Implementation of a Robust Dynamic Walking Controller on a Miniature Bipedal Robot with Proprioceptive Actuation

Junjie Shen[1], Jingwen Zhang[1], Yeting Liu[1], and Dennis Hong[1]

Abstract— **Developing a robust dynamic walking controller for bipedal robots remains challenging as the system is hybrid, highly nonlinear, and strongly restricted. The typical two-level structure of high-level footstep planning and low-level whole-body control has been proven an effective approach for bipedal locomotion. However, practical guidance on its implementation is rarely covered fully in detail. To bridge this gap, this paper presents a detailed implementation of such controller for dynamic walking applications on a miniature bipedal robot with proprioceptive actuation. To the best of our knowledge, this is the first fully-untethered miniature bipedal robot which can achieve robust dynamic walking using this framework. In particular, the high-level planner determines both the location and duration for the next few steps based on the divergent component of motion. The low-level controller leverages the full-body dynamics to establish the foot contact as planned while regulating other task-space behaviors, e.g., center of mass height and torso orientation. Both problems are formulated as small-scale quadratic programs, which can be solved efficiently with guaranteed optimality for real-time execution. Extensive results of simulation and hardware walking experiments are provided to demonstrate the strong robustness of the approach under various disturbances and uncertainties, e.g., external pushes and irregular terrains.**

I. Introduction

Bipedal locomotion has been studied for decades and yet it remains an active research field. Besides the great demand for a reliable hardware platform, various challenges emerge in developing an efficient control algorithm, e.g., the complex robot model and strict real-time requirements.

In the presence of strong perturbations, when the ankle [1] and hip [2] strategies are no more effective, it is necessary to take recovery steps to avoid falling. While linear feedback control [3] or simple heuristic [4] works for small footstep adjustment, a real-time trajectory optimization (TO) scheme outperforms them with the consideration of a time prediction horizon and various physical constraints. However, TO-based approaches heavily depend on system models of the process. Because of the complexity of bipedal robots, e.g., hybrid dynamics, high nonlinearity, strong restrictions, and considerable degrees of freedom (DoF), solving the problem holistically is extremely challenging even offline, i.e., optimizing over contact schedule and contact forces simultaneously with the full-order model [5], [6].

To date, most successful online locomotion control strategies break down the problem into multiple stages and simplify the robot model according to the use of each stage, sacrificing the global optimality while achieving the real-time execution [7]–[9]. A practical paradigm is to have a two-level structure [10]–[14], where the high level plans the footsteps while the low level establishes the foot contact at the desired time and location.

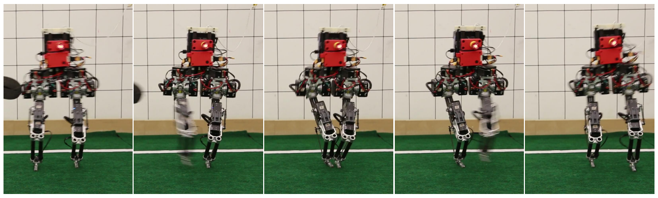

Fig. 1. Snapshots (from left to right) of BRUCE taking recovery steps after its torso was pushed to the right.

The linear inverted pendulum model (LIPM) [15] is widely adopted for the high-level footstep planning, which captures the most salient aspect of the system dynamics, i.e., center of mass (CoM). Considering piecewise constant CoM jerk as input, the LIPM dynamics can be discretized as cubic polynomials. The footstep locations can then be solved through TO in real time, which can be transcribed into a quadratic program (QP), minimizing the overall jerks while restricting the center of pressure (CoP) within the support polygon [16], [17]. Assuming invariant CoP coincident with the footstep location, the LIPM dynamics can be analytically integrated so that the intermediate process can be omitted and thus only an order of magnitude less decision variables are involved in the QP [8], [10]. Nevertheless, in all these strategies, step duration is always fixed to maintain the convexity of the TO problem. Notably, the LIPM dynamics can be decomposed into two parts, one is stable and the other is unstable, and controlling only the unstable part, i.e., the so-called divergent component of motion (DCM), is enough to generate stable walking [18], [19]. Unlike the full CoM states including both position and velocity, the DCM follows a first-order dynamics, which helps simplify other aspects of locomotion analysis and control. Based on the DCM, both location and timing of the next footstep can be adapted using a QP-based TO formulation [12], [14].

For simultaneous execution of multiple tasks in the low level, whole-body control (WBC) is a good candidate which exploits the full capabilities of the entire body of redundant, floating-based robots interacting with the environment. In addition, it is able to realize fast, agile, and compliant motions yet without sacrificing accuracy. Previous approaches [20],

[1] Junjie Shen, Jingwen Zhang, Yeting Liu, and Dennis Hong are with the Robotics and Mechanisms Laboratory, the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA 90095, USA {junjieshen, zhjwzhang, liu1995, dennishong}@ucla.edu

[21] use inverse dynamics (ID) but consider trajectories in joint space, resulting in an incredible amount of required motion details. By contrast, task-space control [22], [23] eases this burden by designing trajectories in the intuitive task space. However, for all these methods, contact stability and torque limits are not properly handled, the violation of which can easily cause poor performance or even control failure. Lately, an elegant way to implement WBC is using optimization, mostly QP, which is able to account for system dynamics, map between the task space and joint space, as well as satisfy various constraints concurrently. Depending on how the task hierarchy is managed, QP-based WBC can be mainly categorized into two types. A weighted WBC scheme sets all operational tasks as objectives of a single QP with priorities implicitly being enforced with weights [7], [11]. A strictly hierarchical framework solves cascaded QPs from the highest priority to the lowest, where the subsequent QP is carried out with additional constraints to preserve the optimality of the previous one [24], [25].

