Reachability Map for Diverse and Energy Efficient Stepping of Humanoids

Christopher McGreavy and Zhibin Li

***Abstract*—In legged locomotion, the relationship between different gait behaviors and energy consumption must consider the full-body dynamics and the robot control as a whole, which cannot be captured by simple models. This work studies the *totality* of robot dynamics and whole-body optimal control as a coupled system to investigate energy consumption during balance recovery. We developed a two-phase nonlinear optimization pipeline for dynamic stepping, which generates reachability maps showing complex energy-stepping relations. We optimize gait parameters to search all reachable locations and quantify the energy cost during dynamic transitions, which allows studying the relationship between energy consumption and stepping locations given different initial conditions. We found that to achieve efficient actuation, the stepping location and timing can have simple approximations close to the underlying optimality, resulting in optimal step positions with a 10.9% lower energy cost than those generated by linear inverted pendulum model. Despite the complexity of this nonlinear process, we found that near-minimal effort stepping locations are within a region of attractions, rather than a narrow solution space suggested by a simple model. This provides new insights into the nonuniqueness of near-optimal solutions in robot motion planning and control, and the diversity of stepping behavior in humans.**

***Index Terms*—Bipedal locomotion, energy-efficiency, optimal control, reachability maps.**

I. Introduction

TO ACHIEVE a long operation time for legged locomotion, it is essential to study the relationship between energy-efficiency and locomotion behaviors. The energy-efficiency and actuation power of a robot during locomotion are highly related to both its multibody dynamics and its control, which therefore must be considered together when studying energy consumption.

Manuscript received October 28, 2021; revised February 5, 2022; accepted April 29, 2022. Recommended by Technical Editor Shaohui Foong and Senior Editor Hong Qiao. This work was supported by the EPSRC CDT in Robotics and Autonomous Systems under Grant EP/L016834/1. *(Corresponding author: Christopher McGreavy.)*

Christopher McGreavy is with the School of Informatics, The University of Edinburgh, EH8 9YL Edinburgh, U.K. (e-mail: c.mcgreavy@ed.ac.uk).

Zhibin Li is with the Department of Computer Science, University College London, WC1E 6BT London, U.K. (e-mail: alex.li@ucl.ac.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2022.3174961.

Digital Object Identifier 10.1109/TMECH.2022.3174961

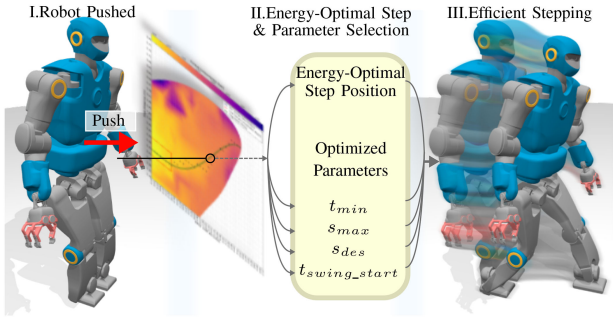


Fig. 1. Building an energy-optimal reachability map by optimizing parameters of whole-body control for online energy-optimal step selection in balance recovery.

For high-DoF robots, coupling hardware and software creates high dimensional state-action and solution spaces, such that studying energy consumption for general locomotion is complex and computationally heavy. This article investigates the stepping behavior and energy consumption and sheds light on this nonlinear relationship for a narrow aspect of locomotion—stepping and balance recovery—as a proof of concept.

To study energy-efficient stepping, we must consider the robot system and its dynamic behavior as a whole. Whole Body Motion Planning (WBM) [1] generates motion plans for the full robot system and considers its dynamics throughout the planning stage, and therefore, can be accurately tracked using Whole Body Control (WBC) [2] and can be formulated to enforce tasks like kinematic reachability [3], [4], collision avoidance [5], and centroidal dynamics [6], but plans are computationally expensive to generate. Mixed Integer Convex Optimization produces WBM plans for online use [3], [7] using conservative constraints such as static stability, but such motions are unsuitable for efficient walking. Therefore, though WBM methods could be used to optimize for energy-efficiency, it is impractical for the study of a large search space.

Alternatively, reduced models such as point mass models [9] can quickly produce reactive step plans [10] and stabilization control [11], and produce energy-efficient motion [12] at the planning level. Rapid computation comes at the cost of accuracy, as reduced models simplify the strong nonlinear and coupling effects of dynamics, therefore, optimality of planned motions is lost when projected to the whole system. We propose a proof-of-concept method that uses WBC to evaluate the true energy cost of tracking trajectories generated by reduced models instead of optimizing efficiency within the reduced model, resulting in a method of rapid trajectory generation with an accurate associated energy cost.

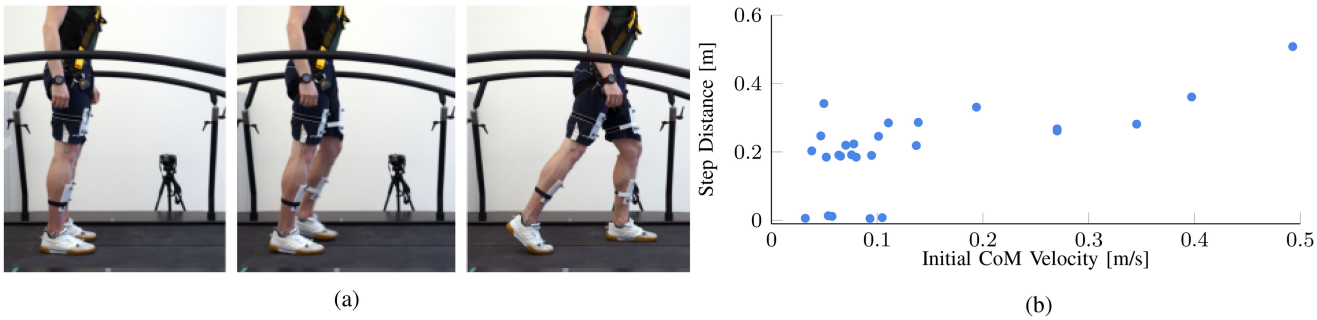


Fig. 2. Human stepping and balance recovery which is hard to model as a biological multibody system. (a) Subject taking a step during push recovery. (b) Distribution of step locations during push recovery [8].

Cost of transport (CoT), commonly used to quantify locomotion efficiency, is a dimensionless approximation of energy cost per unit weight of a body during locomotion, expressed as a single numerical value and is used as a unified metric to compare different bodies moving at different speeds [13]. CoT can aid robot design and control development [14], optimize gait for a given movement speed [15] and, at a high computational cost, for footstep planning [16]. But CoT can be misleading, as not all methods account for nonmechanical energy cost, such as heating loss—around 76% of total energy cost for quadrupeds [17]. We focus on single-step motions for which it is hard to define walking speed to calculate CoT, in simulation, where it is difficult to model nonmechanical energy loss, hence CoT is unsuitable in this case.

For a robotic system consisting of electric motors, the mechanical power of all joints is ultimately converted from electrical power, meaning the mechanical power consumption can be equally quantified by electrical consumption [18]. Joint torque is a good index for energy dissipation [19]–[21], since the torque is directly proportional to current. Therefore, given a constant voltage in a dc motor, torque can be used to reflect energy. In our simulation study, the sum of squared joint torque can be used to quantify whole-body energy use, which is integrated to give torque use during stepping.

We compile the realistic, whole-body energy cost of reduced models into reachability maps (see Fig. 1), which maps trajectories to energy costs and, inversely, automatically generates reduced model trajectories for step locations with optimal whole-body energy cost. Previous work has shown effective use of similar reachability maps for kinematic reachability [22], [23], feasible transition motions [24], [25], obstacle avoidance [5], [26], applied to faster locomotion planning [27], [28], complex end-pose planning [29], dynamic transitions [30], and warm-start learning [31]. Moreover, analysis of our maps shows simple heuristics and a diverse range of efficient stepping locations to plan motions with low computational cost.

Heuristics for energy-efficient step regions provide insights into complex stepping behaviors in humans. Fig. 2(b), from [8], shows step positions for human balance recovery (see Fig. 2(a)), offset by mean initial velocity of nonstepping trials (0.1103 m/s) and shows the stochasticity of human stepping [32], [33]. Our heuristics for the underlying optimality help explain the stochasticity of human stepping.

