MIT Libraries | DSpace@MIT

MIT Open Access Articles

MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot

The MIT Faculty has made this article openly available. ***Please share*** how this access benefits you. Your story matters.

Citation: Bledt, Gerado et al. "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot." IEEE International Conference on Intelligent Robots and Systems, October 2018, Madrid Spain, Institute of Electrical and Electronics Engineers, January 2019. © 2018 IEEE

As Published: http://dx.doi.org/10.1109/iros.2018.8593885

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: https://hdl.handle.net/1721.1/126619

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike

Massachusetts Institute of Technology

MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot

Gerardo Bledt[1,2], Matthew J. Powell[1], Benjamin Katz[1],
Jared Di Carlo[2], Patrick M. Wensing[3], and Sangbae Kim[1]

***Abstract*— This paper introduces a new robust, dynamic quadruped, the MIT Cheetah 3. Like its predecessor, the Cheetah 3 exploits tailored mechanical design to enable simple control strategies for dynamic locomotion and features high-bandwidth proprioceptive actuators to manage physical interaction with the environment. A new leg design is presented that includes proprioceptive actuation on the abduction/adduction degrees of freedom in addition to an expanded range of motion on the hips and knees. To make full use of these new capabilities, general balance and locomotion controllers for Cheetah 3 are presented. These controllers are embedded into a modular control architecture that allows the robot to handle unexpected terrain disturbances through reactive gait modification and without the need for external sensors or prior environment knowledge. The efficiency of the robot is demonstrated by a low Cost of Transport (CoT) over multiple gaits at moderate speeds, with the lowest CoT of 0.45 found during trotting. Experiments showcase the ability to blindly climb up stairs as a result of the full system integration. These results collectively represent a promising step toward a platform capable of generalized dynamic legged locomotion.**

Fig. 1: **MIT Cheetah 3.** The MIT Cheetah 3 quadruped robot platform is an electrically powered robust robot capable of untethered 3D locomotion in uncertain terrains.

I. Introduction

Legged robots have a disruptive potential for application in challenging man-made and natural terrains. In contrast to wheeled or tracked vehicles, legs provide a high degree of versatility in how they are deployed, offering marked advantages for operation in irregular environments. Designing and controlling machines to realize these potentials has long motivated work across the legged robotics community, with increasing intensity since the onset of the recent DARPA Robotics Challenge (DRC) [1]. In the aftermath of this event, highly-capable quadrupeds (e.g., [2], [3]) and humanoids (e.g., [4], [5]) in research labs have begun to lend credibility to vision of legged robots making their way into our human environments to operate by our side or in our stead.

Yet, despite the broad range of capabilities displayed in modern machines, many demonstrations remain instances of success without the broad versatility or reliability necessary for real-world deployments. It is a conundrum that while legged machines may have the capacity to outperform many humans and animals in cognitive tasks of reasoning and intelligence, they still lag behind even basic animals in terms of the physical skills of mobility that makes them attractive in the first place. Many animals naturally learn to walk and even run within hours of being born, yet the ability to do both remains a challenge for legged robots.

Quadruped robots have recently shown impressive advancements in dynamic locomotion capabilities using various actuation methods and control strategies. The use of hydraulic actuators has proven to be successful in legged locomotion with Big Dog by Boston Dynamics [6] and IIT's HyQ quadruped [7]. These robots take advantage of the hydraulic actuation system's ability to output large forces at the joints. ANYmal at ETH [2] features an extensive range of motion for completing autonomous tasks in real-world situations with the use of Series Elastic Actuators (SEAs). SEAs characteristically have good impact mitigation properties that are crucial for high speed locomotion, as well as excellent force control capacity. The MIT Cheetah 2 made use of a custom proprioceptive actuator design that possesses high impact mitigation, force control, and position control capabilities. This design enabled it to autonomously jump over obstacles [3] and bound at high speeds of 6m/s [8], but had a limited range of motion constraining it to sagittal plane locomotion.

Expanding on the results from the MIT Cheetah 2, we present its successor, the MIT Cheetah 3, pictured in Figure 1. This new platform employs similar philosophies as its predecessor, such as the use of proprioceptive electric motors

[1]Department of Mechanical Engineering, MIT, Cambridge, MA 02139, USA: gbledt@mit.edu,mjpowell@mit.edu, benkatz@mit.edu, sangbae@mit.edu

[2]Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA: gbledt@mit.edu

[3]Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA: pwensing@nd.edu

[1,2,3]Authors are associated with the MIT Biomimetic Robotics Lab and contributed equally to the success of the robot

This work was supported by National Science Foundation [NSF-IIS-1350879] and the Air Force Office of Scientific Research [AFOSR Grant FA2386-17-1-4661]

TABLE I: Physical Robot Parameters

Parameter	Symbol	Value	Units
Mass	m	45	kg
Body Inertia	I_{xx}	0.35	$\mathrm{kg \cdot m^2}$
	I_{yy}	2.1	$\mathrm{kg \cdot m^2}$
	I_{zz}	2.1	$\mathrm{kg \cdot m^2}$
Body Length	l_{body}	0.600	m
Body Width	w_{body}	0.256	m
Body Height	h_{body}	0.200	m
Leg Link Lengths	l_1, l_2	0.34	m

and simple control models for planning movement. The new robot possesses many critical improvements, including an expanded range of motion, higher force production capabilities, and an ability to control these forces in full 3D generality. A new general control architecture is presented to make full use of these capabilities. Covering both design and control, this paper provides a system level overview of the important components that result in the MIT Cheetah 3's robust, dynamic locomotion capabilities. Section II presents the main hardware platform design considerations. The control software architecture and components are described in Section III. Results from selected hardware experiments are shown in Section IV. Finally, Section V discusses implications of the robotic platform and future research directions that it will allow the team to explore.

