Reinforcement Learning With Energy-Exchange Dynamics for Spring-Loaded Biped Robot Walking

Cheng-Yu Kuo, Hirofumi Shin, and Takamitsu Matsubara, *Member, IEEE*

***Abstract*—This letter presents a probabilistic Model-based Reinforcement Learning (MBRL) approach for learning the Energy-exchange Dynamics (EED) of a spring-loaded biped robot. Our approach enables on-site walking acquisition with high sample efficiency, real-time planning capability, and generalizability across skill conditions. Specifically, we learn the data-driven state transition dynamics of the robot in the formulation of energy-states, with their interaction characterized as energy-exchange to reduce dimensionality. To improve planning reliability with the learned EED, we design a control space based on a walking trajectory that follows the law of conservation of energy and is formulated by energy-states. We evaluated our approach using a four-degree-of-freedom spring-loaded biped robot in simulation and hardware, and generalizability is validated by using the same learning framework for different walking speeds and terrains in simulation and walking acquisition with hardware. All results showed successful on-site walking acquisition with a compact nine-dimension dynamics model, 40 Hz real-time planning, and on-site learning within a few minutes.**

***Index Terms*—Humanoid and bipedal locomotion, model learning for control, reinforcement learning.**

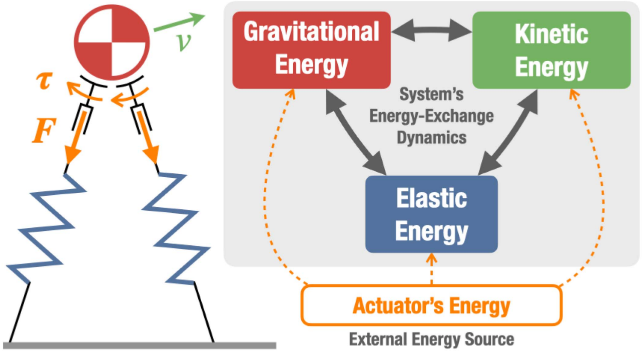

Fig. 1. Illustration of the Energy-Exchange Dynamics (EED) of a spring-loaded biped robot. The energy flows between the gravitational, kinetic and elastic energy, with actuators being energy sources that provide energy to the system. This work uses MBRL to learn this EED to perform biped walking.

I. Introduction

COMPLIANT biped robots have gained attention due to their potential for deployment in human-centric environments, resemblance to humans in anatomy and behavior [1] that can handle impacts and traverse different terrains [2], and the captivating prospect of a non-organic machine performing a human-associated task. However, the increased compliance of such robots results in a complex dynamic system, making analytical approaches challenging.

Several learning-based studies tackled the problem of compliant biped locomotion and achieve different biped maneuvers. These methods include Model-free Reinforcement Learning (MFRL) [3], [4] and Bayesian Optimization (BO) [5], [6]. However, both MFRL and BO are black-box approaches that are task-specific and data-intensive, requiring virtual scaling like Sim-to-Real to learn each gait. To enhance generalizability, parameterized policies are learned with MFRL and BO to achieve different walking speeds [7], [8], walking heights [9], or both [10]. Nevertheless, these policies are limited to the predefined walking parameters involved in the training, and adding new parameter requires long re-learning. This not only lengthens the learning process, but also limits on-site learning capability.

As a promising solution for achieving sample efficiency and generalizability, probabilistic Model-based Reinforcement Learning (MBRL) involves learning a data-driven dynamics model of a robot and perform different skills using probabilistic Model-Predictive Control (pMPC) with varying control objectives [11], [12]. However, due to the computational expense of MBRL [13], previous studies on biped locomotion were mainly conducted in simulation [14], [15] or with physical robot using offline planning [16]. Furthermore, more training samples and high dimension dynamics are needed to accommodate the compliance dynamics of compliant biped robots. The increase in sample size and model dimension largely impacts the control frequency, as the computation load of pMPC depends on either the training sample size [17] or model dimension [18], making implementation of compliant biped locomotion challenging.

