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 robot dynamics and whole-body optimal control as a coupled system to investigate energy consumption during balance recovery. We developed a 2-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. Despite the complexity of the nonlinear process, we show that near-minimal effort stepping locations fall within a region of attractions, rather than a narrow solution space suggested by a simple model. This provides new insights into the non-uniqueness of near-optimal solutions in robot motion planning and control, and the diversity of stepping behavior in humans.

I. INTRODUCTION

To achieve 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 multi-body dynamics and its control, which therefore must be considered together when studying energy consumption. For high-DoF robots, this hardware and software coupling makes studying energy consumption for general locomotion extremely complex and computationally heavy, given the high dimensionality of the state-action and the solution spaces. In this paper, we study a subset of this problem as a proof of concept to investigate stepping behavior and energy consumption, and shed light on this nonlinear relationship.

Even given a reduced complexity of locomotion, such as stepping and balance recovery, the coupling between full-body dynamics and control demands high computation cost due to the large-scale optimization in high-dimensional space. This is the rationale for using simplified models, such as the Linear Inverted Pendulum Model (LIPM) [1] for simple step planning [2], to reduce the computation for locomotion planning, but they do not capture the complex dynamics of all the robot’s links and joints. Therefore, optimal motions based on reduced models are no longer optimal in terms of energy efficiency when considering the whole-body dynamics of a robot.

Due to the disconnect between planning and control, motion plans projected from reduced models onto the whole-body of the robot lose the real optimality of the plan. Moreover, strong nonlinear and coupling effects of dynamics cause deviation between simple model plans and the control during execution, and hence cause errors between predicted and actual energy cost. Therefore, planning and control must be considered together when studying energy consumption, since real optimality depends on the accuracy of control execution given a robot system, making the type of control and its performance an inseparable part of the whole system’s energy consumption.

To plan and control the whole robot, Whole Body Motion Planning (WBM) [4] and Whole Body Control (WBC) [5] can be used for constrained planning and control to enforce tasks, such as: kinematic reachability [6, 7], collision avoidance [8] and realistic centroidal dynamics [9, 10]. Computing WBM plans using Mixed Integer Convex Optimization were studied in [11] [6], which is intended for online use and has no mechanism for comparing all possibilities and choose energy optimal gaits. Therefore, energy consumption during stepping can be studied using whole-body methods if we expand the search to all feasible gait parameters such as step locations and timings. However, searching the whole solution space is complex and computationally heavy, therefore the study of energy consumption given open gait parameters are investigated by pre-computation and offline search.

In our problem formulation, stepping motions that are pre-computed are used to produce volumetric reachability maps of footstep locations subject to sets of constraints. Previous work produced maps with constraints such as: kinematic...
reachability [12, 13], feasible transitions motions [14, 15] and obstacle avoidance [8, 16]. Reachability maps can then be used for faster locomotion planning [17, 18], complex end-
pose planning [19], enabling dynamic transitions [20] and for learning [21]. In this paper, we study reachability maps that encode energy cost for reaching step positions across the whole state space and build simple heuristics to capture a diverse range of efficient stepping motions that can be used to plan motions online with little computational cost.

Though heuristics for energy efficient step regions are nonlinear, they can be learned by humans to achieve complex stepping behaviors. Figure 2b shows step selection during human push recovery for initial Center of Mass (CoM) push velocities from the work in [3], which are offset by mean initial velocity of non-stepping trials (0.1103 m/s) and show the stochasticity of human stepping [22, 23]. In this study, our findings on the underlying optimality of stepping motions also help explain the stochasticity of human stepping.

Our paper is motivated to study the relationship between step location and energy optimality for balance recovery from a set of initial conditions, which simple models cannot characterize. We accurately quantify energy cost for reaching the whole stepping space using a full dynamics based physics simulation to fully explore this relationship. We use Bayesian optimization (BO), which is sample efficient for whole-body tasks [24], to optimize open parameters and achieve a wide range of stepping behaviors with different energy cost.