II. Motivation & Contribution

Comparatively slower development in bipedal robots than quadruped robots in terms of dynamic behaviors is being noticeable. Besides the limited accessibility to reliable hardware, one of the main reasons is that bipedal robots impose a serious challenge on developing dynamic motion controllers due to the required real-time coordination of multiple tasks in a complex, high-dimensional state space. The two-level structure described in Section I for locomotion purpose has been proven an effective approach for many bipedal robots, e.g., Atlas [11], Cassie [13], and Sarcos [14]. However, practical guidance on implementation of such framework is rarely covered fully in detail or out of date [26], [27]. We believe that complete discussions on implementation details, including but not limited to parameter selection experience and matters needing attention, are also valuable to promote research on dynamic behaviors of bipedal robots.

Meanwhile, investigation on the performance of the state-of-the-art bipedal control strategy for platforms in different scales, e.g. full-size and miniature bipedal robots, is not done yet. Differences between them may put new challenges on both control algorithms and hardware design principle. One of the obvious differences is the lower CoM height of miniature bipedal robots, which will lead to higher natural frequency and quicker diverging speed. As for dynamic walking applications, step duration must be reduced accordingly to catch the fall in time but too fast swing trajectories may degenerate the tracking performance.

We presents this paper to discuss similar issues for implementation of a two-level dynamic walking controller on a miniature bipedal robot. The high-level footstep planner is based on the DCM where both the step location and timing can be optimized for stronger robustness. The low-level WBC is using a weighted QP scheme in consideration of computational cost despite sacrificing the strict task priority. Fig. 2 shows the block diagram for the entire system.

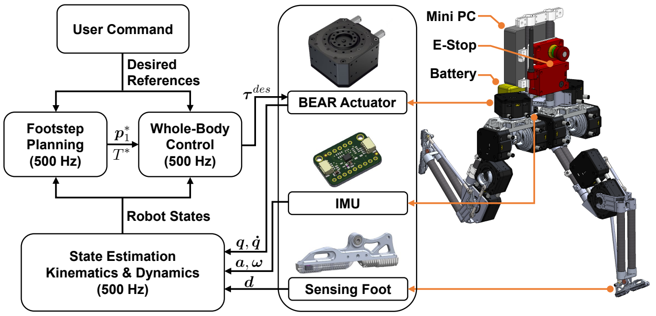

Fig. 2. BRUCE system overview. The robot is designed to be fully untethered, powered by a LiPo battery, controlled by a mini PC, and equipped with a wireless E-Stop. The state estimator uses a complementary filter that fuses information from the onboard IMU (acceleration $\boldsymbol{a}$ and angular rate $\boldsymbol{\omega}$), the joint encoders (position $\boldsymbol{q}$ and velocity $\dot{\boldsymbol{q}}$), and the sensing feet (boolean contact states $\boldsymbol{d}$). Given the current robot states and desired references, the high-level footstep planner determines the next footstep location $\boldsymbol{p}_1^*$ and step duration T^* while the low-level WBC calculates the desired joint torques $\boldsymbol{\tau}^{des}$.

III. System Overview

A. Bipedal Robot Unit with Compliance Enhanced

Bipedal Robot Unit with Compliance Enhanced (BRUCE) [28] is a low-cost miniature bipedal robotic platform for dynamic behaviors. BRUCE has a total height of 660 mm adapting an average human body proportion. It is composed of a torso and two five-DoF legs. Each leg of BRUCE has a spherical hip joint, a single DoF knee joint, and a single DoF ankle joint. Accordingly, each foot has a line contact with the ground. To have better torque transparency and compliance to the unknown environment, proprioceptive actuation is equipped for each joint, using the BEAR actuators [29] from Westwood Robotics [30]. Moreover, BRUCE's legs are designed to have low inertia for performing highly dynamic motions. A two-DoF parallel actuation configuration realized by a cable-driven differential pulley system is applied to the hip joint to reduce the mass and inertia of the femur link. Meanwhile, two pairs of four-bar linkage mechanisms are used to relocate the ankle actuator to the femur link for the light weight of the tibia link.

To have a convenient hardware setup for walking experiments, it is desired to make BRUCE fully untethered. All the electronics are directly mounted onto the torso for easy installation and better weight distribution. A mini PC with an Intel Core i5-7260U Dual-Core CPU at 2.2 GHz is utilized as the onboard computing resource. A 14.8 V 2200 mAh LiPo battery is used to power the whole system with a running time of approximately 20 minutes. For the sake of safety during operation, a wireless E-Stop built in-house is mounted onboard as well to cut the power in emergencies. In total, BRUCE has a net weight of 5.2 kg.

To ensure reliable ground contact information for walking purpose, we modified the design of the sensing foot. Previously, BRUCE used a two-layer structure to mimic the working principle of an electronic switch [28]. However, the plastic contact layer can break from time to time due to touchdown impact. In addition, dust can easily stick to the copper foils, which affects their contact quality. The new design of the sensing foot has a tactile switch directly inserted into the rubber contact layer. In this way, the foot touchdown is more compliant and the contact detection mechanism is fully isolated from the outside environment, which greatly improves the contact sensing.

B. Software Architecture

The software architecture is developed in a multithreaded environment, which includes a state estimation thread combined with robot model computation, a high-level footstep planning thread, and a low-level WBC thread. Data communication utilizes a custom shared memory library [31]. All programs are implemented in Python while some parts, including kinematics, dynamics [32], and state estimation, are precompiled using Numba [33] for acceleration.

C. State Estimation

While Kalman filter has been widely used for legged state estimation in various forms [34], [35], complementary filter works robustly as well in practice [36]. The state estimator makes use of the joint encoders for stance leg kinematics and the onboard IMU. In specific, we use the IMU solely to estimate the base orientation and angular rate while we use both sensors for the base linear position and velocity estimation. Note that this simple approach comes with some practical issues, e.g., yaw drift, global position inaccuracy. However, it works fairly well locally.