With the above considerations in mind, our objective is to develop an MBRL approach that can 1) learn the dynamics of a compliant biped robot on-site with high sample efficiency, 2) achieve locomotion skills at a high online planning frequency, and 3) have generalizability across skill conditions.

This work presents an MBRL approach that learns the Energy-exchange Dynamics (EED) to enable a spring-loaded biped robot to walk. Specifically, we view actuators as energy sources by incorporating the energy conservation equation and use MBRL to learn a data-driven state transition dynamics of the

Manuscript received 16 April 2023; accepted 24 July 2023. Date of publication 9 August 2023; date of current version 22 August 2023. This letter was recommended for publication by Associate Editor S. Ha and Editor J. Kober upon evaluation of the reviewers' comments. *(Corresponding author: Cheng-Yu Kuo.)*

Cheng-Yu Kuo and Takamitsu Matsubara are with the Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara 630-0192, Japan (e-mail: kuo.cheng-yu.jy5@is.naist.jp; takam-m@is.naist.jp).

Hirofumi Shin is with the Honda R&D, Ltd., Saitama 351-0114, Japan (e-mail: hirofumi_shin@jp.honda).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3303786, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3303786

robot in the formulation of energy-states to reduce dimensionality with their interaction characterized as energy-exchange, Fig. 1. For reliable pMPC planning with EED, we establish a state-aware control space that utilize with the observed robot state and an energy-state-based reference walking trajectory we designed by following the law of conservation of energy. Our approach's generalizability is validated through different walk speeds, continuous changing walk speeds, and uneven terrains in simulation and walking acquisition with hardware. All results showed successful on-site walking acquisition with 1) a compact nine-dimension dynamics model, 2) 40 Hz real-time planning capability, and 3) on-site learning within a few minutes.

Comparing to our previous work which utilized energy to learn spring dynamics and perform a hopping task with a simulated robot [19], this work incorporates the energy conservation equation of actuators to further reduce dimensionality, improves pMPC planning reliability, tests generalizability, and successfully implemented onto hardware.

II. Preliminaries

MBRL with GPs dynamics is sample-efficient [12], but its control frequency with pMPC grows exponentially with the training sample size, making it unsuitable for high control frequency tasks like biped locomotion. Building on previous successes of robotic applications [19], [20], we implemented the Fourier-featured Linear Gaussian Model (LGM-FF) [18] to alleviate the sample size and control frequency trade-off.

A. Probabilistic Dynamics Acquisition and State Prediction

Let a dynamical system, $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) + \epsilon$, represents the system's true dynamics as discrete-time state transition. Here, $\boldsymbol{x}_t \in \mathbb{X} \subset \mathbb{R}^D$ is the state, $\boldsymbol{u}_t \in \mathbb{U} \subset \mathbb{R}^U$ is the control, and ϵ is the system noise that follows Gaussian distribution. We approximate a latent dynamics model for each prediction dimension via LGM-FF, $f := \{f_i(\cdot)\}_{i=1,\ldots,D}$, with N collected samples from the system during trials. In particular, the training samples are formulated as input $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N]$, where $\tilde{\boldsymbol{x}}_t := (\boldsymbol{x}_t, \boldsymbol{u}_t) \in \mathbb{R}^{D+U}$, and the target $\boldsymbol{y}_i = [x_{i,2}, \ldots, x_{i,N+1}]$ as the i-th element of $\boldsymbol{x}_t$.

As LGM-FF predicts future state as a Gaussian distribution, recursive uncertainty propagation is required for multi-step predictions. With a Gaussian distributed state $p(\boldsymbol{x}_t)$, the future state distribution $p(\boldsymbol{x}_{t+1}) \approx \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$ is approximated by exploiting moment-matching [18], [21] with the LGM-FF. Specifically, the predictive state $p(\boldsymbol{x}_{t+1})$ is obtained by integrating LGM-FF latent dynamics $f(\cdot)$ over the state distribution $p(\boldsymbol{x}_t)$, denoted as function $f_M(\cdot)$:

$$p\left(\boldsymbol{x}_{t+1}\right) = f_M\left(p\left(\boldsymbol{x}_t\right), \boldsymbol{u}_t\right) = \int f\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) p\left(\boldsymbol{x}_t\right) d\boldsymbol{x}_t, \quad (1)$$

where analytic solutions are provided in a previous study [18].