A. Scope
We develop an optimization pipeline and focus on the following set of stepping motions: taking a single step forward from an initialized standing position with the 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 paper 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 III-D3) Reachability maps that show energy-optimal step positions based on whole-body dynamics and the use of optimal control;

2) Optimization Pipeline (Section III-C, III-D) A parallelized optimization pipeline for whole-body control based stepping in full dynamics in a physics simulation;

3) Reachability Maps (Section III-D2) 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 III-D2) Finding of simple approximations and the disclosure of a funnel of near-optimal step locations.

By building sampling based energy consumption maps, we reveal the findings of a complex and nonlinear distribution of efficient stepping locations, and show regions of feasibility with similar efficiency, indicating simple heuristics and the non-uniqueness of balance recovery strategies – which can only be studied and understood by considering the complexity of whole-body dynamics and the optimal control as a whole. We find that such optimal step regions are very different from those predicted based on simple models which do not capture the whole-body dynamics of the system. Moreover, the reachability maps can be used to quickly select step locations and offer insight into explaining human step selections which do not need to repeat the same twice.

The paper structure is as follows: we mathematically define the pipeline and parameterized control system in Section II, then describe the methods of the pipeline in more detail (Section III). We then present our results in section IV, followed by our discussion (Section V) and conclusions (Section VI).

II. Problem Formulation
To study energy efficiency during stepping, we optimize a set of stepping motions to build reachability maps to encode energy consumption during stepping. We achieve this using our 2-phase pipeline, which is defined mathematically below.

Our pipeline centers on the interaction between parameterized trajectory generators and a Hierarchical Quadratic Programming (HQP) whole-body feedback controller. The HQP, $\mathcal{H}$, tuned for dynamic stepping, solves joint torques, $\tau$, given a set of task constraints:

$$\mathcal{H} : (c, s, t, q, \dot{q}, \ddot{q}) \rightarrow \tau,$$

where $(q, \dot{q}, \ddot{q}) \in \mathbb{R}^n$ are the position, velocity and acceleration of the robot’s joints, and $n$ is the number of joints and $t$ the current time. The reference trajectories $c$ and $s$ are for the
The output of phase 1 is the mapping \( \mathcal{G}(\hat{x}_0, s_{des}) \) to \( \mathcal{G}^*(\hat{x}_0, s_{des}) \). Open parameters are concatenated for sets of motions and outputs a parameter mapping; both mappings are combined for energy-optimal push recovery motions.

Trajectory Generation

CoM Trajectory

\( Eq.2a \)

Swing Foot Trajectory

\( Eq.2b \)

Reachability Maps

Query of Mapping

CoM State Estimation

\( x_0 \)

Trajectory

Environment

Robot

Joint Torques

Forward Dynamics

Reachability & Torque Measurements

Cost

\( J(\hat{x}_0, s_{des}, p) \)

Phase 1: Parameter Optimization

Sampled InitialVelocities \( \hat{x}_0 \in X_0 \)

Sampled Step Positions \( s_{des} \in S \)

Bayesian Optimization \( \argmax_p J(\hat{x}_0, s_{des}, p) \)

Phase 2: Reachability Maps

Sampled InitialVelocities \( \hat{x}_0 \in X_0 \)

Sampled Step Positions \( s_{des} \in S \)

TABLE I: Objective function weights and their values.

| Notation         | Affects               | Value   |
|------------------|-----------------------|---------|
| \( w_f \)        | Failure               | 0.001   |
| \( w_{swing} \)  | Swing Leg Position error | 50      |
| \( w_{z_{mid}} \)| Final CoM Position    | 1       |
| \( w_z \)        | Final CoM Height      | 1       |
| \( w_\tau \)     | Torque Consumption    | 0.0002  |

CoM and the swing foot respectively. Full details of the HQP are given in Section II-A.