D. System Identification

An accurate dynamic model is essential for BRUCE to perform highly dynamic motions since the controller heavily relies on it. Meanwhile, for miniature robots, CAD measurement is less accurate due to the relatively high ratio of the electronics and accessories. Accordingly, mass-inertial parameters are obtained by performing a system identification which is formulated as a least-squares problem with the joint measurement along some excitation trajectories [37]. However, ill-conditioned observation matrix can lead to inaccurate identification. To mitigate this issue, an optimal excitation trajectory is generated by minimizing the condition number of the observation matrix based on a parameterized trajectory with finite fourier series [38], [39], while regularization towards the nominal values obtained from the CAD model is also considered. The regularized least-squares problem is formulated as follows:

$$\min_{\boldsymbol{\Psi}} \ \|\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\Psi}-\boldsymbol{\tau}\| + \lambda\, R(\boldsymbol{\Psi},\boldsymbol{\Psi}_0)\,, \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\tau}$ are respectively the measured joint positions, velocities, accelerations, torques, $\boldsymbol{\Gamma}$ is the state-dependant observation matrix, and $\boldsymbol{\Psi} \in \mathbb{R}^{10n}$ denotes the mass-inertial parameters for an n-link system including mass, first mass moment, and rotational inertia tensor [37]. As for the regularization term R with weight λ, we accept the formulation in [40] to make the difference between $\boldsymbol{\Psi}$ and its nominal value $\boldsymbol{\Psi}_0$ physically consistent but still convex.

IV. Footstep Planning

Using the DCM [18] defined as

$$\boldsymbol{\xi} := \boldsymbol{c} + \dot{\boldsymbol{c}}/\omega, \tag{2}$$

the LIPM dynamics can be split into two first-order differentiation equations:

$$\dot{\boldsymbol{c}} = \omega\left(\boldsymbol{\xi}-\boldsymbol{c}\right), \tag{3a}$$

$$\dot{\boldsymbol{\xi}} = \omega\left(\boldsymbol{\xi}-\boldsymbol{p}\right), \tag{3b}$$

where $\boldsymbol{c}$ is the CoM horizontal position, $\boldsymbol{p}$ is the CoP location on the ground, and ω is the natural frequency of the pendulum. The dynamics (3a) is stable, i.e., the CoM always follows the DCM, whereas the dynamics (3b) is unstable, i.e., the DCM is pushed away by the CoP, which implies that controlling only the unstable part is enough to generate stable walking. Notably, the DCM dynamics (3b) can be analytically solved as an initial value problem with the initial DCM state $\boldsymbol{\xi}_0$ and the fixed CoP location $\boldsymbol{p}_0$:

$$\boldsymbol{\xi}(t) = \boldsymbol{p}_0 + \left(\boldsymbol{\xi}_0 - \boldsymbol{p}_0\right)e^{\omega t}. \tag{4}$$

To stabilize the DCM dynamics, it is proposed in [12] to adapt both the location and timing of the next footstep using QP. While it is sufficient to only consider the next step to recover from any viable state in unconstrained environments, a multistep horizon will enhance the transient behavior such that faster and smoother response can be achieved in case of disturbances, e.g., external push or sudden change in reference. Indeed, a preview of three steps works best according to our experience in terms of system robustness, transient behavior, and computational efficiency, as shown in the supplementary video. Accordingly, we extend the original framework by planning multiple steps ahead of time while keeping the problem convex, i.e., still as a QP:

$$\begin{aligned}
\min_{\boldsymbol{p}_k,\boldsymbol{b}_k,\eta} \quad & \sum_{k=1}^{N_s} \|\boldsymbol{b}_k - \boldsymbol{R}_k\boldsymbol{b}_k^{nom}\|_{\boldsymbol{W}_b}^2 + \|\boldsymbol{l}_k - \boldsymbol{R}_k\boldsymbol{l}_k^{nom}\|_{\boldsymbol{W}_l}^2 \\
& + w_\eta\left|\eta - e^{\omega T^{nom}}\right|^2 && \text{(5a)}\\
\text{s.t.} \quad & \boldsymbol{\xi}_1 = \boldsymbol{p}_0 + \boldsymbol{b}_0 e^{-\omega t_0}\eta, && \text{(5b)}\\
& \boldsymbol{\xi}_k = \boldsymbol{p}_{k-1} + \boldsymbol{b}_{k-1}e^{\omega T^{nom}},\ k=2,\dots,N_s, && \text{(5c)}\\
& \boldsymbol{l}_{\min} \le \boldsymbol{R}_k^\top \boldsymbol{l}_k \le \boldsymbol{l}_{\max},\ k=1,\dots,N_s, && \text{(5d)}\\
& e^{\omega T_{\min}} \le \eta \le e^{\omega T_{\max}}, && \text{(5e)}
\end{aligned}$$

where $\boldsymbol{b}_k = \boldsymbol{\xi}_k - \boldsymbol{p}_k$ is the DCM offset, $\boldsymbol{l}_k = \boldsymbol{p}_k - \boldsymbol{p}_{k-1}$ is the step difference, $\boldsymbol{R}_k$ is the planar rotation matrix for steering at the kth step, T is the step duration, $\eta = e^{\omega T}$ is the corresponding change of variable for convexity, N_s is the number of steps, and the superscript $(\cdot)^{nom}$ corresponds to the nominal gait pattern [12]. In specific, the cost (5a) minimizes the overall tracking errors with different weights for each term, where the weighted vector norm square is defined as $\|\boldsymbol{e}\|_{\boldsymbol{W}}^2 := \boldsymbol{e}^\top\boldsymbol{W}\boldsymbol{e}$ for $\boldsymbol{e} \in \mathbb{R}^n$ and $\boldsymbol{W} \in \mathbb{S}_+^n$. Table I shows the weights for our system. The constraint (5b) predicts the DCM state at upcoming touchdown with step duration adapted based on (4) where t_0 is the phase elapsed time, (5c) further predicts the subsequent DCM evolution

TABLE I

FOOTSTEP PLANNING WEIGHT SETUP

$\boldsymbol{W}_b$	$\boldsymbol{W}_l$	w_η
$(100, 100)$	$(1, 1)$	0.1

Note that $\boldsymbol{W}_b$ and $\boldsymbol{W}_l$ are both 2×2 diagonal positive semi-definite matrices so only the diagonal elements are listed in the order of x and y directions. In addition, $\boldsymbol{W}_b$ typically needs to be set significantly larger than $\boldsymbol{W}_l$ as it is essentially responsible for gait viability[2].

with step duration fixed for convexity, (5d) ensures each footstep is physically reachable, and (5e) bounds the adapted step duration. Note that the walking gait is predefined which is alternating left- and right-stance phases with no double-stance phase. We will now discuss the key factors of the footstep planning.