Our article is motivated to study the relationship between step location and energy optimality for balance recovery from a set of initial conditions by using a whole-body model to evaluate the energy cost of motions generated by simple models. Energy efficiency is not a primary concern in balance recovery, but by studying efficiency we find simple heuristics to rapidly select recovery step positions for full-body, dynamic motions which are limited when using simple models alone. We accurately quantify the energy cost of stepping for the whole stepping space for all joints in a 32-DoF robot in a realistic dynamic physics simulation to fully explore this relationship and develop a method of rapid, energy optimal step planning. We use Bayesian Optimization (BO), a sample efficient, nonlinear optimization method suited to whole-body tasks [34] to optimize open parameters for efficient stepping.

A. Scope

We develop an optimization pipeline (see Fig. 1) and focus on the following set of stepping motions: taking a single step forward from an initialized standing position with the center of mass (CoM) above stance foot and without toe-off motions. This allows us to search for optimal locomotion parameters given different initial states, and obtain results of the torque usage and energy consumption of a full-body humanoid robot with redundancies.

B. Contributions

This article develops a nonlinear optimization pipeline and studies the efficiency of power consumption in humanoid stepping. Our contributions are as follows:

1) Energy-optimal stepping (Section IV-B3): Reachability maps that show energy-optimal step positions based on whole-body dynamics and the use of optimal control.
2) Optimization pipeline (Sections IV-A and IV-B): A parallelized optimization pipeline for whole-body control based stepping in full dynamics in a physics simulation.
3) Reachability maps (Section IV-B2): A proposed method for computing precise reachability maps for dynamic motions which can rapidly select energy-efficient step locations.
4) Finding of simple approximation of optimal stepping (Section IV-B2): Finding of simple approximations and the disclosure of a funnel of near-optimal step locations.

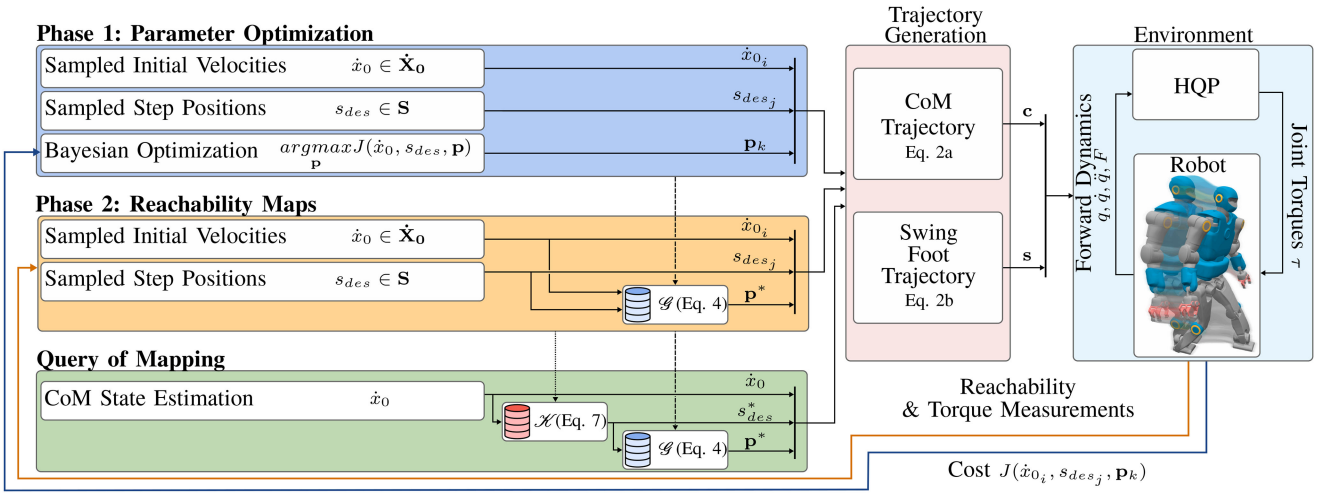


Fig. 3. Pipeline for optimizing trajectory generation and HQP parameters. Phase 1 optimizes trajectory generation parameters for sets of motions and outputs a parameter mapping. Phase 2 uses this mapping to build reachability maps and an energy-optimal step selection mapping; both mappings are combined for energy-optimal push recovery motions.

Our sampling-based energy cost maps reveal a complex, nonlinear distribution of efficient stepping locations, surrounded by regions of similar efficiency, indicating a simple step selection heuristic for balance recovery—which can only be studied and understood by considering the complexity of whole-body dynamics and optimal control as a whole. We find optimal step regions are different from those predicted by simple models which do not capture whole-body dynamics. Reachability maps can also be used for rapid step selection and give insight into nonuniqueness in human step selection.

The rest of this article is organized as follows. We mathematically define the pipeline and parameterized control system in Section II, then describe the methods of the pipeline in more detail in Section IV. We then present our results in Section V, followed by our discussion in Section VI. Finally, Section VII concludes this article.

II. Problem Formulation

To study energy-efficiency during stepping, we generate trajectories for the CoM and swing foot using simplified models and evaluate their energy cost by tracking them with a whole-body feedback controller in a full dynamic simulation. Trajectories each have two open parameters and BO is used to optimize the open parameters for energy-efficient stepping, we then produce maps to associate pairs of initial conditions and trajectories to energy cost. We achieve this using our two-phase pipeline, defined mathematically below.

A hierarchical quadratic programming (HQP) feedback controller [35] is used to map the simple trajectories onto a complex whole-body system with low computational cost due to its quadratic formulation. The HQP $\mathscr{H}$ tuned for dynamic stepping, solves joint torques $\boldsymbol{\tau}$ given a set of task constraints

$$\mathscr{H} : (\mathbf{c}, \mathbf{s}, t, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \longmapsto \boldsymbol{\tau} \tag{1}$$

where $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^m$ are the position, velocity, and acceleration of the robot's joints, and m is the number of joints and t the current time. $\mathbf{c}$ and $\mathbf{s}$ are reference trajectories for the CoM and the swing foot, respectively, additional details are given in Section III-A.

TABLE I
Optimization Variables and Descriptions

Optimization parameter	Description	Affects	Bounds	Optimized range	Unit
t_{min}	Minimum swing time	CoM	0.01–0.99	0.01–0.77	s
s_{max}	Maximum step length	CoM	0.01–0.99	0.06–0.98	m
t_{swing_start}	Swing foot start time	Swing Foot	0.01–0.08	0.026–0.078	s
s_{speed}	Swing foot velocity	Swing Foot	0.2–3.0	0.2–1.34	m/s

Trajectory planning generates spatial paths which are intended to be followed by a specific part of the robot, in our case, we have two trajectories, one for the CoM [36] and one for the swing foot [37], [38]. Similarly, trajectory optimization seeks to find the best spatial paths between two points according to some task or cost [39], which in this case is performed by BO. The trajectories are time indexed position references such that $\mathbf{c}(t) \in \mathbb{R}^3$ and $\mathbf{s}(t) \in \mathbb{R}^3$, generated by the parameterized functions

$$\mathscr{C}_{traj\_gen} : (t_{min}, s_{max}) \longmapsto \mathbf{c} \tag{2a}$$

$$\mathscr{S}_{traj\_gen} : (s_{des}, t_{swing\_start}, s_{speed}) \longmapsto \mathbf{s}. \tag{2b}$$

CoM trajectory generator $\mathscr{C}_{traj\_gen}$ takes minimum step time, t_{min}, and maximum step length, s_{max}, as arguments. Swing foot trajectory generator $\mathscr{S}_{traj\_gen}$ is a function of desired step position s_{des}, swing start time t_{swing_start}, and step speed s_{speed} parameters. Step speed s_{speed} is the swing foot target speed, typically reached at swing apex. Open parameters are tuned in Phase 1 of our pipeline, summarized in Table I, and are concatenated in vector $\mathbf{p}$ (see Fig. 1)

$$\mathbf{p} = [t_{min}, s_{max}, t_{swing\_start}, s_{speed}]^T. \tag{3}$$

Note that s_des is not an open parameter, it is provided as an initial condition (Phases 1 and 2), or generated from (7). The