Trajectories are time indexed position references such that \( c(t) \in \mathbb{R}^3 \) and \( s(t) \in \mathbb{R}^3 \), generated by the parameterized functions:

\[
\mathcal{G}_{\text{traj-gen}} : (t_{\text{min}}, s_{\text{max}}) \mapsto c
\]

\[
\mathcal{J}_{\text{traj-gen}} : (s_{\text{des}}, t_{\text{swing_start}}, s_{\text{speed}}) \mapsto s
\]

CoM trajectory generator \( \mathcal{G}_{\text{traj-gen}} \) takes minimum step time, \( t_{\text{min}} \), and maximum step length, \( s_{\text{max}} \), as arguments. Swing foot trajectory generator \( \mathcal{J}_{\text{traj-gen}} \) is a function of desired step position \( s_{\text{des}} \), swing start time \( t_{\text{swing_start}} \) and step speed \( s_{\text{speed}} \) parameters. Parameters are summarized in Table II. Phase 1 of our pipeline optimizes these open parameters to produce stepping motions. Open parameters are concatenated in vector \( p \):

\[
p = [t_{\text{min}}, s_{\text{max}}, t_{\text{swing_start}}, s_{\text{speed}}]^T
\]

The output of phase 1 is the mapping \( \mathcal{G} \) from initial CoM velocity, \( \hat{x}_0 \), and desired step position, \( s_{\text{des}} \), to a set of optimal parameters \( p^* \) (optimal values are denoted by \( (\cdot)^* \)), defined as:

\[
\mathcal{G} : (\hat{x}_0, s_{\text{des}}) \mapsto p^* = \argmax_p J(\hat{x}_0, s_{\text{des}}, p)
\]

such that maximizing this objective function will minimize all the error terms. To build this mapping, we maximize the objective function \( J \) for pairs of initial conditions \( \hat{x}_0 \) and \( s_{\text{des}} \) using BO.

At each BO iteration, the initial conditions and BO parameters are passed to the trajectory generators (Eq.2a, 2b), which are then passed as input to the HQP (Eq.1), which interfaces with a full dynamic simulation environment where the robot attempts to step towards the desired step position \( s_{\text{des}} \). This is highlighted in blue in Figure 3). The objective function \( J \) is a function of 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 I:

\[
J(\hat{x}_0, s_{\text{des}}, p) = -(w_f(t_{\text{total}} - t_{\text{term}}) + w_{\text{swing}}(s_{\text{des}} - s_{\text{td}})^2 + w_{z_{mid}}(z_f - z_{\text{mid}})^2 + w_z(z_{\text{nom}} - z_f) + w_\tau J_\tau(\hat{x}_0, s_{\text{des}}, p))
\]

The first term applies a cost for early termination, where termination time \( t_{\text{term}} \) is less than total simulation time \( t_{\text{total}} \), indicating that stepping failed. The value of \( t_{\text{total}} \) is predefined and is set to 7s in this paper. Simulations are terminated either when time \( t = t_{\text{total}} \) and the robot is stable having completed the motion, or at \( t_{\text{term}} \) if the robot falls (more details in Section III-B).

The difference between the swing foot position at touchdown, \( s_{\text{td}} \), and the desired step location, \( s_{\text{des}} \), incurs a cost proportional to 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 swing and the stance foot:

\[
\text{cost} = s_{\text{mid}} = s_{\text{stancce}} + (s_{\text{stance}} - s_{\text{td}})
\]

To encourage straight legs after landing, a cost is applied to final CoM height \( z_f \) to be as close as possible to a nominal height \( z_{\text{nom}} = 0.925 \text{ m} \). A term is added to minimize the integral of measured torque \( \tau \) in all the robot’s joints between the time swing foot motion begins \( t_\tau \) and touchdown time \( t_{\text{td}} \):

\[
J_\tau(\hat{x}_0, s_{\text{des}}, p) \mapsto \int_{t_\tau}^{t_{\text{td}}} \tau^2 dt
\]

After BO is completed for each pair of initial conditions, we are left with the mapping function \( \mathcal{G} \) (Eq.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

---

3
generalize to novel initial condition pairs. Here, the objective function $J$ is no longer used, as parameters have already been optimized and stored in the mapping function; instead, the dynamic simulation returns two values, shown in Figure 3: a binary reachability value to encode whether motions are successful, and the integral of the measured torque, $J_r$ (Eq.6), during the swing phase for every pair of initial conditions.

