis Sentis (S'04–M'07) received the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, where he developed leading work in theoretical and computational methods for the compliant control of humanoid robots.

He was in Silicon Valley in the area of clean room automation. He is currently an Associate Professor in aerospace engineering with The University of Texas at Austin, Austin, TX, USA, where he directs the Human Centered Robotics Laboratory, an experimental laboratory focusing on control and embodiment of humanoid robots. He was the UT Austin's Lead for DARPA's Robotics Challenge with NASA Johnson Space Center from 2012 to 2013, where he helped to design and test the Valkyrie humanoid robot. His research interests include real-time optimal control of human centered robots, motion planning for dynamic locomotion, designing high performance series elastic actuators, embedded control systems, devising motion planning methods for contact safety in human robot interactions, and design of software systems for mobile manipulation.

Dr. Sentis received the NASA Elite Team Award for his contributions to NASA's Johnson Space Center Software Robotics and Simulation Division.