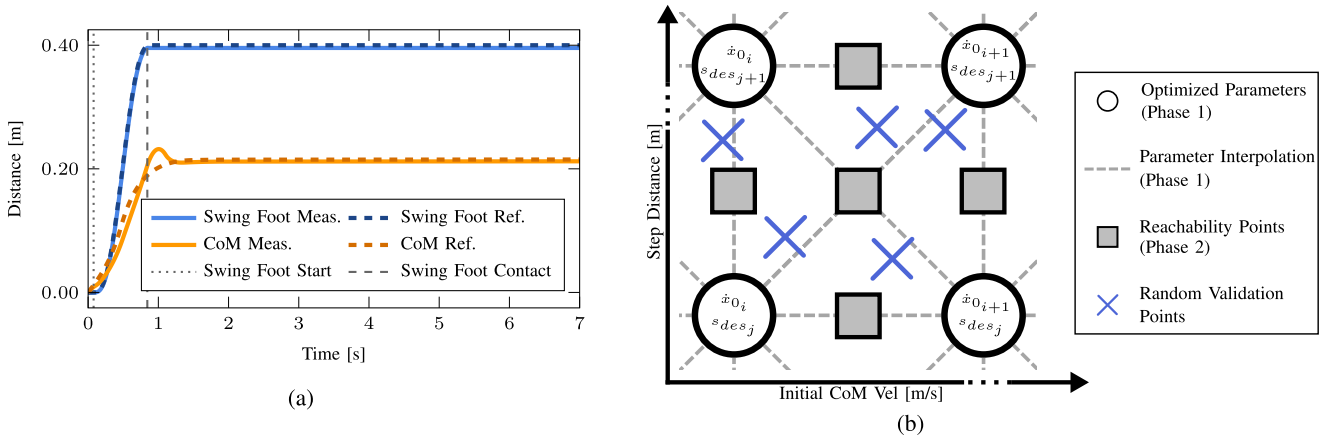


Fig. 4. Technical illustrations of pipeline show (a) example reference trajectories for the CoM and swing foot, and measured trajectories from the HQP. Initial conditions $\dot{x}_0 = 0.15$ m/s, $s_{des} = 0.4$ m. (b) Distribution of parameter, sampling, and validation points.

TABLE II
OBJECTIVE FUNCTION WEIGHTS AND THEIR VALUES

Notation	Affects	Value
w_f	Failure to complete stepping motion	0.001
w_{swing}	Swing foot position error	50
w_{x_mid}	Final CoM position	1
w_z	Final CoM height	1
w_τ	Torque minimization	0.0002

output of Phase 1 is function $\mathscr{G}$

$$\mathscr{G} : (\dot{x}_0, s_{des}) \longmapsto \mathbf{p}^* = \arg\max_{\mathbf{p}} \quad J(\dot{x}_0, s_{des}, \mathbf{p}) \tag{4}$$

which maps initial CoM velocity, $\dot{x}_0 \in \mathbb{R}^1$ in the lateral plane, and desired step position s_{des} to a set of optimal parameters $\mathbf{p}^*$ (optimal values are denoted by $(\cdot)^*$). In Phase 1, the values of s_{des} are set to predetermined intervals increasingly far away from the robot; in Phase 2, optimally efficient step positions s^*_{des} can be generated automatically.

To build this mapping, we use BO to maximize the objective function J (see (5)) for pairs of initial conditions. Initial conditions and BO parameters are iteratively passed to the trajectory generators (see (2a) and (2b)), produced trajectories are then tracked by the HQP (see (1)), in a full dynamic simulation environment, detailed in Section III-C, where the robot attempts to step toward the desired step position s_{des}. This is highlighted in dark blue in Fig. 3. The objective function J takes values from the dynamic simulation, which we consider to be a black-box and each term is assigned a manually tuned weight, each denoted by a subscript of w, defined in Table II

$$\begin{aligned} J(\dot{x}_0, s_{des}, \mathbf{p}) = -(&w_f(t_{total} - t_{term}) \\ &+ w_{swing}(s_{des} - s_{td})^2 + w_{x\_mid}(x_f - s_{mid})^2 \\ &+ w_z(z_{nom} - z_f) + w_\tau J_\tau(\dot{x}_0, s_{des}, \mathbf{p})). \end{aligned} \tag{5}$$

Weights are tuned to promote successful stepping, low energy cost, and scale down large terms. For example, t_{total} is set to 7 s at 1000 Hz, giving 7000 time-steps; termination at 4.5 s would incur a cost of $0.001 \cdot (7000 - 4500) = 2.5$, whereas other terms typically evaluate to less than one during success and are nominally high during failure to amplify w_f. Termination conditions are detailed in Section III-D.

The error between swing foot position at touchdown s_{td} and desired step location s_{des} incurs a cost proportional to the square of the error between them. To increase stability after the step, a cost is applied between final CoM position, x_f, and the midpoint between the stance foot and the desired swing foot position: $s_{mid} = s_{stance} + (s_{stance} - s_{des})$. To encourage straight legs after landing, a cost is applied to the final CoM height z_f to be as close as possible to a nominal height $z_{nom} = 0.925$ m. A term is added to minimize the integral sum of the squared measured torque $\boldsymbol{\tau}$ in all the robot's joints (32 in this case) between the time swing foot motion begins t_{lo} and touchdown time t_{td}

$$J_\tau : (\dot{x}_0, s_{des}, \mathbf{p}) \longmapsto \int_{t_{lo}}^{t_{td}} \boldsymbol{\tau}^2 \, dt. \tag{6}$$

After BO is completed for each pair of initial conditions, we are left with the mapping function $\mathscr{G}$ (see (4)), which outputs optimized parameters for pairs of initial conditions. In Phase 2, we query this mapping with a larger set of initial velocities and desired step positions to test how the optimized parameters generalize to novel initial condition pairs, illustrated in Fig. 4(b). Here, objective function J is no longer used, and we use the optimal parameters in the mapping function; instead, the dynamic simulation returns two values, shown in Fig. 3: a binary reachability value to encode whether motions are successful, and the integral sum of the square of the measured torque (see (6)) for every pair of initial conditions.

A forward map then encodes which step positions are reachable from each initial CoM velocity and another map quantifies the integral sum of the square torque cost during the swing phase of each reachable step position. Using the second map, we create an inverse mapping $\mathscr{K}$ from an initial CoM velocity $\dot{x}_0$ to an energy-optimal step position s^*_{des} and the optimal parameters

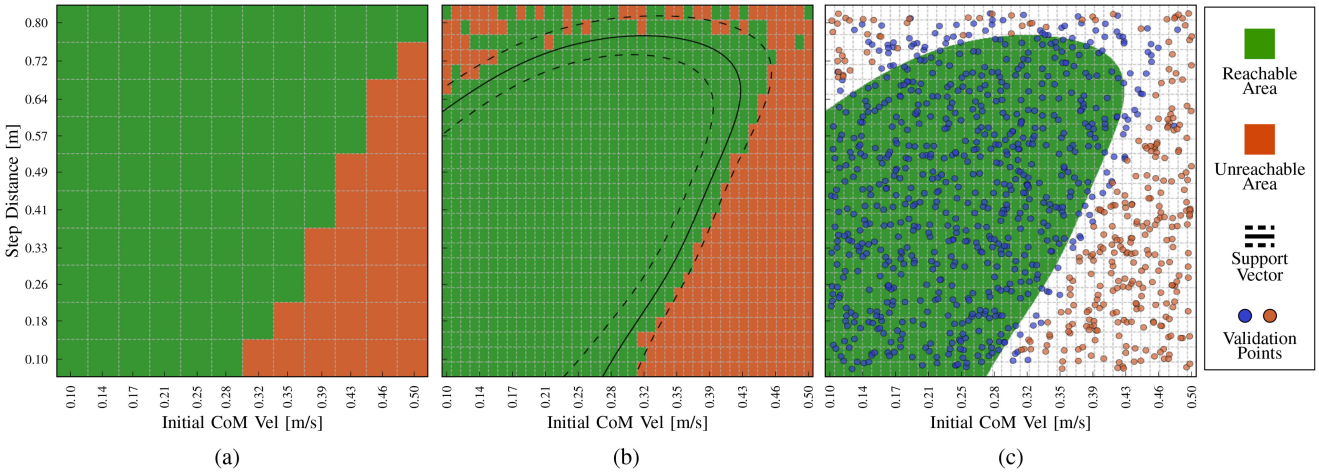


Fig. 5. Binary reachability maps show successful (green) or unsuccessful (red) steps for pairs of initial conditions. (a) Phase 1: Parameter optimization map. (b) Phase 2: Dense reachability map. (c) Validation: SVM generated reachability map with validation testing points.

used to reach that location

$$\mathscr{K} : (\dot{x}_0) \mapsto \begin{cases} \mathbf{p}^* = \mathscr{G}(\dot{x}_0, s^*_{des}) \\ \text{s.t.:} s^*_{des} = \arg\min_{s_{des}} J_\tau(\dot{x}_0, s_{des}, \mathscr{G}(\dot{x}_0, s_{des})). \end{cases} \tag{7}$$

Automatically generated energy-optimal step positions s^*_{des} are passed to $\mathscr{G}$ to reduce the input dimensionality such that only initial CoM velocity $\dot{x}_0$ is required to query optimal trajectory generation parameters $\mathbf{p}^*$.