From these values we build a map to encode which step positions are reachable from each initial CoM velocity and a map to quantify the measured torque during the swing phase of each reachable step position. Using this measured torque map, we create a second mapping, $\mathcal{K}$, from the robot’s initial CoM velocity $x_0$ to an energy-optimal step position $s^*_{des}$.

Since $\mathcal{K}$ automatically generates energy-optimal step positions $s^*_{des}$, it is combined with mapping function $\mathcal{G}$ to reduce the input dimensionality such that only initial CoM velocity $x_0$ is required to generate optimal trajectory generation parameters $p^*$ to reach the energy-optimal step position $s^*_{des}$:

$$\mathcal{K} : (x_0) \rightarrow \{ p^* = \mathcal{G}(x_0, s^*_{des}) \}$$

$$s^*_{des} = \arg \min_{s_{des}} J_r(x_0, s_{des}, \mathcal{G}(x_0, s_{des})).$$

(7)

The maps are built offline and a regression model is used to approximate energy-optimal step positions $s^*_{des}$ for arbitrary initial CoM velocities $x_0$. After this pipeline is complete, we can query these mappings to execute push recovery motions, and we run dynamic simulations validation tests using the mappings and previously unseen initial conditions. We next define the control structure of the low-level HQP controller and the trajectory generation modules.

A. HQP Structure

The HQP feedback controller is a function of forward dynamics and two parameterized trajectories (Eq.1). We tuned the tasks, weights and hierarchy order of an existing HQP Controller [25]. It is worth noting that dynamic stepping is restricted if the CoM task has higher priority than the swing foot task, instead, CoM tasks should be at the same priority, or lower, than the swing foot.

B. Trajectory Generation

CoM and swing foot trajectory generation determine the motion produced by the HQP. Each has 2 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. Phase 1 of our pipeline (Section III-C), optimizes parameters for both trajectory generators for every pair of sampled initial conditions, which creates the mapping $\mathcal{G}$.

1) CoM Trajectory Generation: An existing LIPM based model [26] was used to produce CoM trajectories and is a function of two parameters: minimum step time $t_{min}$ and maximum step length $s_{max}$. An example trajectory is shown in Figure 4. Originally, the parameter $s_{max}$ was intended to reflect the capabilities of the real 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 $5^{th}$ degree minimum-jerk polynomials [27] (Figure 4), 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.

III. BUILDING REACHABILITY MAP: DYNAMIC STEPPING OPTIMIZATION PIPELINE

A. Simulation Setup

For the dynamic simulation, we used Pinocchio dynamics library [28]. In the long term we aim to move the query of mapping implementation onto the real robot, but was not possible for this paper. 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 a simulation frequency of 1000 Hz ($\Delta t = 0.001$ s). The BayesianOptimization package [29] was used for parameter optimization.

B. Experimental Setup

During this proof of concept stage, we constrain motions 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. We start with the
swing foot off the ground to avoid the added complexity of optimizing weight transfer time while we develop the pipeline.

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

C. Parameter Optimization - Phase 1

1) Bayesian Optimization: Phase 1 of our pipeline, shown in the blue box in Figure 3, optimizes a set trajectory generation parameters, $p$, for pairs of initial conditions. Algorithm 1 shows this process: for every initial CoM velocity $\dot{x}_0 \in X_0$ and every desired step position $s_{\text{des}} \in S_{\text{des}}$, a set of parameters $p$ is optimized via BO, such that for each pair, $[\dot{x}_0, s_{\text{des}}]$, is mapped to a set of parameters and stored in the mapping $G$.