A. Center of Mass Height

The CoM height c_z of LIPM essentially defines its natural frequency ω:

$$\omega := \sqrt{g/c_z}, \tag{6}$$

where g is the gravitational acceleration. Notably, ω is more sensitive to c_z when c_z is small. In view of the miniature size of BRUCE, ω is thus updated at each control loop but is still considered constant to keep the problem convex.

Meanwhile, based on (6), it is desirable to have c_z as high as possible so that ω is low and the DCM can diverge at a slow pace per (4), which benefits the overall system since it gives the robot more time to respond. In addition, a higher c_z can relieve the knee from bending too much, making the walking more energy-efficient and natural, which however gives rise to the knee singularity issue. We plan to solve this problem in the future but for now we just choose c_z so that singularity can hardly happen in most scenarios.

B. Center of Pressure Location

For bipedal robot locomotion, accurate modulation of the CoP is limited, not to mention BRUCE with five-DoF legs and tiny line feet. In addition, BRUCE's feet are not equipped with F/T sensors for direct CoP measurement. Accordingly, we do not specifically sample in time to take into account variable CoP. We simply consider the center of the stance foot as the CoP location, which works just fine in practice.

C. Step Duration

The nominal step duration T^{nom} has a significant effect on the nominal gait pattern. The numerical simulation results of LIPM are shown in Fig. 3. We can see that for BRUCE with a CoM height of only around 0.3 m, its natural frequency is much higher than that of a regular full-size humanoid robot per (6), i.e., the DCM diverges faster per (4), and thus T^{nom} is typically expected shorter for a reasonable walking gait. In addition, the robot CoM tends to swing more in the lateral direction as T^{nom} increases, which is undesirable in consideration of the following reasons. First, the real robot does not behave as the LIPM exactly especially when T^{nom} is long and thus the nominal gait pattern becomes meaningless. One can compare the DCM trajectories of Fig. 3 and Fig. 4. We do observe walking of BRUCE gets less stable as T^{nom} increases. Second, even if we try to make the robot naively mimic the LIPM, the performance is poor because we cannot fully control the CoM states as we wish especially when the robot is falling, not to mention the tiny line feet we have on BRUCE. In the end, we find T^{nom} of around 0.22 seconds works best on BRUCE.

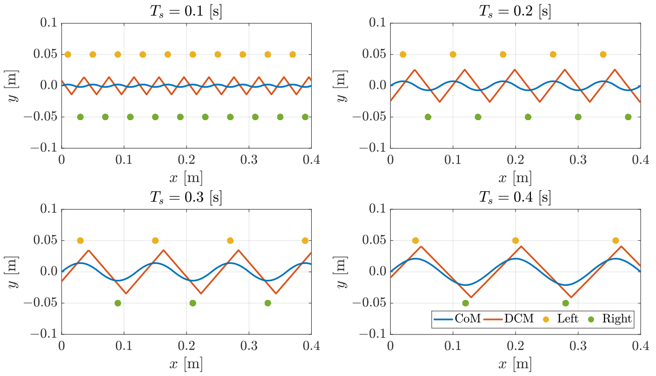

Fig. 3. Nominal gait pattern with different step durations. Note that the robot is walking to the positive x direction with a fixed average CoM speed of 0.1 m/s, a fixed CoM height of 0.3 m, and a fixed step width of 0.1 m.

When the robot is perturbed and deviating from the nominal gait pattern, it is able to recover better by adapting the step duration in addition to the footstep location. We set bounds to the adapted step duration for behavior regulation as (5e). The minimum prevents large swing foot acceleration which mostly depends on the actuator specifications, e.g., motor constants, peak torque. The maximum avoids slow stepping which rarely occurs according to our experience. Meanwhile, to prevent instantaneous stepping, we stop adapting if the remaining time of the current step is less than a threshold of 0.05 seconds.

V. WHOLE-BODY CONTROL

The joint-space equations of motion for a bipedal robot can be written in the following canonical form:

$$\boldsymbol{H}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G} = \boldsymbol{S}_a^\top \boldsymbol{\tau} + \sum_{j=1}^{N_c} \boldsymbol{J}_{c_j}^\top \boldsymbol{f}_j, \tag{7}$$

where $\boldsymbol{q}$ is the vector of generalized coordinates, $\boldsymbol{H}$ is the inertia matrix, $\boldsymbol{C}$ is the vector of centrifugal and Coriolis terms, $\boldsymbol{G}$ is the gravity vector, $\boldsymbol{S}_a$ is the actuation selection matrix, $\boldsymbol{\tau}$ is the joint torque vector, $\boldsymbol{J}_{c_j}$ and $\boldsymbol{f}_j$ are respectively the foot contact Jacobian and contact force at the jth contact vertex, and N_c is the number of contact vertices.