All maps are built offline and a regression model is used to approximate energy-optimal step positions s^*_{des} for arbitrary initial CoM velocities $\dot{x}_0$. We can query these mappings for stepping or push recovery; we run validation tests using the mappings and previously unseen initial conditions.

III. Technical Details

A. HQP Structure

The HQP feedback controller is a function of forward dynamics and two parameterized trajectories (see (1)). We tuned the tasks, weights, and hierarchy order of an existing HQP controller [35]. Upper body joints such as the arms and head are not required for the stepping motion, so a posture task is also added to regulate arm motion and keep them near a nominal position with a low weight in the HQP.

B. Trajectory Generation

CoM and swing foot trajectories are passed to the HQP to determine their energy cost. Each has two open parameters which determine their profile and outputs time-indexed position references for the X-,Y- and Z-axes. Parameters determine the length of the trajectory and its gradient and are optimized in Phase 1 (see Section IV-A).

1) CoM Trajectory Generation: An existing LIPM-based model [36] produces CoM trajectories as a function of two parameters: minimum step time t_{min} and maximum step length s_{max}. An example trajectory is shown in Fig. 4(a). During testing, s_{max} was set to reflect the capabilities of the physical 32-DoF Talos humanoid robot, but this results in falling, so this parameter was added to the optimization. Since this pipeline is modular, this can be replaced with alternative CoM trajectory generation methods.

2) Swing Foot Trajectory Generation: Swing foot trajectories are fifth degree minimum-jerk polynomials [40] (see Fig. 4(a)), parameterized by the time at which the swing foot starts to move, (t_{swing_start}), and the swing foot speed (s_{speed}). The Z-axis consists of two minimum jerk trajectories connected at a via point at the maximum desired swing height z_{max}, for which we found 80 mm to be a reliable value.

C. Simulation Setup

For the dynamic simulation, we used Pinocchio rigid body dynamics library [41], based on the widely used dynamics equations in [42]. In the long term, we aim to move the query of mapping implementation onto the real robot, but was not possible for this article. This pipeline is compatible with any humanoid robot, here we used the Talos humanoid robot model (32 DoF), with the complete dynamic and kinematic properties of the real robot including the position, velocity, acceleration, and torque limits. The simulation environment was fully dynamic, including friction, torque limits, with both simulation frequency and control frequency of 1000 Hz ($\Delta t = 0.001$ s). The `BayesianOptimization` package [43] was used for parameter optimization.

D. Experimental Setup

We constrain motions for this proof-of-concept to gait initiation in the X-axis (forward), where in each simulation episode the robot starts in a standing configuration with the swing foot 1 cm above the ground to reduce the complexity of optimizing weight transfer time while we develop the pipeline.

Initial CoM velocities are induced by directly applying joint torques in simulation, torque values are calculated using the Jacobian from the stance foot to the CoM. On the real robot, we expect CoM velocities to be applied by having the robot lean in a

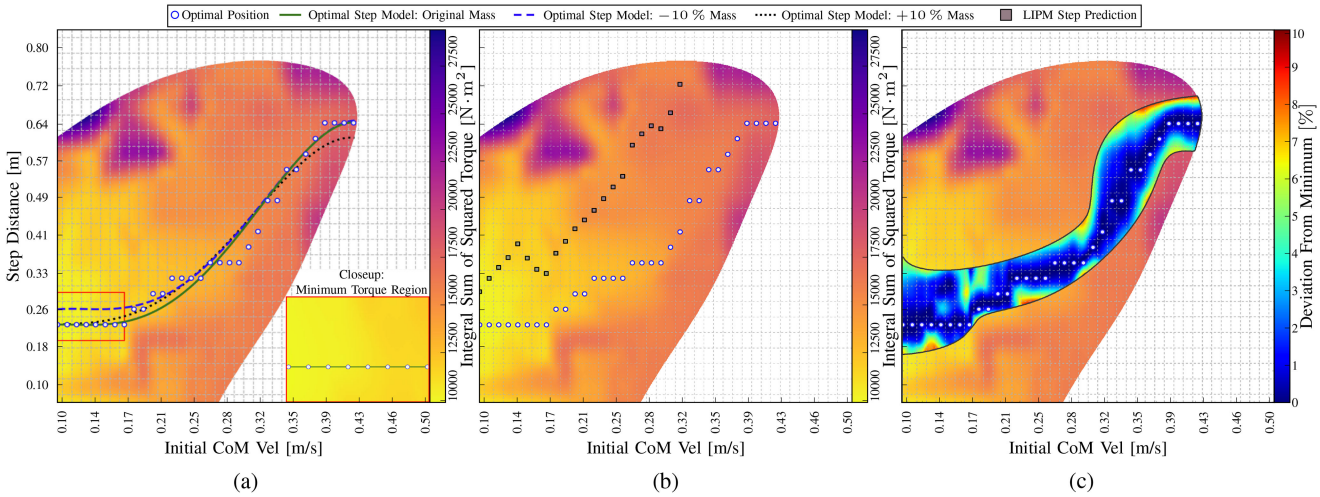


Fig. 6. Maps of the measured integral sum of squared joint torques during the swing phase to reach each position, with optimal step locations marked. (a) Regression model is fitted to optimal step positions; a region with maximum 5% deviation from optimal is shown in the inset. (b) Energy optimal step positions are compared to predictions from the LIPM model. (c) Map showing step locations which deviate from the optimal by a maximum 10% for the same initial CoM velocity superimposed onto 6(a).

Algorithm 1: Bayesian Parameter Optimization.

input : List of sampled CoM velocities:$\dot{\mathbf{X}}_0$,
List of desired step positions:$\mathbf{S}_{des}$
output: Mapping function $\mathscr{G}$ (Eq. 4)

```
for each CoM initial velocity (ẋ_0) in Ẋ_0 do
    for each step position (s_des) in S_des do
        for each k in BayesOptIterations do
            p(k) ← BayesianOptimization (Eq. 5)
            objective ← DynamicSim(ẋ_0, s_des, p(k))
        end
        G(ẋ_0, s_des) ← arg max_p (objective)
    end
end
```

given direction or being pushed. Termination conditions during optimization are as follows: robot reaches desired foot position and remains standing at $t = t_{total}$ (success) or the norm of joint velocities exceed a threshold ($1e6$) (failure). If $t_f < t_{total}$, the remaining sensor readings are filled with a nominal high value.

IV. Building Reachability Map: Dynamic Stepping Optimization Pipeline

A. Parameter Optimization: Phase 1

1) Bayesian Optimization: Phase 1 of our pipeline, shown in the dark blue box in Fig. 3, optimizes a set trajectory generation parameters, $\mathbf{p}$, for pairs of initial conditions. Algorithm 1 shows the nested loops: for every initial CoM velocity $\dot{x}_0 \in \dot{\mathbf{X}}_0$ and every desired step position $s_{des} \in \mathbf{S}_{des}$, a set of parameters $\mathbf{p}$ is optimized via BO, such that for each pair, $[\dot{x}_0, s_{des}]$, is mapped to a set of parameters and stored in the mapping $\mathscr{G}$.