II. Design

A. Overview

The hardware design of Cheetah 3 builds on the actuation paradigm of the MIT Cheetah 1 and 2 robots [9]. By using high torque density electric motors with backdriveable single-stage planetary gear reductions, and low-inertia legs, the Cheetah 3 robot can control ground reaction forces through proprioception, without the use of any force sensors, torque sensors, or series compliance at the joints or feet. The Cheetah 2 robot was designed primarily for fast locomotion in the sagittal plane, and used high performance actuators at the hip and knee joints, but not for abduction/adduction (ab/ad). Cheetah 3 has nearly identical actuators on all three degrees of freedom on each leg, enabling fully 3D control of ground reaction forces.

The Cheetah 3 legs feature a greatly expanded range of motion compared to Cheetah 2. The new ab/ad actuators have a range of motion of more than $\pm 45^\circ$, while new hip and knee designs allow the robot operate identically forwards, backwards, and flipped upside-down, and potentially use its legs for simple manipulation tasks as well as locomotion.

The robot is powered by 450 Watt-hours of on-board Lithium Polymer batteries providing approximately 2 hours of run time, although the robot has internal space for twice the battery capacity, should longer run times be necessary.

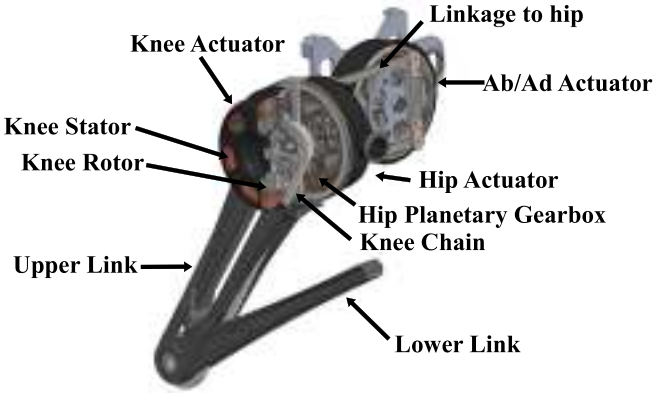

Fig. 2: **Cheetah 3 Leg Design.** One leg showing the 3 actuators, with a cutaway view of the knee actuator.

B. Leg Design and Actuation

The proprioceptive actuation strategy used on Cheetah 3 results in exceptional force bandwidth, force capability, and speed, at the cost of force accuracy compared to systems with closed-loop force control. The static torque error of Cheetah 3's actuators is roughly $\pm 10\%$ primarily due to gear friction. By comparison, the series elastic actuators used in ANYmal, which measure the deflection of a series spring to control output torque, have far greater torque accuracy, but much lower peak torque, speed, and torque bandwidth, especially at high torques [2]. However, we believe high force accuracy is not a requirement for dynamic legged locomotion, and our experiments have supported this claim.

Each of Cheetah 3's actuators consists of a custom high torque density electric motor coupled to a single-stage 7.67:1 planetary gear reduction. The slightly higher gear ratio compared to Cheetah 2 (5.8:1) was chosen to improve the load-carrying ability and low-speed efficiency of the robot. The legs are serially actuated, but to keep leg inertia low, the hip and knee actuators are co-axially located at the hip of each leg as shown in Figure 2.

The lower link is driven by a roller chain which passes through the upper link, providing an additional 1.15x gear reduction. By using a chain rather than a linkage, the lower link can rotate 330°, letting the robot arbitrarily change the orientation of its knees to point forwards or backwards. While the roller chain does introduce slight torque and speed ripple at the link, and adds some compliance between the knee actuator and the lower link, in practice these effects have not caused high-level performance to suffer.

The hip joint can rotate continuously, limited only by the length of the wires to the knee actuator, allowing the robot to potentially operate upside-down, climb up tall obstacles, or use its feet for manipulation above its body. Two ab/ad actuators are located between each pair of legs, and are coupled to the legs by linkages. See Figure 2 for a cutaway view of leg actuation. The leg links are machined aluminum, and in total have a mass of 2.7 kg for all four legs, or only 6% of the mass of the robot. The end of each leg has a cover made from 10mm thick 60A urethane rubber,

TABLE II: Actuator Parameters

Parameter	Value	Units
Gear Ratio	7.67	
Max Torque	230	Nm
Max Joint Speed	21	Rad/s

providing cushioning and traction regardless of the contact angle between the leg and the ground. Cheetah 3's actuation capabilities are summarized in Table II.

Cheetah 3's actuation system was designed to give the robot performance headroom for tasks including high speed locomotion, jumping, carrying loads, and recovering from extreme disturbances. As a simple performance metric, with the leg minimally extended, the robot is capable of producing a purely vertical ground reaction force of over 700 N, about 1.6 times the weight of the robot, *per leg*. And at 70% extension, a typical configuration during operation, vertical force capability exceeds 1000 N per leg. By contrast, other quadruped robots for which data are available have far lower force-to-bodyweight capabilities: At minimal leg extension, ANYmal can produce around 0.54 bodyweights per leg [2], and HyQ2Max 0.84 bodyweights [7].

C. Computing and Low-Level Control Architecture

Cheetah 3 has a tiered computing architecture which enables low-level leg and motor control to be run at higher loop frequencies than locomotion control, and allows easy expansion of computing resources as needed for future sensing, planning, and navitgation tasks. Locomotion control and state estimation are handled by the locomotion computer: an embedded PC with a 2nd Gen Core i7 laptop CPU, running Ubuntu Linux (kernel 4.1.33) with the CONFIG_PREEMPT_RT patch. The locomotion computer receives user commands and logs data using Lightweight Communication and Marshalling (LCM) [10]. LCM will allow additional computers for vision, planning, and other tasks to easily communicate with the locomotion computer in the future. Leg-level control is performed by an ARM Cortex-A8-based processor. The leg controller communicates with the locomotion computer at 1 kHz over EtherCAT, and performs leg-level control tasks like cartesian impedance control or joint PD control at 4.5 kHz. The higher sample frequency at the leg level allows for high-rate joint velocity filtering as well as higher bandwidth tracking during swing phase and other position-controlled actions. Finally, each leg controller sends torque commands and receives joint encoder measurements from the brushless motor drives for the three motors in each leg, which perform the current control on the motors at 20 kHz sample rate with 400 Hz closed-loop bandwidth.