Comparing the computation cost of exploiting moment-matching with standard GPs model, $\mathcal{O}(DN^2)$, LGM-FF results in $\mathcal{O}(DM^2)$ which is independent to collected sample size N, where M is the feature size used in LGM-FF, and was demonstrated that $M \ll N$ in robotic tasks [18], [19].

B. Finite-Horizon Probabilistic Model Predictive Control

Given an input state distribution $p(\boldsymbol{x})$, pMPC returns the optimal H-step control $\boldsymbol{u}^\star := [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_H]$ by minimizing the expected finite-horizon loss $\mathcal{L}(p(\boldsymbol{x}))$ derived by recursive moment-matching via (1) [12], [20]:

$$\left.\begin{array}{ll} \text{minimize}_{\boldsymbol{u}^\star} & \mathcal{L}\left(p\left(\boldsymbol{x}\right)\right) = \sum_{k=2}^{H+1} \mathbb{E}\left[\ell\left(\hat{\boldsymbol{x}}_k\right) | p\left(\hat{\boldsymbol{x}}_k\right)\right] \\ \text{subject to} & p\left(\hat{\boldsymbol{x}}_{k+1}\right) = f_M\left(p\left(\hat{\boldsymbol{x}}_k\right), \hat{\boldsymbol{u}}_k\right) \\ & p\left(\hat{\boldsymbol{x}}_k\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \ k = 1, \ldots, H+1 \\ & \hat{\boldsymbol{u}}_k \in \mathbb{U}, \ k = 1, \ldots, H \end{array}\right\} \quad (2)$$

where $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{H+1}$ are the predictive state rollouts with initial state distribution $p(\hat{\boldsymbol{x}}_1) = p(\boldsymbol{x})$, and $\ell : \mathbb{X} \to \mathbb{R}$ is the immediate loss. Within a task with a full horizon H_f, pMPC re-solves an optimization problem over a receding horizon H ($H < H_f$) with newly observed state of all time instants, making it robust to environmental changes. The computation cost of pMPC with LGM-FF is $\mathcal{O}(HDM^2)$.

III. Proposed Method

We propose an MBRL approach consisting of three strategies to achieve walking on a spring-loaded biped robot:

1) *EED Model and Energy-state:* Motivated by the law of conservation of energy [22], we use LGM-FF with energy-state to model the EED of the robot to express its CoM, elastic components and actuators' dynamics in the formulation of energy with their interaction characterized by energy-exchange. The use of EED can reduce dimensionality as some state instances can be implicitly expressed.
2) *Energy-state-based Walking Trajectory:* As pMPC leverages multi-step state predictions for control planning, the use of energy-state converts the intended walking task into an energy control problem. To achieve successful walking with pMPC, we incorporate the law of conservation of energy and design an energy-state-based reference walking trajectory.
3) *State-aware Control Space:* To address the challenge of ensuring reliable controls in pMPC planning, we propose a state-aware control space that utilizes both the provided energy-state-based trajectory and the observed robot state to constrain the pMPC exploration.

A. Simplified System's Variables and Gait Parameters

In this section, we use a simplified spring-loaded planer biped system to explain our approach, and our implementation to a spring-loaded biped robot is detailed in Section IV.

1) Simplified System's Variables: Consider an m-kg point-mass planer biped robot system with k spring-loaded actuators, characterized by a spring constant matrix $\boldsymbol{K} \in \mathbb{R}^{k \times k}$. The following variables are assumed available: the Center-of-Mass (CoM) height h, CoM velocity $\boldsymbol{v} \in \mathbb{R}^V$, spring deflections $\boldsymbol{\Theta}_S \in \mathbb{R}^k$, and actuators' position and velocity $\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}} \in \mathbb{R}^k$.