In total, 150 sets of parameters were optimized, 15 initial CoM velocities, 10 step positions, with 170 BO iterations for each (100 random, 70 Bayesian) using objective Eq.5.

2) Parameter Interpolation: During optimization, the discrete points which were optimized represented a tiny proportion of the state space, so we interpolate between optimized points to create a continuous mapping $G$ (Eq.4) from arbitrary values of $\dot{x}_0$ and $s_{\text{des}}$ to a set of interpolated, optimized parameters $p^*$. We used element-wise grid interpolation to output a set of linearly interpolated parameter values between the nearest 4 sets of optimized parameters for any given pair of initial conditions.

D. Reachability Maps - Phase 2

The purpose of phase 2, shown in the orange box in Figure 3, is to simulate stepping motions from a greater number initial conditions than in phase 1 to produce dense reachability maps. This process is similar to phase 1, but instead of tuning parameters we query the optimal parameters from mapping function $G$ and pass these to the trajectory generation functions. Dense maps have three purposes: testing interpolation of optimized parameters, measuring the reachable space of the swing foot, and accurate measurement of joint torque during the swing phase of each step position.

We used 1000 initial condition pairs (40 initial CoM velocities, 20 desired step positions) to sample the mapping function, compared to 150 in phase 1. Initial velocities were between $X_0 = [0.1 \text{ m/s}, \ldots, 0.5 \text{ m/s}]$ at $0.017 \text{ m/s}$ intervals and desired step positions: $S_{\text{des}} = [0.1 \text{ m}, \ldots, 0.8 \text{ m}]$ at $0.029 \text{ m}$ intervals. A dynamic simulation episode was executed for each initial condition pair using parameters $p^* = G(\dot{x}_0, s_{\text{des}}) \forall \dot{x}_0 \in X_0, s_{\text{des}} \in S_{\text{des}}$, Eq.4.

Simulation episodes returned two values: a binary value to denote if the motion was successful, and the integral of joint torques, $\tau$, using Eq.6. As a result, each pair of initial conditions has an associated reachability value and measured torque value that we use to build the reachability maps.

1) Binary Reachability Map: A binary reachability map shows which step positions can be reached from initial conditions, shown in Figure 5b where green cells denote success and red failure. Parameters returned by the mapping $G$ lead to successful stepping over much of the map, but in at the top right map success is more noisy. It is likely that at the extremes

| Step Distance [m] | Initial CoM Vel [m/s] |
|-------------------|-----------------------|
| 0.10              | 0.21                  |
| 0.26              | 0.25                  |
| 0.33              | 0.28                  |
| 0.41              | 0.32                  |
| 0.57              | 0.35                  |
| 0.64              | 0.39                  |
| 0.80              | 0.43                  |

![Reachable Area](image)

![Unreachable Area](image)

![Support Vector](image)

![Validation Points](image)

Fig. 5: Binary reachability maps show successful (green) or unsuccessful (red) steps for pairs of initial conditions: (a) Parameter optimization map, (b) Dense reachability map, (c) SVM generated reachability map with validation testing points.
of the map, the range of parameters that lead to successful stepping is more narrow than at other points and interpolated parameters do not fall within this range. Optimizing more parameters would give better coverage, but would take longer to tune, we instead trim noisy regions from the map.

We train an Support Vector Machine (SVM) model to separate the reliable step locations from the noisy areas. Figure 5b shows the support vector overlaid on the reachability map. We then built a high resolution representation of the safe stepping area in Figure 5c by querying the SVM model with new initial condition pairs. We used a 3rd order SVM with a radial basis function kernel, reachable points had a weight of 1 and unreachable locations had a weight of 14.