III. Control Architecture

The overall architecture of the system is depicted in the block diagram in Figure 3. Each block is designed to be modular so it can be easily substituted without any modification to the other parts of the system. The operator provides high-level commands by giving a desired translational velocity, $\dot{\boldsymbol{p}}_d$, and turning rate, $\dot{\psi}_d$. These commands are used to plan a smooth and controllable CoM reference trajectory that is relayed to the body and leg controllers. Various controllers and planners use the user-input commands and the estimated robot state to generate force commands if the leg is in stance or position commands if the leg is in swing. The following sections describe the implementation of the major components of the system on the MIT Cheetah 3.

A. Gait Planning

The Cheetah's gait is defined by an event-based finite state machine that uses a leg-independent phase variable to schedule nominal contact and swing phases. This allows for flexible gait definitions and fluid transitions between them. Trotting, bounding, and pacing gaits are presented in this paper, but the framework allows implementation of any gait definition with ease. The gaits were designed to mimic natural animal gaits by controlling the independent phases of each leg. This nominal gait plan is modified during unexpected contact events on the legs. Since the robot does not use any external environment sensors, a contact detection algorithm probabilistically fuses encoder measurements, estimated force, and expected gait phase to estimate the likelihood that each leg is in contact with an object [11]. Scheduled contacts are defined by independent boolean variables $s_\phi \in \{0 = swing, 1 = contact\}$, while estimated contacts are given by $\hat{s} \in \{0 = swing, 1 = contact\}$. With this information, the robot can differentiate between normal operation, unexpected early contacts, and late missed contacts to adjust its control actions appropriately [11].

B. Controller Model

Due to the irregularities of the hybrid nature arising from the repeated contact mode switches as legs enter or leave stance or swing, it is often difficult to use traditional control methods for balancing the robots. Rapidly transitioning between periods of underactuation make the problem even more complex. Therefore, the controllers used on the robot make use of a simplified control model templates to optimize ground reaction forces at the footstep locations.

By design, the robot has light limbs with low inertia as compared to the overall body. For this reason, we can reasonably simplify the control model to ignore the effects of the legs for planning ground reaction forces from the stance feet. In particular, the Cheetah 3 controller model employs a commonly used linear relationship [12], [13] between the robot's COM translational $\ddot{\boldsymbol{p}}_c$ and body angular acceleration $\dot{\boldsymbol{\omega}}_b$ and the forces $\boldsymbol{F} = (\boldsymbol{F}_1^T, \boldsymbol{F}_2^T, \boldsymbol{F}_3^T, \boldsymbol{F}_4^T)^T$ acting on each of the robot's four feet. The controller model is given by

$$\underbrace{\begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ [\boldsymbol{p}_1 - \boldsymbol{p}_c]\times & \dots & [\boldsymbol{p}_4 - \boldsymbol{p}_c]\times \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{F} = \underbrace{\begin{bmatrix} m(\ddot{\boldsymbol{p}}_c + \boldsymbol{g}) \\ \boldsymbol{I}_G \dot{\boldsymbol{\omega}}_b \end{bmatrix}}_{\boldsymbol{b}}, \tag{1}$$

where m and $\boldsymbol{I}_G$ are the robot's total mass and centroidal rotational inertia, $\boldsymbol{g}$ is the gravity vector and $\boldsymbol{p}_i, i \in \{1, 2, 3, 4\}$

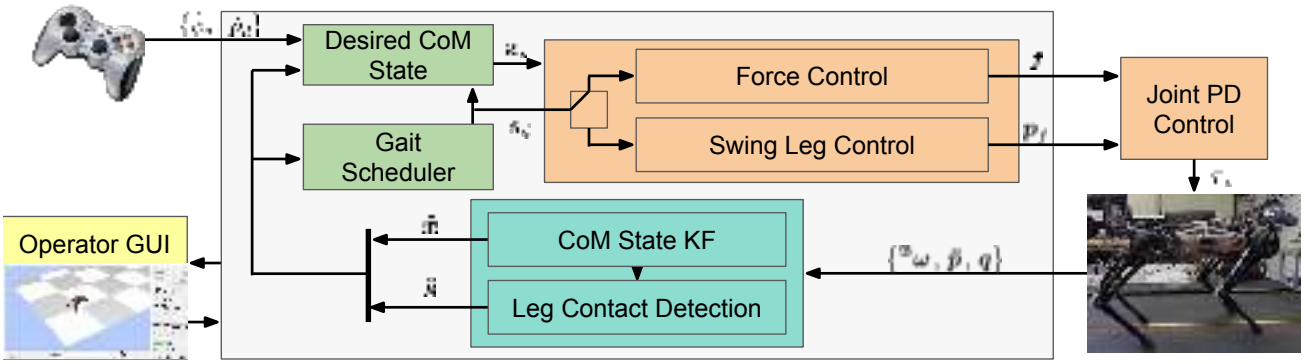

Fig. 3: **System Architecture Block Diagram.** The user sends velocity and turning commands as well as general tunable parameters to the main computer. The main Cheetah control computer is composed of three main parts: higher-level planning (green), leg and body control (red), and state estimation (blue). The force and position commands are sent to the microcontrollers for each leg that relay the motor command to the robot.

are the positions of the feet. The term $[\boldsymbol{p}_i - \boldsymbol{p}_c]\times$ is the skew-symmetric matrix representing the cross product $(\boldsymbol{p}_i - \boldsymbol{p}_c) \times \boldsymbol{F}_i$.