2) Gait Parameters: The L_{swg}, L_{sup} denotes the leg length of the swing and support leg as well as their respective orientation in world-space $\psi^w_{swg}, \psi^w_{sup}$, Fig. 2. L and ψ^w are assumed obtainable from actuator $\boldsymbol{\Theta}$ and spring's positions $\boldsymbol{\Theta}_S$. The walking gait is then described by four parameters: the stance leg length L_{stan}, the lifted leg length L_{lift}, the desired stride angle θ_{stride}, and the desired CoM velocity at the peak $\boldsymbol{v}_p \in \mathbb{R}^V$. This walking gait is divided into two phases with distinct objectives, Double-Support (DS) and Single-Support (SS), with the rear leg

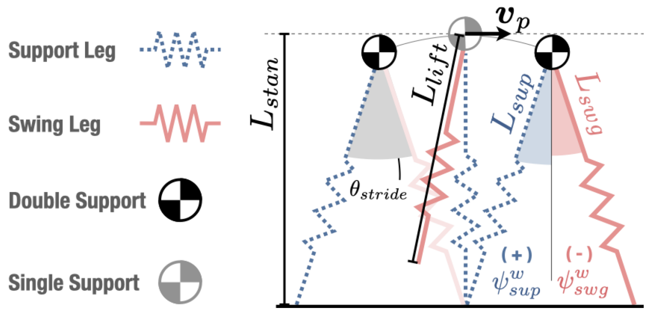

Fig. 2. Walking gait with the swing/support leg definition and gait parameters.

during DS being the support leg. Conditions for phase transition is detailed in Section IV-C1.

3) Gait Motion: With the gait parameters, the target gait motion is shown in Fig. 2, with following requirements:

- DS phase: the angle between two legs matches θ_{stride} and the swing leg's length matches the stance leg length. We denote these requirements as $\boldsymbol{R}^{DS} = \{\psi^w_{swg} = -0.5\theta_{stride}, \psi^w_{sup} = 0.5\theta_{stride}, L_{swg} = L_{stan}\}$.
- SS phase: the support leg's length matches the stance leg length such that the system act as an inverted pendulum, with requirement denoted as $\boldsymbol{R}^{SS} = \{L_{sup} = L_{stan}\}$.

B. EED Model and Energy-State

Using MBRL to learn the EED model of the robot, $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) + \epsilon$, requires an energy-state $\boldsymbol{x}$ and a user-defined control $\boldsymbol{u}_t$ based on how actuators are controlled, our definition is detailed in Section IV-C3. First, we denote the system energy $\boldsymbol{E} \in \mathbb{R}^{V+2}$ as

$$\boldsymbol{E} := [E_g, \boldsymbol{E}_k, E_S]^\top = [\mathrm{m}|\boldsymbol{g}|h,\ \frac{1}{2}\mathrm{m}\boldsymbol{v}^{\circ 2},\ \frac{1}{2}(\boldsymbol{K}\boldsymbol{\Theta}_S)^\top\boldsymbol{\Theta}_S]^\top, \tag{3}$$

where E_g is the gravitational energy, $\boldsymbol{E}_k$ is the kinetic energy, E_s is springs' total elastic energy [19], $\boldsymbol{g}$ is the gravity vector and $\circ$ denotes element-wise operations. Without an external energy source, the system's total energy remain constant; that is, $\mathbf{1}^\top\boldsymbol{E} =$ const., where $\mathbf{1}$ is a vector of ones.

Next, we assume the actuators are the only energy source of the robot system. The energy of the actuator, E_{act}, is derived by taking the integral of motion [23] of their Equation-of-Motion: $\boldsymbol{\tau} = \boldsymbol{J}\ddot{\boldsymbol{\Theta}} + \boldsymbol{K}\boldsymbol{\Theta} + \boldsymbol{D}\dot{\boldsymbol{\Theta}}$:

$$\begin{aligned} E_{act} &= \int \boldsymbol{\tau}^\top\dot{\boldsymbol{\Theta}}dt \\ &= \frac{1}{2}(\boldsymbol{J}\dot{\boldsymbol{\Theta}})^\top\dot{\boldsymbol{\Theta}} + \frac{1}{2}(\boldsymbol{K}\boldsymbol{\Theta})^\top\boldsymbol{\Theta} + \int(\boldsymbol{D}\dot{\boldsymbol{\Theta}})^\top\dot{\boldsymbol{\Theta}}dt, \end{aligned} \tag{4}$$