2) Measured Torque Maps: The torque measured during the swing phase of motions from each initial condition pair is shown in Figure 6a, where each point in the trimmed reachability map is colored according to the integral of joint torques (Eq.6); darker colors denote higher measured torque. Step positions with the lowest measured torque for each sampled initial velocity are marked with a star. The distribution of measured torque patterns is highly nonlinear, but the pattern of steps with minimum measured torque forms a simple trend which can be used for footstep prediction. We consider the energy-optimal stepping positions to be those with the lowest measured 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 $4^{th}$ order polynomial regression, which can quickly approximate energy-optimal step positions given an initial CoM Velocity, as in Eq.7. Figure 6a shows this model captures the minimal energy step positions trends with a mean error of $216.12 \text{N}\cdot\text{m}$ (StD.$=287.84 \text{N}\cdot\text{m}$, Min.$=0 \text{N}\cdot\text{m}$, Max.$=1159.39 \text{N}\cdot\text{m}$). This gives us a quick method for selecting energy efficient step positions given the initial CoM velocity of the robot.

Alternatively, energy-optimal positions can be captured more closely using higher order polynomials, which would be slower than our method, or, a set of three linear models could also approximate these step locations less accurately, but more quickly by fitting at the start, end, and connecting them to approximate the middle points.

E. Query of Mapping

After the mappings have been built, we can query them in order to perform push recovery, this is shown in the green box in Figure 3, where we combine the mapping functions from the previous two phases. Therefore, for a given CoM velocity, an energy optimal step position is output by the mapping $s_{des}^* = \mathcal{K}(\hat{x}_0)$ (Eq.7), which is then used to generate the stepping parameters to produce this motion using the mapping $p^* = \mathcal{G}(\hat{x}_0, s_{des}^*)$ (Eq. 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 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 $\mathcal{G}$ under a range of initial conditions and the mapping $\mathcal{K}$ 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.

IV. RESULTS

A. Reachability Map Validation

We generated 1000 random pairs of initial conditions, $[\hat{x}_0, s_{des}]$, inside the safe stepping region in (Figure 5c) and ran an episode of dynamic simulation using the mapping $\mathcal{G}$. For each pair, stepping was dynamic simulated using parameters from the mapping $\mathcal{G}$ which was judged successful or unsuccessful according to the termination conditions in Section III-B. Figure 5c shows the initial conditions in the test set projected onto the safe reachability map where successful trials are marked in lighter green and all trials were successful.
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 \( s^*_{des} \) (Section III-D3). All trials were successful; Figure 7 shows snapshots of these motions, with more in the attached video.

Figure 8 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 Figure 7c. During the swing phase, the measured torque in the swing leg joints is lower than in the stance leg since it serves as a rigid pivot for the rest of the body. The landing impact is also clear in the measured torque, after the swing foot makes contact with the ground and since this is close to the actuation limits it can help explain the noisy results in Figure 5b. Comparing actuation profiles quasi-statically stable walking to explore the effects of dynamic stepping on the measured torque at the joints would be an interesting direction for future work.

C. LIPM Comparison

The mapping \( G \) was then queried forward simulate the LIPM model to predict one step capturability step positions with the same initial conditions, as shown in Figure 6b. After 0.32 m/s, the LIPM prediction is beyond the reachable area for the robot, so are not included. We calculated the error in measured torque between the optimal step positions from our optimization and those from the LIPM prediction, with a mean root squared error of 1364.5 N·m (StD. = 829.6 N·m, Min. = 164.3 N·m, Max. = 2531.9 N·m). Comparing our results with the LIPM shows the extent of the difference between the reduced and full dynamic models: the energy cost is higher when using simple models and many of step locations are outside the reachable area.

D. Diversity in Balancing Strategies

In addition to the simple trend of energy-optimal step positions, a diverse range of near-optimal step locations are highlighted on the map. The highlighted area in Figure 6a shows step positions where energy costs are a maximum of 5% above optimal for the same initial condition and Figure 6c shows a band of step positions which deviate from measured optimal by a maximum of 10%.

E. Underlying Energy Optimality

Near-optimal regions form simple heuristics: for low initial CoM velocities, stepping between 22 cm - 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.