C. Force Control - Balance Controller

One of the Cheetah 3 support leg control modes is a Balance Controller, an implementation (with slight modification) of the controller described in [12]. The Balance Controller enforces PD control on the center of mass and body orientation, while also ensuring that foot forces satisfy friction constraints. The PD control law is given by

$$\begin{bmatrix} \ddot{\boldsymbol{p}}_{c,d} \\ \dot{\boldsymbol{\omega}}_{b,d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{p,p}(\boldsymbol{p}_{c,d} - \boldsymbol{p}_c) + \boldsymbol{K}_{d,p}(\dot{\boldsymbol{p}}_{c,d} - \dot{\boldsymbol{p}}_c) \\ \boldsymbol{K}_{p,\omega}\log(\boldsymbol{R}_d\boldsymbol{R}^T) + \boldsymbol{K}_{d,\omega}(\boldsymbol{\omega}_{b,d} - \boldsymbol{\omega}) \end{bmatrix} \quad (2)$$

The desired angular acceleration reflects PD control on $SO(3)$ wherein the desired and actual body orientations are described using rotation matrices $\boldsymbol{R}_d$ and $\boldsymbol{R}$, respectively, and the orientation error is obtained using the exponential map representation of rotations [14], [15].

The goal of the Balance Controller is to resolve an optimal distribution of leg forces $\boldsymbol{F}$ that drive the approximate COM dynamics to the corresponding desired dynamics given by

$$\boldsymbol{b}_d = \begin{bmatrix} m(\ddot{\boldsymbol{p}}_{c,d} + \boldsymbol{g}) \\ \boldsymbol{I}_G\,\dot{\boldsymbol{\omega}}_{b,d} \end{bmatrix}. \quad (3)$$

Since the model (1) is linear, the controller can naturally be expressed as the solution of a quadratic program (QP) [16]

$$\begin{aligned} \boldsymbol{F}^* = \min_{\boldsymbol{F}\in\mathbb{R}^{12}} \; & (\boldsymbol{A}\boldsymbol{F} - \boldsymbol{b}_d)^T\boldsymbol{S}(\boldsymbol{A}\boldsymbol{F} - \boldsymbol{b}_d) \\ & + \alpha\|\boldsymbol{F}\|^2 + \beta\|\boldsymbol{F} - \boldsymbol{F}^*_{\text{prev}}\|^2 \quad (4) \\ \text{s.t.} \quad & \boldsymbol{C}\boldsymbol{F} \le \boldsymbol{d} \end{aligned}$$

The cost function in (4) reflects a tradeoff between three goals: driving the COM dynamics to the desired values, minimizing the forces used, and penalizing deviations between the current QP solution and the solution at the previous timestep, $\boldsymbol{F}^*_{prev}$. The matrix $\boldsymbol{S}$ determines the relative priority in control over the rotational and translational motion, and the gains $\alpha > 0$ and $\beta > 0$ dictate the influence of the force normalization and solution filtering. Constraints $\boldsymbol{C}\boldsymbol{F} \le \boldsymbol{d}$ are enforced to ensure that the optimized forces lie in the friction pyramid and that the normal forces lie within feasible bounds; these switch between support-leg and swing-leg bounds according to the scheduled contact, s_ϕ, as described in Section III-A.

D. Force Control - Model Predictive Control

As an alternative to the Balance Controller, the ground force control block can be replaced with a model-predictive controller (MPC) that reasons about trajectory outcomes over a longer temporal horizon. A model-predictive force-planning approach can be formulated to plan ground reaction forces $\boldsymbol{u}$ to minimize a cost function

$$J = \sum_{i=0}^{k}(\boldsymbol{x}_i - \boldsymbol{x}_i^{\text{ref}})^\top\boldsymbol{Q}_i(\boldsymbol{x}_i - \boldsymbol{x}_i^{\text{ref}}) + \boldsymbol{u}_i^\top\boldsymbol{R}_i\boldsymbol{u}_i \quad (5)$$

where $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ are positive semidefinite matrices of weights, $\boldsymbol{x}_i \in \mathbb{R}^{12}$ is the robot's state at timestep i, consisting of position, velocity, orientation, and angular velocity, $\boldsymbol{x}_i^{\text{ref}}$ is the desired state of the robot, and k is the horizon length. Solving this MPC is challenging, as orientation dynamics have nonlinear coupled effects. In recent work [17] approximate dynamics are enforced with linear equality constraints based on a time-varying approximation that enables formulating the MPC problem as a quadratic program. The resulting controller is able to anticipate and plan around periods of flight and underactuation. Optimal forces are updated at 30 Hz and are sent directly to the leg controllers [17].

E. Swing Leg Control

Each of the footstep locations are calculated from the corresponding hip location using a linear combination of the Raibert heuristic [18] and a velocity-based feedback term from the capture point formulation [19]. Since the robot does not have external environment sensors, the footstep locations are projected onto an assumed ground plane. Therefore, the step location on the 2D ground plane for foot i is calculated from the hip of the robot by the following

$$\boldsymbol{p}_{step,i} = \boldsymbol{p}_{h,i} + \underbrace{\frac{T_{c_\phi}}{2}\dot{\boldsymbol{p}}_{c,d}}_{\text{Raibert Heuristic}} + \underbrace{\sqrt{\frac{z_0}{\|\boldsymbol{g}\|}}(\dot{\boldsymbol{p}}_c - \dot{\boldsymbol{p}}_{c,d})}_{\text{Capture Point}} \quad (6)$$

where T_{c_ϕ} is the nominal scheduled contact phase (stance) time, z_0 is the nominal height of locomotion, and $\boldsymbol{p}_{h,i}$ provides the position of the corresponding hip i.