Optimal parameters were found for 150 initial conditions: 15 initial CoM velocities, 10 step positions—sparsely covering the state space, as shown in Fig. 4(b)—with 170 BO iterations for each (100 random, 70 Bayesian) using objective (5). Fig. 5(a) shows the result, where initial conditions are projected onto a 2-D space and the area around each is colored according to a binary classification of whether successful stepping parameters were found. All desired step locations are reachable from low initial CoM velocity and as this increases, steps which are closer to the robot's starting position are no longer reachable, as the robot is moving too quickly to take a single short step in that direction.

2) Parameter Interpolation: Linear, element-wise interpolation between the nearest four sets of optimized connected the sparsely optimized points (see Fig. 4(b)) to create a continuous mapping $\mathscr{G}$ (4) from arbitrary initial values to a set of interpolated, optimized parameters $\mathbf{p}^*$.

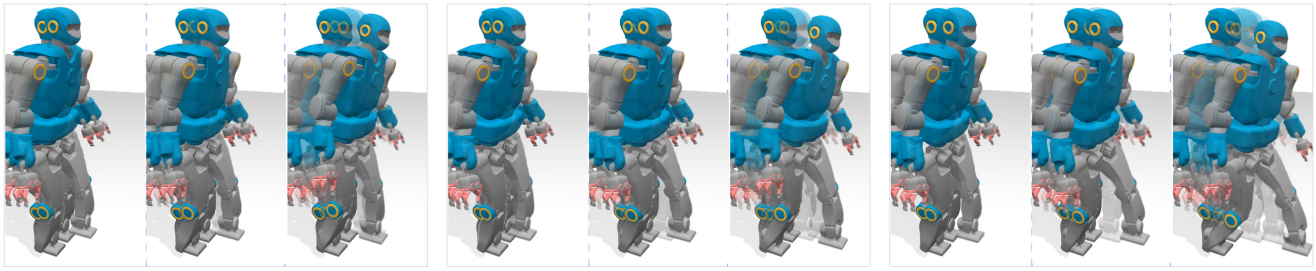
B. Reachability Maps: Phase 2

The purpose of Phase 2 (orange box, Fig. 3) is to use the optimized and interpolated parameters from Phase 1 to create higher resolution maps to denote the reachability and efficiency stepping parameters, illustrated in Fig. 4(b); no further parameter optimization is performed.

The outputs of this Phase are—a high resolution forward map of reachable step positions, a map of the integral sum of square torque cost to reaching these positions using the simplified model trajectories, and an inverse mapping for selecting optimally efficient step positions from any initial condition (see (7)).

We used 1000 initial condition pairs (40 initial CoM velocities, 25 desired step positions) to sample the mapping function, compared to 150 in Phase 1. Initial velocities were between $\dot{\mathbf{X}}_0 = [0.1 \text{ m/s}, \ldots, 0.5 \text{ m/s}]$ at 0.017 m/s intervals and desired step positions: $\mathbf{S}_{des} = [0.1 \text{ m}, \ldots, 0.8 \text{ m}]$ at 0.029 m intervals. A dynamic simulation episode was executed for each initial condition pair using parameters $\mathbf{p}^* = \mathscr{G}(\dot{x}_{0_i}, s_{des_j}) \quad \forall\, \dot{x}_{0_i} \in \dot{\mathbf{X}}_0, s_{des_j} \in \mathbf{S}_{des}$ (4).

Simulation episodes return two values—a binary value to denote if the stepping motion was successful, and the integral sum of square joint torques, $\boldsymbol{\tau}$, using (6). As a result, each pair of initial conditions has an associated reachability value and



(a) (b) (c)

Fig. 7. Validation of stepping motions using our automatic step selection validation. Initial CoM velocities 0.164, 0.292, 0.398 $\mathrm{m/s}$, were mapped to step positions: 0.248, 0.409, 0.617 m for (a), (b), and (c), respectively.

measured integral sum of square joint torque value that we use to build the reachability maps.

1) Binary Reachability Map: The binary reachability map (see Fig. 5(b)) projects every sampled pair of initial conditions from Phase 2 onto a 2-D space and each pair is colored according to a binary classification of if the robot successfully reached the desired position and stayed standing or not. Similar to Fig. 5(a), when using the interpolated parameters many steps are successful at lower initial CoM velocities and as this increases the step locations closer to the robot are unreachable. However, the space of parameters that lead to successful motions appears to be much smaller for longer steps, but this is not captured by the interpolation between sparsely sampled points, resulting in noisy regions at the top of the map. In these areas where success is uncertain, the robot may not fall in every case, and it may be possible to take extra steps, but here we consider one-step capturability to be unsuccessful in the noisy region.

A denser set of optimized parameters would remove this noise but would require more optimization time and since the robot is operating at the limits of its workspace, would still be sensitive to minor changes in modeling or sensor error on the real robot. Instead, we chose a more conservative approach to trim the noisy extremes and leave only the conditions which reliably lead to successful stepping. We train a support vector machine (SVM) model to separate the reliable step locations from the noisy areas, shown in Fig. 5(b). By querying the SVM model, we create a cleaner representation of the safe stepping area (see Fig. 5(c)). We used a third-order SVM with a radial basis function kernel, reachable points had a weight of one, unreachable points had a weight of 14.

2) Measured Torque Maps: The integral sum of square torque measured during the swing phase of motions from each initial condition pair is shown in Fig. 6(a), where each point in the trimmed reachability map is colored according to the integral sum of the squared joint torques (see (6)); darker (more purple) colors denote higher measured integral sum of square joint torque and the lowest cost step positions for each sampled initial velocity are marked. The distribution of torque patterns is highly nonlinear, but steps with minimum measured integral sum of square joint torques form a simple trend that can be used for footstep prediction. We consider the energy-optimal stepping positions to be those with the lowest integral sum of squared torque integral for all joints.

3) Minimum Energy Step Selection: The relationship between initial CoM velocity and energy-optimal step positions can be modeled using a simple fourth-order polynomial regression, which can quickly approximate energy-optimal step positions given an initial CoM velocity (see (7)). Fig. 6(a) shows this model captures the minimal energy step positions trends with a mean error of 216.12 $\mathrm{N \cdot m^2}$ (StD.$=287.84$ $\mathrm{N \cdot m^2}$, Min.$=0$ $\mathrm{N \cdot m^2}$, Max.$=1159.39$ $\mathrm{N \cdot m^2}$).

C. Query of Mapping

Resultant mappings can be queried to produce trajectories for stepping motions, shown in the green box in Fig. 3. Therefore, for a given CoM velocity, an energy-optimal step position is output by the mapping $s^*_{des} = \mathscr{K}(\dot{x}_0)$ (see (7)), which is then used to generate the stepping parameters to produce this motion using the mapping $\mathbf{p}^* = \mathscr{G}(\dot{x}_0, s^*_{des})$ (see (4)). This allows us to automatically generate stepping for a given initial CoM velocity during push motions.

Since the parameters in Phase 1 are optimized for a set of discrete points, in the next section, we run a series of validation tests to show that motions are still produced reliably given continuous initial conditions. In these validation tests, we test the mapping function $\mathscr{G}$ under a range of initial conditions and the mapping $\mathscr{K}$ to test the energy optimal step selection. In each case, initial conditions passed to one or both of the mapping functions and their outputs are used to run an episode of the dynamic simulation.

V. Results

A. Reachability Map Validation

Validation was run for two purposes: first, to verify the interpolation of parameters at random, novel points using the original robot model (see Fig. 4(b)); second is to induce simulated discrepancies to the robot model to validate the robustness of the parameters to modeling disturbances.

In each case, 1000 random pairs of initial conditions, $[\dot{x}_0, s_{des}]$, were generated which were *not* the same as those tested in the previous phases over the entire initial search space. For each pair, an episode of dynamic simulation was run using parameters from the mapping $\mathscr{G}$ and episodes were judged successful (colored blue) or unsuccessful (colored red) according to the termination conditions in Section III-D.