Near-optimal efficiency regions suggest an underlying optimality in dynamic stepping gives us insight into human step selection. If similar regions exist in dynamic multi-body systems in general, humans can learn similar heuristics.
and use them for simple, rapid, near-optimal step selection. Given sensing delays in humans and inaccuracies in sensing disturbances, having near-optimal heuristics that work despite delays would be beneficial to developing humans.

Figure 2b, shows a clustering of selected steps similar to that in highlighted area in Figure 6a, but with a higher range of initial CoM velocities. Humans are able to 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 \( t_{\text{swing, start}} + 8 \text{ds} / s_{\text{speed}} \), and plotted in Figure 9b), showing all initial condition pairs and their optimized swing time, with only optimal points projected into 2D in Figure 9c. 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, \( s_{\text{swing, speed}} \), with only minor changes induced by the swing start time, \( t_{\text{swing, start}} \), as shown by the optimal ranges in Figure 9d.

F. Analysis of Computation Time

Training used an Intel Core i7-8700k with 12 cores (6 physical), 32GB 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 hours, reachability map building takes around 40 minutes. Parameters for one swing foot were used for the opposite foot.

Figure 9a 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 ms, 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.

V. DISCUSSION

The study of energy efficient locomotion is a complex process which involves the whole-body dynamics and control together as a whole and requires complex optimization, as well as global search of global energy-optimal step location and timing, which are all highly nonlinear. The complex interplay between multi-body dynamics, control and gait parameters is shown by the optimal gait parameters in Figure 9b, where this nonlinear relationship and non-smooth 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 piece-wise approximations, as shown in Figures 6a & 9c. 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 [3].

VI. CONCLUSION

In this paper, 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 Figure 9a). Hence reachability maps can be used for rapid step selection. 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] 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 International Conference on Intelligent Robots and Systems, vol. 1, 2001, pp. 239–246.

[2] J. A. Castano, Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, “Dynamic and reactive walking for humanoid robots based on foot placement control,” International Journal of Humanoid Robotics, vol. 13, no. 02, p. 1550041, 2016.

[3] C. McGreavy, K. Yuan, D. Gordon, K. Tan, W. J. Wolfslag, S. Vijayakumar, and Z. Li, “Unified push recovery fundamentals: Inspiration from human study,” in IEEE International Conference on Robotics and Automation, 2020, pp. 10876–10882.

[4] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in IEEE/RAS International Conference on Humanoid Robots, 2014, pp. 295–302.

[5] O. Khatib, L. Senti, J. Park, and J. Warren, “Whole-body dynamic behavior and control of human-like robots,” International Journal of Humanoid Robotics, vol. 1, no. 01, pp. 29–43, 2004.

[6] R. Deits and R. Tedrake, “Footstep planning on uneven terrain with mixed-integer convex optimization,” in Proc. IEEE/RSJ International Conference on Humanoid Robots, 2014, pp. 279–286.

[7] Y.-Y. Chung, H. H. Lee, “Real-time feasible footstep planning for bipedal robots in three-dimensional environments using particle swarm optimization,” IEEE/ASME Transactions on Mechatronics, vol. 25, no. 1, pp. 429–437, 2019.

[8] K. Yuan, 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,” Advanced Robotics, vol. 31, no. 22, pp. 1238–1252, 2017.

[9] Z. Li, J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, “The pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives,” in Proc. IEEE/RSJ International Symposium on System Integration, 2019, pp. 614–619.

Christopher McGreavy received his M.Sc in Computational Neuroscience and Cognitive Robotics from the University of Edinburgh, and M.Sc.(R) from the University of Edinburgh in Robotics and Autonomous Systems. He is currently a PhD student at the University of Edinburgh with research interests in robotic and human locomotion, enhancing understanding and application of efficient, dynamic walking.

Zhibin (Alex) Li is an assistant professor at the School of Informatics, University of Edinburgh. He obtained a joint PhD degree in Robotics at the Italian Institute of Technology (IIT) and University of Genova in 2012. His research interests are in creating intelligent behaviors of dynamical systems with human comparable abilities to move and manipulate by inventing new control, optimization and deep learning technologies.