where $\boldsymbol{\tau}$ contains actuators' torque, $\boldsymbol{J}$ is the inertia matrix, $\boldsymbol{D}$ is the damping matrix. By introducing the actuator's energy, the system's energy conservation equation becomes:

$$\begin{aligned} \mathbf{1}^\top\boldsymbol{E} &= E_{act} + \mathrm{Const.} \\ &= \frac{1}{2}(\boldsymbol{J}\dot{\boldsymbol{\Theta}})^\top\dot{\boldsymbol{\Theta}} + \frac{1}{2}(\boldsymbol{K}\boldsymbol{\Theta})^\top\boldsymbol{\Theta} + \mathrm{Const.}, \end{aligned} \tag{5}$$

which shows the robot's system energy $\boldsymbol{E}$ is a function of the actuator's position $\boldsymbol{\Theta}$ and velocity $\dot{\boldsymbol{\Theta}}$. As the system's energy $\boldsymbol{E}$ is obtained from observed CoM state and spring deflections, the actuator's position $\boldsymbol{\Theta}$ and velocity $\dot{\boldsymbol{\Theta}}$ can be implicitly expressed by one another. In this work, we select the actuator's position and design the energy-state as

$$\boldsymbol{x}_t = \left[\boldsymbol{\Theta}_t^\top, \boldsymbol{E}_t^\top\right]^\top \in \mathbb{R}^{k+V+2}, \tag{6}$$

which reduces $2k - 1$ dimensions from the standard state

$$\boldsymbol{x}_t = \left[\boldsymbol{\Theta}_t^\top, \dot{\boldsymbol{\Theta}}_t^\top, h, \boldsymbol{v}, \boldsymbol{\Theta}_s^\top\right]^\top \in \mathbb{R}^{3k+V+1}. \tag{7}$$

C. Energy-State-Based Walking Trajectory

Double-Support: Based on the law of conservation of energy and the provided gait parameters, the system has to posses the target energy $E^\star \in \mathbb{R}$ to follow the desire gait:

$$E^\star = E_g^\star + \mathbf{1}^\top\boldsymbol{E}_k^\star = \mathrm{m}|\boldsymbol{g}|L_{stan} + 0.5\mathrm{m}\mathbf{1}^\top\boldsymbol{v}_p^{\circ 2}, \tag{8}$$

where $E_g^\star \in \mathbb{R}$ and $\boldsymbol{E}_k^\star \in \mathbb{R}^V$ are the target gravitational and kinetic energy, with $\boldsymbol{v}_p$ the desired CoM velocity at the peak. Therefore, the objective is provide a trajectory that pMPC can find optimal controls that supply the system with necessary energy to progress forward. As the target energy remain constant, the reference trajectory at time-step t are:

$$E_{g,t}^\star = E_g^\star = \mathrm{m}|\boldsymbol{g}|L_{stan}, \tag{9}$$

$$\boldsymbol{E}_{k,t}^\star = \boldsymbol{E}_k^\star = 0.5\mathrm{m}\boldsymbol{v}_p^{\circ 2}. \tag{10}$$

and the corresponding reference trajectory of the DS phase is

$$\boldsymbol{T}_t^{DS} = \left\{E_{g,t}^\star, \boldsymbol{E}_{k,t}^\star\right\}. \tag{11}$$

Single-Support: Assuming the system has gained energy after the DS phase, the objective of the SS phase is to execute a stable leg swing with the swing leg in order to progress to the DS phase. To synchronize the inverted pendulum's swing motion, the swing leg's references, $L_{swg}^\star$ and $\psi_{swg}^\star$, must be a function of the support leg orientation ψ^w_{sup}. As long as the swing leg's terminal configuration matches the defined stance configuration of the DS phase, the swing leg's intermediate trajectory can be chosen according to the desired gait pattern. We denote the swing leg's trajectory as follows:

$$L_{swg,t}^\star = S_L\left(\psi^w_{sup,t}\right), \tag{12}$$

$$\psi_{swg,t}^\star = S_\psi\left(\psi^w_{sup,t}\right), \tag{13}$$

where $S_L, S_\psi : \mathbb{R} \to \mathbb{R}$ are mappings between support leg orientation ψ^w_{sup} and reference swing leg's trajectories. Our application is detailed in Section IV-C2. Therefore, the reference trajectory of the SS phase is

$$\boldsymbol{T}_t^{SS} = \left\{L_{swg,t}^\star, \psi_{swg,t}^\star\right\}. \tag{14}$$