A PD controller with a feedforward term is used to compute joint torques for tracking the Cartesian swing trajectory for each foot. The feedforward force is computed on the onboard embedded PC at 1 kHz using the leg dynamics

$$\boldsymbol{\tau}_{\text{ff},i} = \boldsymbol{J}_i^\top \boldsymbol{\Lambda}_i({}^{\mathfrak{B}}\boldsymbol{a}_{i,\text{ref}} - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}}_i) + \boldsymbol{C}_i\dot{\boldsymbol{q}}_i + \boldsymbol{G}_i \tag{7}$$

where $\boldsymbol{J}_i$ is the foot Jacobian, $\boldsymbol{\Lambda}_i$ is the operational space inertia matrix, ${}^{\mathfrak{B}}\boldsymbol{a}_{i,\text{ref}}$ is the reference acceleration for the swing trajectory, $\boldsymbol{q}_i$ is a vector of joint configurations, $\boldsymbol{C}_i$ is the Coriolis matrix, and $\boldsymbol{G}_i$ is the torque due to gravity. The full controller for tracking swing trajectories is

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^\top[\boldsymbol{K}_p({}^{\mathfrak{B}}\boldsymbol{p}_{i,\text{ref}} - {}^{\mathfrak{B}}\boldsymbol{p}_i) + \boldsymbol{K}_d({}^{\mathfrak{B}}\boldsymbol{v}_{i,\text{ref}} - {}^{\mathfrak{B}}\boldsymbol{v}_i)] + \boldsymbol{\tau}_{\text{ff},i} \tag{8}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are diagonal matrices of proportional and derivative gains. The Cartesian PD controller runs on the leg controllers at 4.5 kHz.

To ensure the swing-leg PD controller remains stable, the gains are scaled with the apparent mass of the leg to maintain approximately the same natural frequency

$$K_{p,j} = \omega_{\text{des}}^2 \Lambda_{jj}(\boldsymbol{q}) \tag{9}$$

where $K_{p,j}$ is the j-th diagonal entry in $\boldsymbol{K}_p$, ω_{des} is the desired natural frequency, and Λ_{jj} is the j-th diagonal entry in the operational space inertia matrix.

F. Virtual Predictive Support Polygon

In order to take advantage of the hybrid nature of legged locomotion, a virtual support polygon is defined to provide a desired CoM location that generalizes across all gaits. By anticipating contact mode switches, the virtual support polygon biases away from legs nearing the end of their contact phase and towards legs about to touchdown. This strategy enables the robot to maintain its forward momentum during the gait while using selected footstep locations to create a smooth reference trajectory that is automatically adapted to the footholds online.

We define a phase-based approach that assigns each foot's contribution to the predictive support polygon, including feet in swing. The goal is to design an informed nonlinear phase-dependent weighting strategy by exploiting the robot's knowledge of which feet will be the next to lift off or touch down and when these state changes are scheduled to occur. Weighting factors are generated based on the contact and swing phases

$$K_{c_\phi} = \frac{1}{2}\left[\text{erf}\left(\frac{\phi}{\sigma_{c_0}\sqrt{2}}\right) + \text{erf}\left(\frac{1-\phi}{\sigma_{c_1}\sqrt{2}}\right)\right] \tag{10}$$

$$K_{\bar{c}_\phi} = \frac{1}{2}\left[2 + \text{erf}\left(\frac{-\phi}{\sigma_{\bar{c}_0}\sqrt{2}}\right) + \text{erf}\left(\frac{\phi-1}{\sigma_{\bar{c}_1}\sqrt{2}}\right)\right] \tag{11}$$

$$\Phi = s_\phi K_{c_\phi} + \bar{s}_\phi K_{\bar{c}_\phi} \tag{12}$$

where K_{c_ϕ} and $K_{\bar{c}_\phi}$ correspond to the adaptive weighting factor during the scheduled contact and swing phase respectively. The quantity Φ represents the total weighting factor for the foot. Roughly, this notifies the robot that the closer the leg is to the middle of its contact phase, the more confidently the robot can rely on that the leg as a support point, while the closer it is to the middle of the swing phase, the less the leg can be used for balance in the near future.

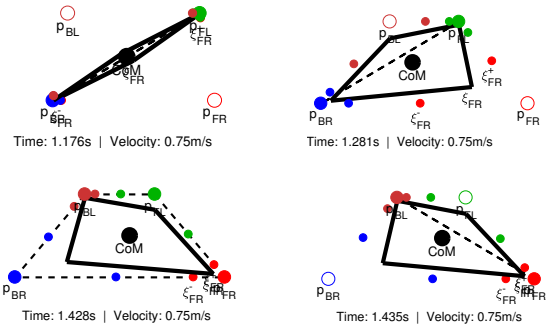

Fig. 4: **Predictive Support Polygon.** As the robot trots with velocity to the right, the predictive support polygon (solid black) anticipates leg touchdown and guides the CoM away from the instantaneous physical support line (dashed black) towards the future support as contact modes switch. The vertices of the predictive support polygon are given by $\boldsymbol{\xi}_i$.

Using the phase-based weighting factors, a set of virtual points is defined for each leg. These virtual points lend a method to continuously weight the predicted relative availability of each leg as compared to its adjacent legs. These virtual points slide between the 2D ground projection line between the foot and the corresponding adjacent foot during the gait cycle. Each leg, $i \in \{FR, FL, BR, BL\}$, has two virtual points given by

$$\begin{bmatrix}\boldsymbol{\xi}_i^- \\ \boldsymbol{\xi}_i^+\end{bmatrix} = \begin{bmatrix}\boldsymbol{p}_i & \boldsymbol{p}_{i^-} \\ \boldsymbol{p}_i & \boldsymbol{p}_{i^+}\end{bmatrix}\begin{bmatrix}\Phi_i \\ 1-\Phi_i\end{bmatrix} \tag{13}$$

where the superscript signifies a reference to the adjacent leg clockwise, $-$, or counterclockwise, $+$. With the current foot position, we can now describe the predictive support polygon vertex for leg i to be

$$\boldsymbol{\xi}_i = \frac{\Phi_i\boldsymbol{p}_i + \Phi_{i^-}\boldsymbol{\xi}_i^- + \Phi_{i^+}\boldsymbol{\xi}_i^+}{\Phi_i + \Phi_{i^-} + \Phi_{i^+}} \tag{14}$$

The desired CoM position is then given as the average of these weighted virtual support polygon vertices

$$\hat{\boldsymbol{p}}_{CoM,d} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\xi}_i \tag{15}$$

Figure 4 shows an example evolution of this virtual predictive support polygon during trotting, but the method extends to all gaits. The virtual points defined in (13) and (14) are labeled for the front right foot. In the sequence, the CoM moves from the support line in the middle of stance, towards the future support polygon as it expects the swing feet to touchdown and provide support.