1) Interpolated Parameter Validation: This validation uses the original robot model and tests the interpolated region between optimized points to verify that this region is in fact reliable

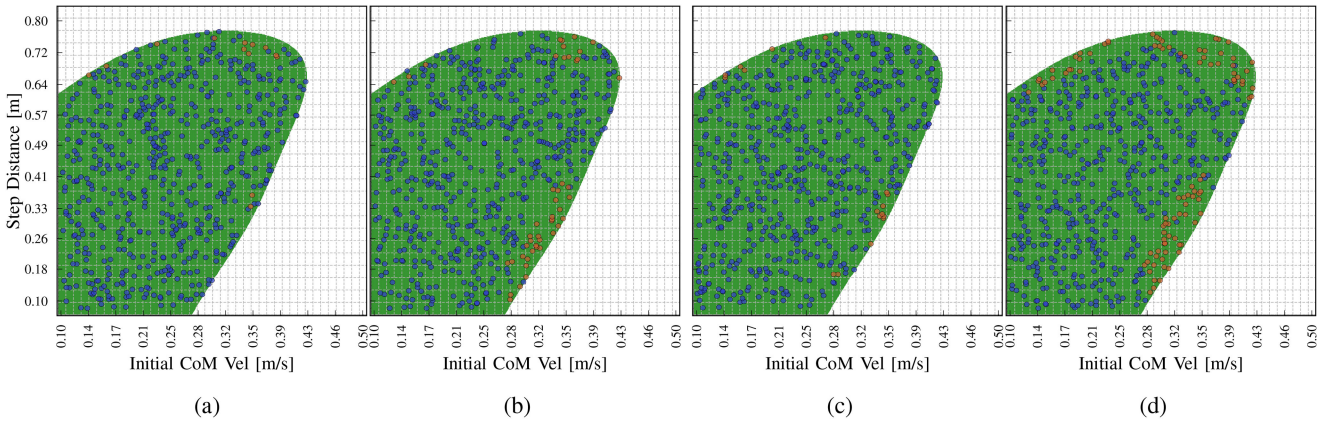


Fig. 8. Validation testing for mass discrepancies (a) +13.1% mass, (b) +18.5% mass and feedback delay, (c) 20 ms, and (d) 30 ms. Only internal points are shown for clarity.

for the robot used during optimization. Fig. 5(c) shows the random initial conditions projected onto the trimmed reachability map, highlighting the noisy, uncertain regions for longer step distances and showing that the safe region is conservative given the region of successful trials surrounding this area.

2) Modeling Error Validation: Next, we introduce modeling errors to validate the robustness of the parameters to dynamic discrepancies. Random values were generated between $\pm 20\%$ of the original robot mass and the robot model was adjusted according to these values, with no changes to the HQP feedback gains and for each mass variant, 1000 new random pairs of initial conditions were generated and tested.

Fig. 8(a) and (b) shows representative validation trials projected onto the safe reachability region for the original mass robot. Full plots are shown in the Supplementary Material (Figs. S1 and S2) and omitted here for clarity. In all test cases where the mass is lower than the original robot ($-5.8\%, -12.9\%, -17.3\%$), an increasing number of trials outside the original safe region were successful, making it appear more conservative. Increasing robot mass 6.2%, 13.1%, 18.5%, (Fig. 8(a) and (b) show 13.1%, 18.5%, respectively), causes failures to increase proportionally to the mass such that a more conservative safe region would be needed for these variations under the current settings. Given the large mass variation without changing the HQP feedback gains, this shows that even under modeling error, the optimized parameters still lead to successful stepping in a majority of cases.

3) Feedback Delay Validation: Control loop feedback delays were also added to validate parameter applicability to real robot hardware. Delays were simulated by providing the HQP with noncurrent feedback about the robot state, resulting in modeling errors. The robot starts each simulation in motion, so feedback from t_0 was given until the current timestep was greater than the delay being modeled.

Using the original robot mass and HQP parameters, 10, 20, and 30 ms feedback delays were tested, each at 1000 random initial conditions throughout the state space. Fig. 8(c) and (d) shows successful and unsuccessful validation points within the original reachability area for the 20 and 30 ms delay cases. Validation points outside the reachable area are omitted here for clarity, but shown in full in Supplementary Material (see Fig. S3), along with the 10 ms delay case.

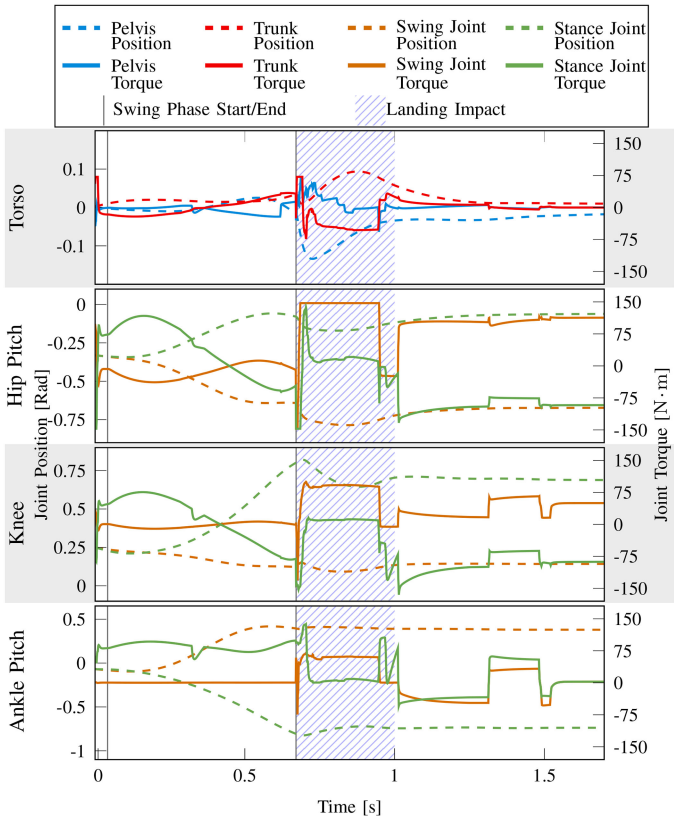


Fig. 9. Joint profiles in the torso and legs during one validation stepping motion in Fig. 7(c).

Unsurprisingly, the 30 ms delay (see Fig. 8(d)) causes the most errors in the reachable area. In the lower right, total step time is barely higher than the 30 ms delay, so the robot falls before the HQP can react. The upper edge shows the delay causes failures for longer steps, as the body is accelerating throughout the swing motion, leading to inaccurate touchdown estimation. Feedback delays are unavoidable on the real robot due to communication latency and filtering, but these results show that the reachability map can still be applied to these cases if they are adjusted to be more conservative to account for this.

B. Energy-Optimal Step Selection Validation

Energy optimal step selection was validated with 150 initial CoM velocities, randomly generated within the safe region (0.1:0.43 m/s). A dynamic simulation episode was executed for each point, using an energy-optimal step position s^*_{des}. All trials were successful as shown in Fig. 7 and the attached video.

Fig. 9 shows the measured joint positions and torques of the torso (trunk and pelvis), hip pitch, knee, and ankle pitch joints in both legs during the validation shown in Fig. 7(c). All joints are included in the motion analysis, but the effect of arm and head joints are negligible so it is not analyzed further. During the swing phase, swing leg joint torque is lower than in the stance leg which serves as a rigid pivot. The landing impact is clear in the measured torque (see Fig. 9 hatched area) and is close to the actuation limits which helps explain noisy regions in Fig. 5(b).

Validation experiments were carried out to investigate how changes in robot mass affect optimal step positions. Phase 2 of the pipeline was run with a range of mass variations using the optimal output from Phase 1 which is based on the original robot mass. Fig. 6(a) shows regression models $+10\%$ (blue) and -10% (black) variation alongside that of the original robot mass (green). Optimized parameters, interpolation, and HQP feedback gains as tuned for the original robot mass were used.

Negative mass variations result in optimal step positions at low initial CoM velocities being further from the robot and being steady for longer than for the original mass, then converging to the same location as initial CoM velocities increase. The variations by increasing the mass result in divergences away from the optimal step locations, which initially match those of the original mass, at around $0.14\ \mathrm{m/s}$, and diverge further from the original optimal as initial CoM velocity increases. These tests demonstrate that parameters are robust to modeling errors such as robot mass, and there is an underlying trend to the dynamics of a multibody system such that the optimal step position can be calculated given the construction and dynamics of a robot.