Although the SS phase's reference trajectories are formulated as positions; however, given the EED model is characterized as position and energies, energies are incorporated to predict future positions in pMPC.

D. State-Aware Control Space

The state-aware control space for pMPC modifies the exploration control space $\mathbb{U}$ in (2) to reduce the chance of the pMPC producing unreliable controls. At each time-step t, the modification is based on system's state $\boldsymbol{x}_t$, the reference trajectory $\boldsymbol{x}_t^\star$,

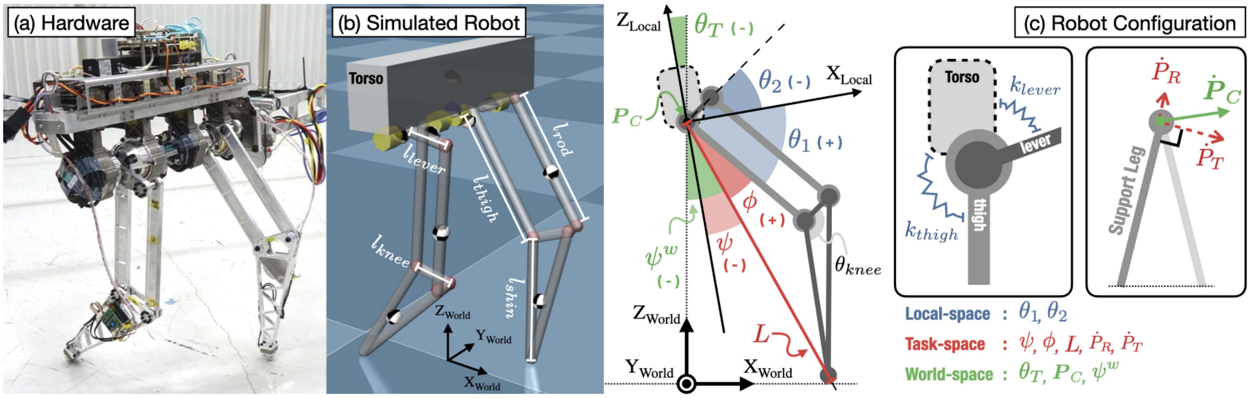

Fig. 3. (a) The spring-loaded biped robot [1]. (b) The simulated robot in Mujoco. (c) The biped robot configuration with variable descriptions in Table I. Robot parameters: mass m $= 11.25$ kg; $l_{thigh}, l_{shin}, l_{rod} = 0.25$ m; $l_{lever}, l_{knee} = 0.0705$ m; $\theta_{knee} = 110^\circ$; $k_{thigh}, k_{lever} = 150$ Nm/rad.

the user's prior knowledge of which control signals are more reliable:

$$\mathbb{U}'_t = g\left(\boldsymbol{x}_t, \boldsymbol{x}^\star_t, \mathbb{U}\right), \tag{15}$$

where $g : \mathbb{X} \times \mathbb{U} \to \mathbb{U}' \subset \mathbb{U}$ modifies the control space, and our application of this method is discussed in Section IV-C4.

IV. Setup for Experiment and Implementation

A. Environment

1) Hardware: We selected a spring-loaded biped robot [1], featuring lightweight legs with parallel-linkage structures and four spring-loaded actuators driving its lever and thigh links for prismatic compliance, The lightness of the legs enabled us to approximate the robot's CoM with its hip joint. Actuators were velocity controlled at 40 Hz with limit of ±6 rad/s. The robot was attached to a rotational boom and constrained in the sagittal plane. Conversion between local, task, and world-space is provided in Appendix A. Hardware results were computed on the on-robot computer with an i7-4700EQ CPU.