G. Posture Adjustment on Sloped Terrain

As mentioned, part of the Cheetah 3 controller design philosophy is to create a robust algorithm without reliance on contact sensors and vision systems. To enable Cheetah

to traverse stairs and sloped terrain without vision, we use measurements of each footstep location $\boldsymbol{p}_i = (p_i^x, p_i^y, p_i^z)$ to approximate the local slope of the walking surface and adjust the robot's desired posture. In particular, the walking surface is modeled as a plane

$$z(x, y) = a_0 + a_1 x + a_2 y. \tag{16}$$

Coefficients $\boldsymbol{a} = (a_0, a_1, a_2)^T$ of (16) are obtained through the solution of the least squares problem

$$\boldsymbol{a} = (\boldsymbol{W}^T \boldsymbol{W})^\dagger \boldsymbol{W}^T \boldsymbol{p}^z \tag{17}$$

$$\boldsymbol{W} = \begin{bmatrix} \mathbf{1} & \boldsymbol{p}^x & \boldsymbol{p}^y \end{bmatrix}_{4\times 3} \tag{18}$$

that finds the plane that best fits the collection of measurements of the most recent contact point for each leg. Note that $\boldsymbol{p}^x$, $\boldsymbol{p}^y$ and $\boldsymbol{p}^z$ contain data for each leg, e.g. $\boldsymbol{p}^x = (p_1^x, p_2^x, p_3^x, p_4^x)$, and that $(\boldsymbol{W}^T \boldsymbol{W})^\dagger$ is the Moore-Penrose pseudoinverse of $\boldsymbol{W}^T \boldsymbol{W}$. As described in Section IV, the estimated walking surface is used to adjust the robot's posture to accommodate uneven terrain.

H. State Estimation

Cheetah 3 estimates is it body states through a two-stage sensor fusion algorithm that decouples estimation of body orientation from estimation of the body position and velocity. It first uses its IMU to estimate orientation. Following this stage, it then fuses accelerometer measurements with leg kinematics to estimate base position and velocity via a procedure inspired by [20]. A similar two-stage scheme has also been recently applied for state-estimation in the HRP2 humanoid, as reported in [21].

The first stage of the state estimation employs an orientation filter [22] using both the IMU gyro and accelerometer readings. The main idea of the filter is that the gyro provides an accurate reading of the high-frequency orientation dynamics, whereas the presence of a gravity bias on the accelerometer allows it to de-drift the estimate at a comparatively lower frequency. Letting the orientation estimate ${}^0\hat{\boldsymbol{R}}_b$ as the orientation of the body relative to the i.c.s., the filter updates this estimate according to

$${}^0\dot{\hat{\boldsymbol{R}}}_b = {}^0\hat{\boldsymbol{R}}_b \left[{}^b\boldsymbol{\omega}_b + \kappa \boldsymbol{\omega}_{corr} \right] \times \tag{19}$$

where $\kappa > 0$ is a correction gain and $\boldsymbol{\omega}_{corr}$ is a correction angular velocity to align the accelerometer reading $\boldsymbol{a}_b$ with its gravity bias

$$\boldsymbol{\omega}_{corr} = \frac{\boldsymbol{a}_b}{\|\boldsymbol{a}_b\|} \times {}^0\hat{\boldsymbol{R}}_b^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The time constant of the de-drifting from this term can be approximated by κ^{-1}. In practice κ is heuristically decreased during highly-dynamic portions of the gait where $\|\boldsymbol{a}_b\| >> g$ with

$$\kappa = \kappa_{\text{ref}} \max(\min(1, 1 - ||\boldsymbol{a}_b - g||/g), 0) \tag{20}$$

where g is the acceleration of gravity and κ_{ref} is chosen as $\kappa_{\text{ref}} = 0.1$. This process is effective to de-drift pitch and roll, however, error accumulation on yaw is unavoidable without fusion using exteroceptive information such as vision [23].

The second stage of the state estimation uses the orientation estimate ${}^0\hat{\boldsymbol{R}}_b$ along with kinematic measurements from the legs to estimate the base position and velocity. In contrast to previous state estimation techniques that posed this problem as an extended Kalman Filter [20], the two-stage approach allows this secondary fusion to be posed as a conventional Kalman filter. This simplifies analysis and tuning of the filter and guarantees that the filter equations will never diverge in finite time.

In continuous time, the process equations are modeled as

$${}^0\dot{\boldsymbol{p}}_b = {}^0\boldsymbol{v}_b \tag{21}$$

$${}^0\dot{\boldsymbol{v}}_b = {}^0\hat{\boldsymbol{R}}_b \boldsymbol{a}_b + {}^0\boldsymbol{a}_g + \boldsymbol{w}_{\boldsymbol{v}} \tag{22}$$

$${}^0\dot{\boldsymbol{p}}_i = \boldsymbol{w}_{\boldsymbol{p}_i} \quad \forall i = \{1, \ldots, 4\} \tag{23}$$

where ${}^0\boldsymbol{p}_b$ is the position of the body, ${}^0\boldsymbol{v}_b$ its velocity, ${}^0\boldsymbol{a}_g = [0,\ 0,\ -g]^T$ the gravitational acceleration, and ${}^0\boldsymbol{p}_i$ the position of foot i. The white noise terms $\boldsymbol{w}_{\boldsymbol{v}}$ and $\boldsymbol{w}_{\boldsymbol{p}_i}$ characterize the noise in the accelerometer and any variability in the foot positions. As in [20] the variability in the foot position is modeled to account for effects of foot slip in stance, or can also capture foot swing by inflating the process noise on that foot to a high level. Viewing ${}^0\hat{\boldsymbol{R}}_b \boldsymbol{a}_b + {}^0\boldsymbol{a}_g$ as an input to the system, these equations represent a linear time invariant process, and thus are amenable to conventional Kalman filtering. The continuous time model is converted into discrete time with a 1ms time step using an identical procedure to [20].