C. LIPM Comparison

We compared one-step-capturable step positions predicted by the LIPM model given an initial CoM velocity to the energy-optimal step positions from our optimization, as shown in Fig. 6(b). To do this, we queried the mapping $\mathscr{G}$ at the optimal step locations for each sampled initial CoM velocity, forward simulated the LIPM using these parameters and recorded the position of the capture point at the end of the swing time. Final capture points were projected onto the measured integral sum of squared joint torque map to extract the torque cost of moving to that location, we compared this to the measured integral sum of square joint torque of the optimal step location.

After $0.32\ \mathrm{m/s}$, the LIPM predicted step location is beyond the reachable area for the robot in our simulations, so are not included since the torque cannot be measured. We calculated the error in the integral sum of square joint torque between the optimal step positions from our optimization and those from the LIPM prediction, with a root-mean-squared error of 1364.5 $\mathrm{N \cdot m^2}$ (10.9% increase) (StD. $=$ 829.6 $\mathrm{N \cdot m^2}$(6.0%), Min. $=$ 164.3 $\mathrm{N \cdot m^2}$(2.1% increase), Max. $=$ 2531.9 $\mathrm{N \cdot m^2}$(20.0% increase). Comparison of preferred step positions with the LIPM shows the effect of modeling discrepancies between using the full dynamic model and the simplified model. Each LIPM step location in Fig. 6(b) uses the same step parameters as the optimized step location below it which uses our full-body model. This demonstrates the extent to which optimality planned during LIPM motion planning is lost when tracking the trajectories to the full-body model. Moreover, using the LIPM to predict step locations reduces the potential workspace of the robot, potentially harming the one-step recovery ability of the robot.

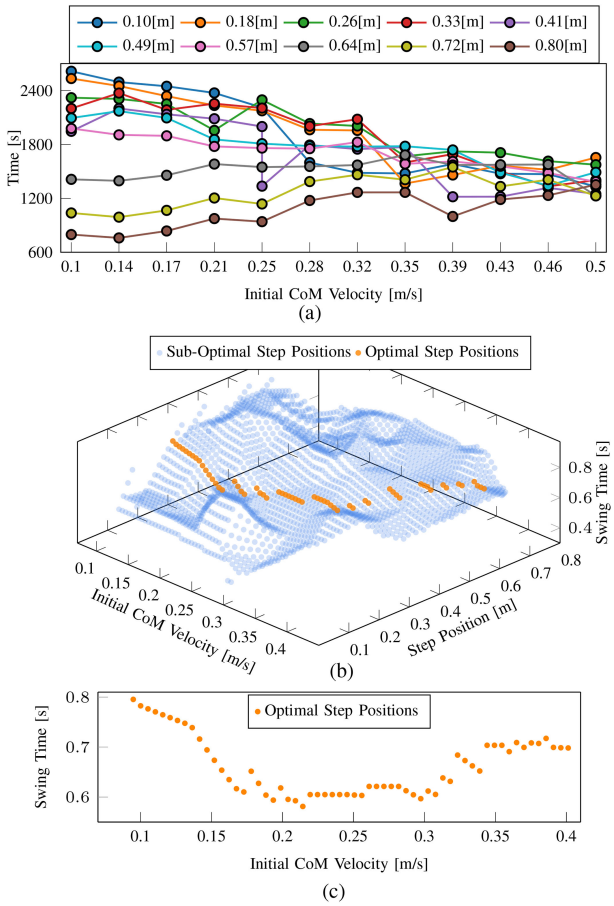


Fig. 10. Analysis of pipeline performance and parameter results. (a) Computational time to optimize parameters for each pair of initial conditions, step positions denoted by colored traces. (b) Three-dimensional plot of swing time for optimal and suboptimal points. (c) Two-dimensional projection of swing time for optimal points.

D. Diversity in Efficient Step Selection

In addition to the simple trend of energy-optimal step positions, a diverse range of near-optimal step locations are present on the map. The highlighted area in Fig. 6(a) shows step positions where energy costs are a maximum of 5% above optimal for the same initial condition and Fig. 6(c) shows a band of step positions superimposed onto Fig. 6(a) which deviates from the optimal to a maximum of 10%, as a result, any step positions in this range we define near-optimal.

Near-optimal regions form simple heuristics: for low initial CoM velocities, stepping between 22 and –30 cm will result in minimal impact on energy-efficiency, even if the optimal position is not reached, allowing coarse, yet rapid step selection with trivial changes in energy-efficiency. Additionally, since the regions span a range of initial CoM velocities, inaccurate CoM state estimation can still lead to efficient stepping.

E. Underlying Energy Optimality

Near-optimal efficiency regions suggest an underlying optimality in dynamic stepping gives us insight into human step selection. If similar regions exist in dynamic multibody systems in general, humans can learn similar heuristics and use them for simple, rapid, near-optimal step selection. Having near-optimal heuristics for step selection that are robust to sensing delays and inaccuracies would be beneficial to developing biological humans.

Fig. 2(b) shows a clustering of selected steps similar to that in the highlighted area in Fig. 6(a), but with a higher range of initial CoM velocities. Humans can withstand higher magnitudes using foot tilting behavior. This is a limitation of our robot control, which does not consider underactuated foot tilting control, hindering the range of feasible step positions.

We also gain insight into energy optimality by looking at the optimized step parameters. Total swing time can be calculated from the optimal parameters ($t_{swing\_start} + s_{des}/s_{speed}$), and plotted in Fig. 10(b), showing all initial condition pairs and their optimized swing time, with optimal points projected into 2-D in Fig. 10(c), from the initial foot position $[0, 0, 1]cm$, swing time is dependent on the desired step position, displayed on the vertical axis. This shows a piece-wise relationship between initial CoM velocity and optimal swing time, where swing time initially drops, levels off, then rises as the initial CoM velocity increases. The majority of the swing time effect is caused by the swing speed parameter s_{speed} with only minor changes induced by the swing start time t_{swing_start} as shown by the optimal ranges in Table I.

F. Analysis of Computation Time

Training used an Intel Core i7-8700 k with 12 cores (6 physical), 32 GB RAM, Ubuntu 16.04, and Pinocchio 2.5.0. The pipeline is parallelized, with each core optimizing all step positions for one initial CoM velocity, and took around 5 h, reachability map building takes around 40 min. Parameters for one swing foot were used for the opposite foot.

Fig. 10(a) shows the computation time for parameter optimization for each pair of initial conditions, with 170 episodes for each pair. For larger initial CoM velocities and larger step distances, the number of early terminations leads to quicker computation. Parallelization scales linearly with the number of cores, where large-scale distribution, with one core per initial condition pair, would lead to 45 $\min$ optimization.

Swing foot trajectories are generated in 0.5 ms, and CoM trajectories in 1.4 s, due to the nonlinear optimization formation used for generation, but can be reduced to 0.5 ms using the same generation as the swing foot. Querying the energy-optimal step selection takes 0.13 ms.

VI. Discussion

The study of energy-efficient locomotion is a complex process that involves the whole-body dynamics and control together as a whole and requires complex optimization, as well as a global search of global energy-optimal step location and timing, which are all highly nonlinear. The complex interplay between multibody dynamics, control, and gait parameters is shown by the optimal gait parameters in Fig. 10(b), where this nonlinear relationship and nonsmooth gradient would cause standard gradient search methods to get stuck in the local minima. However, the mapping between initial conditions and key gait parameters, such as step location and swing time, suggests that the gait parameters can be represented by piecewise approximations, as shown in Fig. 10(b) and (c), resulting in a method for rapid step selection. This indicates that despite the complexity of the whole process, this nonlinear relationship in human gait can be possibly learned by humans by prior trials and experience.

Additionally, balance can be recovered using similar step locations and swing times for different initial CoM velocities by trading off energy optimality, which can potentially explain the large variations in step location in human study [8].

VII. Conclusion

In this article, we investigated energy-efficient step selection using nonlinear optimization to build offline reachability maps. We identified that selecting energy-efficient steps during push recovery or finding a set of diverse stepping regions are difficult to characterize with simple models and are hard to compute online (see Fig. 10(a)). Hence, reachability maps can be used for rapid step selection on flat ground. Results also give us insight into the possibility and feasibility of diverse step selection for humanoids. In our future work, we plan to extend this pipeline to study energy-efficient locomotion in different modes and implement the query of mapping for warm-start solution for online optimization.