Given desired operational space acceleration $\ddot{\boldsymbol{x}}_i^{des}$ for the ith task, the goal of WBC is then to find the instantaneously required, dynamically consistent $\ddot{\boldsymbol{q}}$, $\boldsymbol{\tau}$, and $\boldsymbol{f}_j$. This can be

[2] The gait viability is originally imposed as a hard constraint, which bounds the DCM offset. However, it can overconstrain the problem along with the reachability constraint (5d). In practice, the viability constraint is omitted and instead, it is implicitly executed by setting $\boldsymbol{W}_b$ large to guide the planned DCM offset as close to the nominal value as possible.

done using a weighted WBC scheme, which is formulated as the following QP:

$$\min_{\ddot{\boldsymbol{q}}, \boldsymbol{f}_j} \; \sum_{i=1}^{N_t} \left\| \boldsymbol{J}_i \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i \dot{\boldsymbol{q}} - \ddot{\boldsymbol{x}}_i^{des} \right\|^2_{\boldsymbol{W}_i} + \sum_{j=1}^{N_c} \|\boldsymbol{f}_j\|^2_{\boldsymbol{W}_f} + \|\ddot{\boldsymbol{q}}\|^2_{\boldsymbol{W}_{\ddot{q}}} \tag{8a}$$

$$\text{s.t.} \quad \boldsymbol{S}_f \left(\boldsymbol{H}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G} - \sum_{j=1}^{N_c} \boldsymbol{J}_{c_j}^\top \boldsymbol{f}_j \right) = \boldsymbol{0}, \tag{8b}$$

$$\boldsymbol{f}_j \in \boldsymbol{\mathcal{C}}_j, \; j = 1, \cdots, N_c, \tag{8c}$$

where $\boldsymbol{J}_i$ is the ith task Jacobian and N_t is the number of tasks. As we can see, the ith operational task is set as a QP cost with priority implicitly being enforced with weight $\boldsymbol{W}_i$. In particular, $\ddot{\boldsymbol{x}}_i^{des}$ consists of both feedforward and feedback terms, which is specified with the form of

$$\ddot{\boldsymbol{x}}_i^{des} = \boldsymbol{a}_i^{ref} + \boldsymbol{K}_p\big(\boldsymbol{p}_i^{ref} - \boldsymbol{p}_i\big) + \boldsymbol{K}_d\big(\boldsymbol{v}_i^{ref} - \boldsymbol{v}_i\big), \tag{9a}$$

$$\ddot{\boldsymbol{x}}_i^{des} = \boldsymbol{\alpha}_i^{ref} + \boldsymbol{K}_p \mathrm{Log}\big(\boldsymbol{R}_i^\top \boldsymbol{R}_i^{ref}\big) + \boldsymbol{K}_d\big(\boldsymbol{\omega}_i^{ref} - \boldsymbol{\omega}_i\big), \tag{9b}$$

for the linear and angular motions, respectively, where $\boldsymbol{a}_i$, $\boldsymbol{v}_i$, $\boldsymbol{p}_i$ are the linear acceleration, velocity, position while $\boldsymbol{\alpha}_i$, $\boldsymbol{\omega}_i$, $\boldsymbol{R}_i$ are the angular acceleration, velocity, orientation, the superscript $(\cdot)^{ref}$ corresponds to the reference, and $\boldsymbol{K}_p/\boldsymbol{K}_d$ is the proportional/derivative (P/D) feedback gain matrix. Note that the logarithm operator Log : $\mathsf{SO}(3) \to \mathbb{R}^3$ converts a rotation matrix to its corresponding axis–angle representation. Table II shows all the task weights and gains for our system. In addition to the task costs, regularization costs are added to the decision variables $\ddot{\boldsymbol{q}}$ and $\boldsymbol{f}_j$ with small weights $\boldsymbol{W}_{\ddot{q}}$ and $\boldsymbol{W}_f$ respectively to ensure the overall QP cost is strictly positive definite even when the task Jacobians contain singularities, which avoids potential numerical issues. Let us now go over the details of the WBC framework.

A. System Dynamics

The consistency of the variables $\ddot{\boldsymbol{q}}$, $\boldsymbol{\tau}$, and $\boldsymbol{f}_j$ with the system dynamics (7) must be strictly enforced. Notably, the equations can be split into the floating base dynamics and joint dynamics. To accelerate the QP performance, only the floating base dynamics are considered as (8b), where $\boldsymbol{S}_f$ is the base selection matrix. In this manner, variables for $\boldsymbol{\tau}$ can be removed if it is assumed that there is always enough torque to achieve the generated motion, i.e., no torque limits. Once the QP is solved with optimal solution $\ddot{\boldsymbol{q}}^*$ and $\boldsymbol{f}_j^*$, the corresponding joint torques can be computed as follows:

$$\boldsymbol{\tau}^* = \boldsymbol{S}_a \left(\boldsymbol{H}\ddot{\boldsymbol{q}}^* + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G} - \sum_{j=1}^{N_c} \boldsymbol{J}_{c_j}^\top \boldsymbol{f}_j^* \right). \tag{10}$$

B. Stance Foot

For BRUCE with line feet, we consider two point contacts per foot, the toe and heel. The constraint (8c) ensures each contact force is bounded and lies within the local friction cone $\boldsymbol{\mathcal{C}}_j$ which is approximated by a square pyramid for linearity, e.g., on the even ground we have

$$\begin{bmatrix} \pm 1 & 0 & -\mu \\ 0 & \pm 1 & -\mu \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{f}_j \leq \begin{bmatrix} 0 \\ 0 \\ -f_{\min} \\ f_{\max} \end{bmatrix}, \tag{11}$$

where μ is the friction coefficient and $f_{\min}/f_{\max}$ is the minimum/maximum normal force. Note that a positive minimum can prevent loose contact, or otherwise one of the feet might lose contact if the CoM is shifted to the other one.