Leg kinematics provide measurements of the relative position vector between each foot and the body to de-drift estimates ${}^0\hat{\boldsymbol{p}}_b$, ${}^0\hat{\boldsymbol{v}}_b$, and ${}^0\hat{\boldsymbol{p}}_i$ for each foot. Letting ${}^0\boldsymbol{p}_{rel}(\boldsymbol{q}_i, {}^0\hat{\boldsymbol{R}}_b)$ denote the relative foot position as computed by kinematics, a measurement residual is generated

$$\boldsymbol{e}_{p,i} = \left({}^0\hat{\boldsymbol{p}}_i - {}^0\hat{\boldsymbol{p}}_b \right) - {}^0\boldsymbol{p}_{rel}(\boldsymbol{q}_i, {}^0\hat{\boldsymbol{R}}_b)$$

Similarly, the velocity of the foot relative to the body can be computed from the leg angles, velocities, and the body orientation and angular velocity. This computation is denoted as ${}^0\dot{\boldsymbol{p}}_{rel}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, {}^0\hat{\boldsymbol{R}}_b, {}^b\boldsymbol{\omega}_b)$. Under the assumption that each foot is fixed, this provides an associated measurement residual

$$\boldsymbol{e}_{v,i} = \left(-{}^0\hat{\boldsymbol{v}}_b \right) - {}^0\dot{\boldsymbol{p}}_{rel}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, {}^0\hat{\boldsymbol{R}}_b, {}^b\boldsymbol{\omega}_b)$$

Finally, a contact height h_i is assumed for each foot with associated measurement residual:

$$\boldsymbol{e}_{h_i} = \left([0,\ 0,\ 1]^T\, {}^0\hat{\boldsymbol{p}}_i \right) - h_i$$

Random multivariate Gaussian's are used to capture the measurements errors for each residual. Similar to the process noise $\boldsymbol{w}_{\boldsymbol{p}_i}$ on the feet, measurement covariances for foot i are increased to a high value during swing such that the swing leg measurements are effectively ignored in this fusion process. Although space permits a full development, the residuals are used within the innovations step of a Kalman filter to provide estimates ${}^0\hat{\boldsymbol{p}}_b$, ${}^0\hat{\boldsymbol{v}}_b$, and ${}^0\hat{\boldsymbol{p}}_i$ used for feedback to the Balance Control or MPC components.

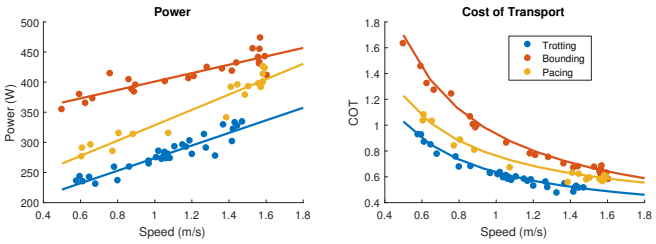

Fig. 5: **Power consumption and Cost of Transport for Dynamic Gaits.** Scattered points correspond to 1 second of averaged data. Lines are a linear fit to power consumption.

IV. Results

This section presents results from experimental testing with Cheetah 3. The selected results highlight key features of the system: the ability to control different dynamic gaits within the same framework, design-enhanced capabilities that leverage Cheetah's large range of motion, and the ability to traverse stairs without external terrain information. A video of the results is included as supplementary material.

A. Steady State Locomotion

With locomotion being the primary function of the robot, a number of experimental trials were run with three different gaits at varying speeds to quantify the efficiency of the robot during steady state behavior. Trotting is achieved through the use of both the instantaneously reactive QP controller described in Section III-C, as well as the convex MPC described in Section III-D. Bounding and pacing gaits are possible only through the use of the convex MPC as they require predictive knowledge of the upcoming flight times to remain balanced throughout the gait.

Experiments were held indoors on a flat treadmill with adjustable velocities. Power consumption was logged by measuring battery current, and does not include power consumption of the computer, which is an additional 85 watts. The gaits and controllers tested used were not optimized for CoT or efficiency. CoT is calculated as $\frac{P}{mgv}$, where v is the linear velocity of the robot, m is the mass of the robot, and P is electrical power consumption. Experimentally determined CoT results for each of the three gaits are shown in Figure 5. As opposed to animals which use internal springs and muscles that are efficient in locking, the robot's electric motors are dominated by force production cost rather than the mechanical CoT. Therefore, the most efficient gait for the robot at a given speed does not necessarily correspond to the one used by animals as determined in [24].

B. Design-Enhanced Capabilities

The robot's design and robust control architecture allow for a variety of unique behaviors. Important improvements to the design over the MIT Cheetah 2 greatly enhance the possibilities for this new robot to be used in real-world applications. Since the robot ground reaction force control planning is independent of the leg configuration, locomotion can occur regardless of which way the knees are pointed as shown in Figure 6a. With the new design incorporating the powerful custom electric motor for the ab/ad DoF, the robot gains the ability to apply large lateral forces. The robot gains an improved range of motion that allows for exaggerated yaw motions that are depicted in Figure 6b. In addition, having no defined forward or top, the robot is able to rotate its legs fully above its body. Using the chain in the leg design gives the robot the ability to omni-directionally rotate the knee joint as in Figure 6c. In the future this can be taken advantage of to allow the robot to use its legs for both locomotion as well as to interact with its surroundings and carry out tasks.