2) Simulation: A simulated replica of the robot was built in Mujoco (Fig. 3(b)), with friction and damping in all joints to mimic the real robot. The simulated robot was constrained in the sagittal plane (xz-plane) and controlled at a frequency of 40 Hz, matching the real robot. Simulated experiments were conducted using an Apple M1 Max-equipped computer.

B. EED Model Configuration for MBRL

Following (7), the robot's energy-state is defined as:

$$\boldsymbol{x} = \left[\boldsymbol{\Theta}^\top, \theta_T, \boldsymbol{E}^\top\right]^\top \in \mathbb{R}^9, \tag{16}$$

where θ_T is added to obtain world-space leg orientation. $\boldsymbol{\Theta}$ and $\boldsymbol{E}$ are defined as

$$\boldsymbol{\Theta} = \left[\phi_{swg}, \psi_{swg}, \phi_{sup}, \psi_{sup}\right]^\top \in \mathbb{R}^4, \tag{17}$$

$$\boldsymbol{E} = \left[E_g, E_R, \frac{\dot{P}_T}{|\dot{P}_T|}E_T, E_S\right]^\top \in \mathbb{R}^4, \tag{18}$$

with ϕ being the inner angle of each leg corresponding to the leg length L; E_g is the gravitational energy; E_S is the total elastic energy of all springs; E_R and E_T are the kinetic energies obtained from radial and tangential CoM velocities ($\dot{P}_R, \dot{P}_T$), referring to the support leg, Fig. 3(c). Directions are added to the tangential kinetic energies for additional information (positive when walking forward). Both ϕ and ψ include spring deflections and are configured in task-space to simplify the walking control problem, and are obtained from actuator position and spring deflections, as detailed in Appendix A. Dynamics models for the DS and SS phase are independently trained with samples collected in that phase.

C. pMPC Planning Objective

1) Walking Gait Parameters: Stance leg length L_{stan} and lifted leg length L_{lift} were measured when the joint angles $\phi_{\min} = 14^\circ$ and $\phi_{\max} = 37^\circ$; the target stride angle is $\theta_s = 28^\circ$; the target velocity at peak is $\boldsymbol{v}_p$ is 0.4 m/s in the x-direction. Therefore, we obtained the target energy as

$$E^\star = \mathrm{m}|\boldsymbol{g}|L_{stan} + 0.5\mathrm{m}\mathbf{1}^\top \boldsymbol{v}_p^{\circ 2} \approx 54.426\ \mathrm{J}. \tag{19}$$

The transition between phases is determined by feet's ground reaction forces, $\boldsymbol{F}_{swg}$ and $\boldsymbol{F}_{sup}$, with conditions: a) DS → SS if $|\boldsymbol{F}_{sup}| \leq 5\mathrm{N}$, and b) SS → DS if $|\boldsymbol{F}_{swg}| \geq 20\mathrm{N}$. $\boldsymbol{F}_{swg}$ and $\boldsymbol{F}_{sup}$ are only used to identify phase transitions and are not used in the pMPC planning.

2) Reference Swing Leg Trajectory for the SS Phase: We designed a human-inspired reference trajectory for our application by incorporating a recent human gait analysis [24]. To achieve a smooth motion, the reference swing leg trajectory was heuristically fitted to the human gait [24] using multiple sigmoid functions, as shown by the black dashed lines in Fig. 8. This allowed us to execute a stable leg swing.

3) Control Strategy and pMPC Objective Function: Followings are designed associating with the reference trajectories $\boldsymbol{T}^{DS}_t, \boldsymbol{T}^{SS}_t$ from (11) and (14), and the requirements $\boldsymbol{R}^{DS}, \boldsymbol{R}^{SS}$ from Section III-A3.