TABLE II

WBC Task Weight and Gain Setup

Task	$\boldsymbol{W}$	$\boldsymbol{K}_p$	$\boldsymbol{K}_d$
Linear Momentum	$(1, 1, 100)$	$(1, 1, 100)$	$(5, 5, 10)$
Angular Momentum	$(1, 1, 1)$	/	$(10, 10, 1)$
Torso Orientation	$(10, 10, 10)$	$(500, 500, 200)$	$(25, 25, 20)$
Stance Contact	$(10^3, 10^3, 10^3)$	/	/
Swing Position	$(10, 10, 10)$	$(100, 100, 100)$	$(10, 10, 10)$
Swing Orientation	$(/, 1, 1)$	$(/, 50, 300)$	$(/, 10, 50)$

Note that $\boldsymbol{W}$, $\boldsymbol{K}_p$, $\boldsymbol{K}_d$ are all 3×3 diagonal positive semi-definite matrices so only the diagonal elements are listed in the order of x, y, and z directions. The symbol / means the object is not used.

Moreover, for each contact force constraint, a corresponding zero contact acceleration constraint needs to be specified to prevent the stance foot from moving:

$$\boldsymbol{J}_{c_j}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{c_j}\dot{\boldsymbol{q}} = \boldsymbol{0}. \tag{12}$$

In practice, we treat (12) as one of the task-space objectives, i.e., a soft constraint, which can generally speed up the QP and give better numerical stability [7]. Then with sufficient task weight, it will act as a nullspace projector so that other tasks will properly respect the nonmoving contact condition.

C. Centroidal Momentum

The control of centroidal momentum is a critical component of WBC for bipedal robots, which consists of the linear momentum $\boldsymbol{l}$ as well as the centroidal angular momentum (CAM) $\boldsymbol{k}$ about the robot CoM. While the linear part has a straightforward relationship with the CoM velocity, the angular part is abstract. Notably, biomechanics studies have shown that for human walking, the CAM is well regulated to near zero by the neuro-control system [41]. Accordingly, the angular momentum task is to damp out excessive CAM:

$$\dot{\boldsymbol{k}}^{des} = -\boldsymbol{K}_d \boldsymbol{k}. \tag{13}$$

The linear momentum task is decoupled in the vertical and horizontal directions. For the vertical direction, we want the robot CoM to maintain a nominal height above the ground to match the LIPM in Section IV. For the horizontal direction, it is simply tracking the velocity command with low priority and PD gains since the robot movement is mainly realized by taking steps. Another reason is that our BRUCE robot has only five DoFs for each leg, which means during single stance not all the six spatial DoFs can be directly controlled. We prioritize the CoM height and torso orientation for a good posture, which generally contributes to the walking stability. Note that for the momentum tasks, the centroidal momentum matrix [42] is used as the task Jacobian.

D. Torso Orientation

Controlling the torso orientation is essential for a good walking posture, e.g., to avoid unwanted torso oscillations which can largely affect the robot CoM and complicate the control process. Accordingly, since we also have a small yaw drift rate, all the three angles are controlled globally.

E. Swing Foot

For a multi-layered control scheme, accurate execution of the high-level plan is important for the low-level controller. For our case, the high-level planner determines when and where to take the next step in an optimal manner, which is essentially realized by the swing foot task of WBC. Note that for the foot position, instead of considering some point on the foot bottom which is sensitive to the ankle joint, we choose to control the ankle position for simplicity. As a result, the ankle joint is only responsible for foot orientation. This decoupling is also beneficial for the tuning process.

1) Trajectory Generation: The swing foot orientation can be simply set constant relative to the torso. However, recall that each leg of BRUCE has only five DoFs and thus the rotation in the foot roll direction is excluded since it has the least effect for a line foot. In addition, because the robot is walking blindly with no terrain information, the gain in the pitch direction is intentionally set low, which can make the foot adaptive to a certain range of terrains.

The swing foot position trajectory needs to be carefully designed to adapt the changing footstep location and timing. In the horizontal direction, when a new step is planned with optimal solution $\boldsymbol{p}_1^*$ and T^*, the trajectory is regenerated using a fifth-order polynomial to ensure continuity up to acceleration, with boundary conditions given as follows:

$$\boldsymbol{p}_f^{ref}(t_0) = \boldsymbol{p}_f^{pre}(t_0), \quad \boldsymbol{p}_f^{ref}(T^*) = \boldsymbol{p}_1^*, \tag{14a}$$

$$\boldsymbol{v}_f^{ref}(t_0) = \boldsymbol{v}_f^{pre}(t_0), \quad \boldsymbol{v}_f^{ref}(T^*) = \boldsymbol{0}, \tag{14b}$$

$$\boldsymbol{a}_f^{ref}(t_0) = \boldsymbol{a}_f^{pre}(t_0), \quad \boldsymbol{a}_f^{ref}(T^*) = \boldsymbol{0}, \tag{14c}$$

where $\boldsymbol{p}_f^{pre}(t)$ is the previously generated trajectory. In the vertical direction, the swing foot height first increases to a fixed apex value until a fixed time, and then decreases to prepare for landing on the ground with a similar trajectory regeneration method in the horizontal direction. In addition, due to modeling and state estimation errors, the landing height of the swing foot needs to be adjusted based on the stance foot to mitigate the touchdown impact.

2) Inverse Kinematics Compensation: While ID-based WBCs such as ours are able to provide compliant behaviors and strong robustness, they heavily depend on the high quality of the dynamic model which is often difficult to obtain in practice. In addition, considering the short step duration for BRUCE, the swing foot acceleration is usually significant and thus accurate control of the foot movement is typically hard even with a good system identification. On the contrary, IK-based approaches only require the robot kinematic model which is much easier to get. On top of that, utilizing joint position PD control benefits bipedal systems due to its modeling error compensation and high updating frequency [7], [43], e.g., the BEAR actuator runs internal control loop at 2 kHz which is four times faster than our WBC. As a result, in addition to the optimal joint torques (10) from WBC, i.e., think of it as the feedforward term, we take into account the joint position and velocity references by solving the swing foot IK, which can greatly enhance the tracking performance:

$$\tau^{des} = \tau^* + k_p\big(q^{ref} - q\big) + k_d\big(\dot{q}^{ref} - \dot{q}\big), \tag{15}$$

where k_p/k_d is the P/D feedback gain for each joint.