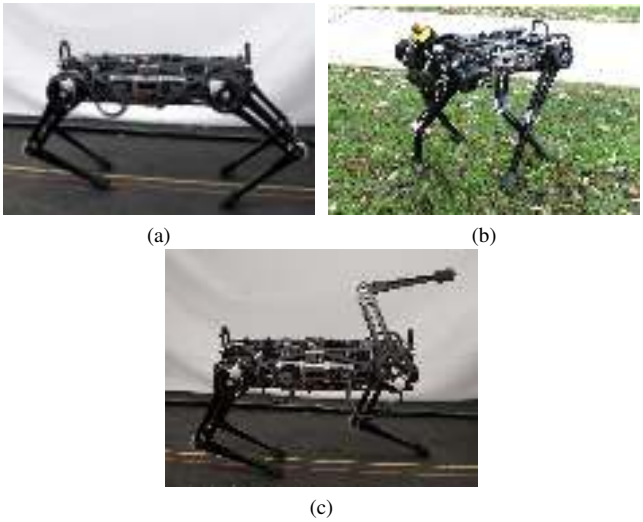

Fig. 6: **Unique Design Capabilities.** The robot is able to use the controllers described regardless of leg orientation (a), balance in extreme configurations (b), and make use of a wide range of motion to position its limbs (c).

C. Blind Stair Climbing

Although the robot has no preemptive knowledge of the terrain, the control architecture described allows robust locomotion over highly unstructured and abrupt terrain changes such as staircases. Since there are no external sensors, the robot assumes a flat, unobstructed world, so it must act reactively to its current sensed state when encountering unforeseen obstacles. With the contact detection algorithm mentioned in Section III-A, unexpected collision with objects such as stairs can be detected and used to modify the swing foot trajectories and balance strategy through the event-based gait scheduler. Once the robot senses the step, it adjusts its posture based on Section III-G and expects contacts to be along the estimated plane.

A staircase was built for testing with the US standard of a 7in (17.8cm) rise and 11in (27.9cm) run. Figure 7 shows the robot during a stair climbing experiment as it successfully reaches a platform at the top of the staircase. Several tests were run in which the robot successfully climbed up the stairs with failures being due to operator errors attempting to take the robot outside of its kinematic limits rather than due to failures in the control architecture itself. The attached video demonstrates the ability of the robot to reject large disturbances as it is pulled with a rope off of the steps. In all cases it regains balance and continues to try to climb the

Fig. 7: **Blindly Climbing Stairs.** With the control framework described throughout the paper, the robot was able to successfully climb up stairs without the use of external environment sensors.

steps after being forced off of its intended objective. This ability to reactively sense and adapt to unexpected obstacles indicates that the robot is capable of robustly operating and adapting to realistic, unstructured, and changing terrains without relying on precise perception and planning.

V. Conclusions and Future Work

This paper has introduced the MIT Cheetah 3 quadruped robot as a new legged robot platform. An improved design over MIT Cheetah 2, this robot has a larger range of motion with full 3D control capabilities. The robustness demonstrated without the use of external sensing signifies that it is capable of successful locomotion in challenging situations without relying on knowledge of the environment prior to executing its control strategy.

The general control architecture lends itself to the use of any number of different controllers with minimal modifications to the software or hardware. Promising initial results of a novel nonlinear Policy Regularized Model Predictive Control framework (PR-MPC) further expand on the current controllers presented by optimizing both ground reaction forces as well as footstep locations using simple physics-based heuristics to regularize the prediction [25]. The robot will intelligently decide on where to best place the feet for any given gait and take advantage of its dynamics for steady state locomotion and disturbance rejection.

While the robot currently operates by taking in user inputs for commanded velocity and turning rate, a higher level path planner will be developed to allow autonomous operation in the world. This higher level planner can be naturally integrated in the framework both as a path planner, as well as providing information for enhancing the controllers and state estimation.

References

[1] "DARPA Robotics Challenge." http://www.theroboticschallenge.org/, Nov. 2012.

[2] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 38–44, Oct 2016.

[3] H.-W. Park, P. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Proceedings of Robotics: Science and Systems*, (Rome, Italy), July 2015.

[4] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, pp. 1–27, 2015.

[5] N. A. Radford and et al., "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[6] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and the BigDog Team, "Bigdog, the rough-terrain quaduped robot," in *Proceedings of the 17th World Congress*, pp. 10822–10825, 2008.

[7] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max," *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 635–646, April 2017.

[8] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the MIT Cheetah 2: Control design and experiments," *International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.

[9] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the MIT Cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[10] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4057–4062, Oct 2010.

[11] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *IEEE Int. Conf. on Robotics and Automation (to appear)*, 2018.

[12] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, vol. 41, pp. 259–272, Jan 2017.

[13] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 52–59, Dec. 2010.

[14] F. Bullo and R. M. Murray, "Proportional derivative (PD) control on the Euclidean group," in *Proc. of the European Control Conference*, (Rome, Italy), pp. 1091–1097, June 1995.

[15] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[16] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3287–3292, May 2013.

[17] J. DiCarlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the MIT Cheetah 3 through convex model predictive control." Submitted to IROS 2018.

[18] M. H. Raibert, *Legged robots that balance*. Cambridge, MA, USA: MIT Press, 1986.

[19] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS Int. Conf. on Humanoid Robots*, (Genova, Italy), pp. 200–207, Dec. 2006.

[20] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots - consistent fusion of leg kinematics and IMU," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[21] T. Flayols, A. D. Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, "Experimental evaluation of simple estimators for humanoid robots," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 889–895, Nov 2017.

[22] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[23] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing," in *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 112–119, Nov 2014.

[24] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3307–3312, May 2013.

[25] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT Cheetah," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017.

View publication stats