References

[1] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Proc. IEEE/RAS Int. Conf. Humanoid Robots*, 2014, pp. 295–302.

[2] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-body dynamic behavior and control of human-like robots," *Int. J. Humanoid Robot.*, vol. 1, no. 1, pp. 29–43, 2004.

[3] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE/RAS Int. Conf. Humanoid Robots*, 2014, pp. 279–286.

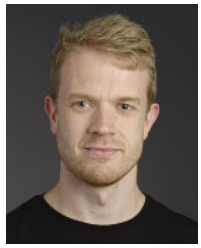
[4] Y.-D. Hong and B. Lee, "Real-time feasible footstep planning for bipedal robots in three-dimensional environments using particle swarm optimization," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 1, pp. 429–437, Feb. 2020.

[5] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse, and E. Yoshida, "Real-time replanning using 3D environment for humanoid robot," in *Proc. IEEE/RAS Int. Conf. Humanoid Robots*, 2011, pp. 584–589.

[6] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots.*, 2016, pp. 842–849.

[7] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taïx, and A. Del Prete, "SL1M: Sparse l1-norm minimization for contact planning on uneven terrain," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 6604–6610.

[8] C. McGreavy *et al.*, "Unified push recovery fundamentals: Inspiration from human study," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 10 876–10 882.
[9] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, vol. 1, pp. 239–246.
[10] J. A. Castano, Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, "Dynamic and reactive walking for humanoid robots based on foot placement control," *Int. J. Humanoid Robot.*, vol. 13, no. 2, 2016, Art. no. 1550041.
[11] Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, "Compliance control for stabilizing the humanoid on the changing slope based on terrain inclination estimation," *Auton. Robots*, vol. 40, no. 6, pp. 955–971, 2016.
[12] J. Ding, C. Zhou, and X. Xiao, "Energy-efficient bipedal gait pattern generation via com acceleration optimization," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, 2018, pp. 238–244.
[13] V. A. Tucker, "The energetic cost of moving about: Walking and running are extremely inefficient forms of locomotion. Much greater efficiency is achieved by birds, fish-and bicyclists," *Amer. Scientist*, vol. 63, no. 4, pp. 413–419, 1975.
[14] E. Ambrose, W.-L. Ma, C. Hubicki, and A. D. Ames, "Toward benchmarking locomotion economy across design configurations on the modular robot: AMBER-3M," in *Proc. IEEE Conf. Control Technol. Appl.*, 2017, pp. 1270–1276.
[15] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1140–1154, 2016.
[16] M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Optimizing energy consumption and preventing slips at the footstep planning level," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*., 2015, pp. 1–7.
[17] S. Seok *et al.*, "Design principles for energy-efficient legged locomotion and implementation on the MIT cheetah robot," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1117–1129, Jun. 2015.
[18] Z. Li, N. G. Tsagarikis, D. G. Caldwell, and B. Vanderborght, "Trajectory generation of straightened knee walking for humanoid robot iCub," in *Proc. 11th Int. Conf. Control Autom. Robot. Vis.*, 2010, pp. 2355–2360.
[19] M. S. Erden and K. Leblebicioglu, "Torque distribution in a six-legged robot," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 179–186, Feb. 2007.
[20] J. E. Bobrow, B. Martin, G. Sohl, E. Wang, F. C. Park, and J. Kim, "Optimal robot motions for physical criteria," *J. Robot. Syst.*, vol. 18, no. 12, pp. 785–795, 2001.
[21] D. P. Garg and M. Kumar, "Optimization techniques applied to multiple manipulators for path planning and torque minimization," *Eng. Appl. Artif. Intell.*, vol. 15, no. 3/4, pp. 241–252, 2002.
[22] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, "Fast humanoid robot collision-free footstep planning using swept volume approximations," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 427–439, Apr. 2012.
[23] M. Fallon *et al.*, "An architecture for online affordance-based perception and whole-body planning," *J. Field Robot.*, vol. 32, no. 2, pp. 229–254, 2015.
[24] M. Murooka, I. Kumagai, M. Morisawa, F. Kanehiro, and A. Kheddar, "Humanoid loco-manipulation planning based on graph search and reachability maps," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1840–1847, Apr. 2021.
[25] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (Cat. No. 01CH37180)*, 2001, pp. 500–505.
[26] M. Kallman and M. Mataric, "Motion planning using dynamic roadmaps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 4399–4404.
[27] S. J. Jorgensen, M. Vedantam, R. Gupta, H. Cappel, and L. Sentis, "Finding locomanipulation plans quickly in the locomotion constrained manifold," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 6611–6617.
[28] F. Burget and M. Bennewitz, "Stance selection for humanoid grasping tasks by inverse reachability maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5669–5674.
[29] Y. Yang, W. Merkt, H. Ferrolho, V. Ivan, and S. Vijayakumar, "Efficient humanoid motion planning on uneven terrain using paired forward-inverse dynamic reachability maps," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2279–2286, Oct. 2017.
[30] P. Fernbach, S. Tonneau, and M. Taïx, "Croc: Convex resolution of centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.
[31] Y.-C. Lin, L. Righetti, and D. Berenson, "Robust humanoid contact planning with learned zero-and one-step capturability prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2451–2458, Apr. 2020.
[32] L. Kaul and T. Asfour, "Human push-recovery: Strategy selection based on push intensity estimation," in *Proc. 47st Int. Symp. Robot.*, 2016, pp. 1–8.
[33] R. Kopitzsch, D. Clever, and K. Mombaur, "Optimization-based analysis of push recovery during walking motions to support the design of rigid and compliant lower limb exoskeletons," *Adv. Robot.*, vol. 31, no. 22, pp. 1238–1252, 2017.
[34] K. Yuan, I. Chatzinikolaidis, and Z. Li, "Bayesian optimization for whole-body control of high-degree-of-freedom robots through reduction of dimensionality," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2268–2275, Jul. 2019.
[35] A. Del Prete, N. Mansard, O. E. Ramos, O. Stasse, and F. Nori, "Implementing torque control with high-ratio gear boxes and without joint-torque sensors," *Int. J. Humanoid Robot.*, vol. 13, no. 1, 2016, Art. no. 1550044.
[36] W. Hu, I. Chatzinikolaidis, K. Yuan, and Z. Li, "Comparison study of nonlinear optimization of step durations and foot placement for dynamic walking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 433–439.
[37] J. Englsberger *et al.*, "Overview of the torque-controlled humanoid robot Toro," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*., 2014, pp. 916–923.
[38] C. Zhou, X. Wang, Z. Li, and N. Tsagarakis, "Overview of gait synthesis for the humanoid coman," *J. Bionic Eng.*, vol. 14, no. 1, pp. 15–25, 2017.
[39] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.
[40] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, 1985.
[41] J. Carpentier *et al.*, "The Pinocchio c library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *Proc. IEEE/SICE Int. Symp. Syst. Integration*, 2019, pp. 614–619.
[42] R. Featherstone, *Rigid Body Dynamics Algorithms*. Berlin, Germany: Springer, 2014.
[43] F. Nogueira, "Bayesian optimization: Open source constrained global optimization tool for python," 2014. [Online]. Available: https://github.com/fmfn/BayesianOptimization



Christopher McGreavy received the M.Sc. degree in computational neuroscience and cognitive robotics from the University of Birmingham, Birmingham, U.K., and the M.Sc.(R) degree in robotics and autonomous systems in 2018 from the University of Edinburgh, Edinburgh, U.K., where he is currently working toward the Ph.D degree in robotics and autonomous systems.

His research interests include robotic and human locomotion, enhancing understanding, and application of dynamic and efficient legged locomotion.

Zhibin (Alex) Li received the joint Ph.D. degree in robotics from the Italian Institute of Technology, Genova, Italy and the University of Genova, Genova, Italy, in 2012.

He is currently an Associate Professor with the Department of Computer Science, University College London, London, U.K. His research interests include inventing new control, optimization, and deep learning technologies, and creating intelligent behaviors of dynamical systems with human comparable abilities to move and manipulate.