F. Task Transition

During contact changes, i.e., foot transition from stance to swing and vice versa, task transition needs to be performed, e.g., for a stance foot, the stance contact task is activated while the swing position and orientation tasks are deactivated. This can be handled by simply changing the relative task weight, e.g., if a task is deactivated, its weight can be assigned zero or tiny value and we can easily bring it back if it becomes activated again. Note that foot contact transition happens instantaneously and thus smooth task transition [44] is unnecessary for the walking scenario. No chatter of the joint acceleration is observed either in the simulation or on the hardware. In addition, the contact force constraint (8c) needs to be adjusted accordingly as well, e.g., the limit $f_{\min}/f_{\max}$ should be reduced to zero for a swing foot.

VI. Experimental Results

In this section, an extensive series of simulation and hardware experiments were conducted to show the capability of our walking controller. The simulation of BRUCE is built based on Gazebo [45], an open-source 3D robotics simulator using the ODE physics engine. The hardware experiments were conducted fully untethered as introduced in Section III. The QPs (5) for the footstep planning and (8) for the WBC are both implemented with optimized Python code and solved using the off-the-shelf QP solver OSQP [46], which can achieve an updating frequency of 500 Hz, sufficient for real-time feedback control. The footage of all the experiments is available in the supplementary video.

A. Omnidirectional Walking

In this experiment, to verify DCM-based footstep planning is effective for generating stable walking motion in general, the robot was commanded and managed to walk omnidirectionally, e.g., forward and backward, left and right, stationary yaw rotation, as well as any combinations of them. Note that only a velocity command needs to be specified, e.g., sagittal and lateral velocity, yaw rate. In addition, due to modeling error, a velocity calibration is needed as the robot may drift slightly even with a zero velocity command.

B. Push Recovery

To gauge the overall system robustness in terms of external disturbance, a push recovery test was conducted. In the simulation, BRUCE was commanded to walk with a forward velocity of 0.3 m/s. At t around 2, 4, 6, 8 seconds, a constant

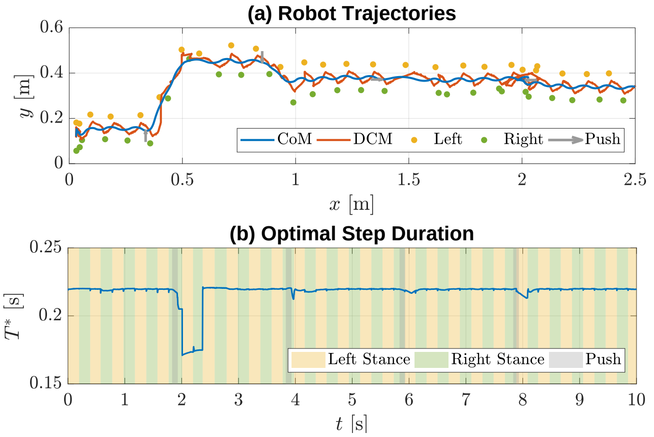

Fig. 4. Simulation results of push recovery. (a) Position trajectories of robot CoM, DCM, left and right feet. Note that the robot was locomoting to the positive x direction with a forward velocity command of 0.3 m/s. (b) Time series of the optimal step duration suggested by the footstep planner. The amber and green shaded areas indicate the left- and right-stance phases, respectively. The gray areas indicate the duration of the external pushes.

external force with a duration of 0.1 s and a magnitude of 12 N (BRUCE has a weight of only 5.2 kg) was exerted on its torso in the left, right, forward, and backward direction, respectively. The simulation results are shown in Fig. 4. As we can see, BRUCE was able to recover within the next few steps by adapting both footstep location and step duration. The push recovery test was also conducted on the real hardware. We pushed BRUCE on different parts, e.g., torso and leg, in various directions, e.g., sagittal and lateral, at random times with random durations, and BRUCE managed to survive. An example clip is shown in Fig. 1.

C. *Irregular Terrains*

In this experiment, to gauge the overall system robustness in terms of terrain uncertainty, BRUCE was challenged to a series of irregular terrains, as shown in Fig. 5. Note that BRUCE was walking blindly without any terrain information.

1) Uneven Terrain: In the simulation, BRUCE was commanded to walk with a forward velocity of 0.3 m/s and on the ground there were random wood slats with different sizes but a fixed height of 1 cm (BRUCE has a CoM height of only 30 cm). Thanks to the robust footstep planning and compliance of the WBC, BRUCE successfully conquered this uneven terrain. In the real world, BRUCE was also able to walk with small ground height variations.

2) Soft Terrain: In this scenario, BRUCE was stepping on a yoga mat in the real world. This kind of soft terrain is challenging as it is difficult for the stance foot to remain stationary due to the surface compliance, which can easily cause oscillations and even instability of the system. As we can see in the supplementary video, the stance leg (in particular the ankle) was constantly adapting in order to keep the balance and our walking controller was able to stabilize the system on this soft terrain.

3) Sliding Terrain: In this scenario, BRUCE was stepping on foam boards which can easily slide on the ground. The sliding can also mess up the state estimation which assumes fixed contact location. However, as we can see in the supplementary video, our walking controller could still stabilize the system with an adapted gait.

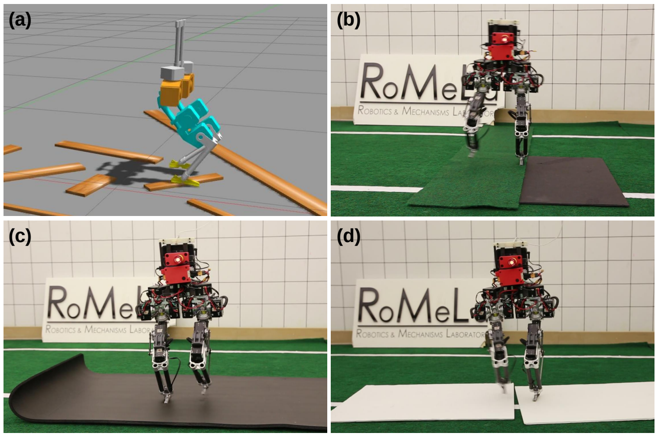

Fig. 5. Snapshots of BRUCE walking on irregular terrains. (a) Uneven terrain. (b) Height variation. (c) Soft terrain. (d) Sliding terrain.

VII. Conclusion and Future Work

In this paper, a two-level dynamic walking controller was presented. Specifically, the DCM-based high-level footstep planner is solving a TO with multiple steps being planned in advance, which optimally determines the footstep location and timing. The ID-based low-level WBC is finding the instantaneously required, dynamically consistent joint torques to best realize the task-space behaviors. Both problems can be transcribed into small-scale QPs which can be solved efficiently with guaranteed optimality for real-time applications. Additionally, detailed implementation guidance of the control framework is provided on BRUCE, a miniature bipedal robot with proprioceptive actuation. To the best of our knowledge, this is the first fully-untethered miniature bipedal robot which can achieve robust dynamic walking using this framework. We believe our practical experience and insightful discussion will benefit the robotics community. Lastly, an extensive series of simulation and hardware walking experiments were carried out, including omnidirectional walking, push recovery, and irregular terrains, which demonstrate the strong robustness of the approach in the presence of various disturbances and uncertainties.

In the future, an upgraded upper body with arms will be added on BRUCE for more capabilities, e.g., arm-assisted disturbance rejection, standing up on its own, and loco-manipulation. Other dynamic behaviors including running and jumping are under exploration as well.

Acknowledgment

This work was partially supported by the Office of Naval Research through grant N00014-15-1-2064.

References

[1] S. Kajita *et al.*, "Biped walking stabilization based on linear inverted pendulum tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, 2010.

[2] G. Wiedebach *et al.*, "Walking on partial footholds including line contacts with the humanoid robot Atlas," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 1312–1319, 2016.

[3] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying divergent component of motion," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 266–272, 2014.

[4] M. H. Raibert, H. B. Brown Jr, and M. Chepponis, "Experiments in balance with a 3D one-legged hopping machine," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984.

[5] J. Koenemann *et al.*, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3346–3351, 2015.

[6] M. Neunert *et al.*, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[7] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the Atlas robot," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 120–127, 2014.

[8] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Robust dynamic walking using online foot step optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5373–5378, 2016.

[9] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, "Variable horizon MPC with swing foot dynamics for bipedal walking control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2349–2356, 2021.

[10] S. Faraji, S. Pouya, C. G. Atkeson, and A. J. Ijspeert, "Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach," in *IEEE International Conference on Robotics and Automation*, pp. 1943–1950, 2014.

[11] T. Koolen *et al.*, "Design of a momentum-based control framework and application to the humanoid robot Atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 1, p. 1650007, 2016.

[12] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 35–42, 2016.

[13] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot Cassie," in *Robotics: Science and Systems*, vol. 101, p. 14, 2018.

[14] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Walking control based on step timing adaptation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 629–643, 2020.

[15] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 239–246, 2001.

[16] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1121–1126, 2008.

[17] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[18] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1084–1091, 2009.

[19] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[20] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *IEEE International Conference on Robotics and Automation*, pp. 3406–3412, 2010.

[21] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Control of legged robots with optimal distribution of contact forces," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 318–324, 2011.

[22] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots," in *IEEE International Conference on Robotics and Automation*, pp. 1963–1969, 2006.

[23] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, 2010.

[24] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[25] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[26] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong, "Development of open humanoid platform DARwIn-OP," in *SICE Annual Conference 2011*, pp. 2178–2181, 2011.

[27] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer Handbook of Robotics*, pp. 1203–1234, Springer, 2016.

[28] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong, "Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors," in *International Conference on Robotics and Automation*, pp. 8547–8553, 2022.

[29] T. Zhu, J. Hooks, and D. Hong, "Design, modeling, and analysis of a liquid cooled proprioceptive actuator for legged robots," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 36–43, 2019.

[30] Westwood Robotics. `https://www.westwoodrobotics.io`. Accessed: 2022-10-17.

[31] J. Yu, J. Hooks, X. Zhang, M. Sung Ahn, and D. Hong, "A proprioceptive, force-controlled, non-anthropomorphic biped for dynamic locomotion," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 1–9, 2018.

[32] R. Featherstone, *Rigid Body Dynamics Algorithms*. New York: Springer, 2008.

[33] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based python JIT compiler," in *Workshop on the LLVM Compiler Infrastructure in HPC*, 2015.

[34] M. Bloesch *et al.*, *State Estimation for Legged Robots: Consistent Fusion of Leg Kinematics and IMU*, pp. 17–24. MIT Press, 2013.

[35] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6058–6064, 2013.

[36] Y. Ding *et al.*, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[37] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

[38] W. Khalil and E. Dombre, *Modeling Identification and Control of Robots*. CRC Press, 2002.

[39] K.-J. Park, "Fourier-based optimal excitation trajectories for the dynamic identification of robots," *Robotica*, vol. 24, no. 5, pp. 625–633, 2006.

[40] T. Lee, P. M. Wensing, and F. C. Park, "Geometric robot dynamic identification: A convex programming approach," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 348–365, 2019.

[41] H. Herr and M. Popovic, "Angular momentum in human walking," *The Journal of Experimental Biology*, vol. 211, no. 4, pp. 467–481, 2008.

[42] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 1, p. 1550039, 2016.

[43] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," 2019.

[44] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1260–1277, 2012.

[45] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2149–2154, 2004.

[46] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.