Double Support: a kicking motion is anticipated to provide the system with necessary energy, thus define the control as

$$\boldsymbol{u}^{DS} := \left[\dot{\phi}_{sup}\right], \tag{20}$$

and the immediate loss for the pMPC is set accordingly:

$$\ell_{DS}\left(\boldsymbol{x}_k\right) := -\exp\left(-|E^\star - \mathbf{1}^\top \boldsymbol{E}_k|\right), \tag{21}$$

TABLE I
SYSTEM VARIABLES (VAR.) IN LOCAL-, TASK-, AND WORLD-SPACE

Space	Var.	Description
Local	θ_1	Joint 1 position. Actuator 1 position with spring deflection.
	θ_2	Joint 2 position. Actuator 2 position with spring deflection.
Task	ψ	Leg's orientation. Obtained from θ_1, θ_2.
	ϕ	Leg's inner angle. Obtained from θ_1, θ_2.
	L	Leg's length. Obtained from ϕ.
World	θ_T	Torso orientation. Obtained from IMU.
	$\mathbf{P}_C$	Center-of-mass position. Approximated with the hip joint. Obtained from support leg.
	ψ^w	Leg's orientation in world-space. Obtained from ψ and θ_T

where the directional kinetic energy in (18) encourages gaining energy by kicking forward. Following $\boldsymbol{R}^{DS}$, positions of ϕ_{sup}, ψ_{sup}, and ψ_{swg} are fixed via PD controllers.

Single Support: the goal is to perform a leg swing while keeping its body orientation at $-5°$ to avoid exceeding the actuator limits due to the hardware constraints. Given $\boldsymbol{R}^{SS}$, the support leg length is held by PD controllers with additional $\psi_{sup} = (-5° - \theta_T)$ to counteract the torso tilt. Meanwhile, given the reference state the optimal swing leg control:

$$\boldsymbol{u}^{SS} := \left[\dot{\phi}_{swg}, \dot{\psi}_{swg}\right]^\top, \tag{22}$$

is obtained through pMPC with an immediate loss set to:

$$\begin{aligned}\ell_{SS}\left(\boldsymbol{x}_k\right) := \; & -0.5 \exp\left(-|\phi^\star_k - \phi_{swg,k}|\right) \\ & - 0.5 \exp\left(-|\psi^\star_k - \psi^w_{swg,k}|\right),\end{aligned} \tag{23}$$

where the conversion from $L^\star_{swg,k}$ to $\phi^\star_{swg,k}$ and obtaining the world-space leg orientation ψ^w is provided in Appendix A.

4) State-Aware Control Space: The purpose is providing a reliable control space to constrain pMPC exploration. As the reference state of the pMPC is already known ($\phi^\star_{swg}, \psi^\star_{swg}, E^\star$), we find an theoretical control relative to these references and establish a control range for pMPC. Specifically, we modify the control space as adding a range of velocities $[-\delta, \delta]$ above the theoretical velocities $u^\star_i$:

$$\mathbb{U}'_t := \{u_i \in [u^\star_i - \delta_i, u^\star_i + \delta_i]\}_{\forall u_i \in \boldsymbol{u}_t} \subset \mathbb{U}. \tag{24}$$

For all steps of the DS phase, the theoretical kicking velocity $\dot{L}^\star_{sup}$ is obtained from the required radial kinetic energy $E^\star_R$:

$$E^\star_R = \max\left(E^\star - E_g - E_T, 0\right), \tag{25}$$

$$\dot{L}^\star_{sup} = \sqrt{2\mathrm{m}^{-1} E^\star_R}, \tag{26}$$

where (25) calculates the lacking energy that the radial kinetic energy can provide, and 26 converts $E^\star_R$ into the theoretical velocity, which is converted to $\dot{\phi}^\star_{sup}$ via (37).

For all steps in the SS phase, the theoretical velocities $\dot{\psi}^\star_{swg}$ and $\dot{\psi}^\star_{sup}$ are obtained by scaling the distance between the current state and the reference trajectory:

$$\dot{\phi}^\star_{swg,t} = \alpha\left(\phi^\star_{swg,t} - \phi_{swg,t}\right), \tag{27}$$

$$\dot{\psi}^\star_{swg,t} = \alpha\left(\psi^\star_{swg,t} - \psi_{swg,t}\right), \tag{28}$$

where α is a scaling factor, and we set to control frequency $\alpha = 40$ for exact conversion from position to velocity.

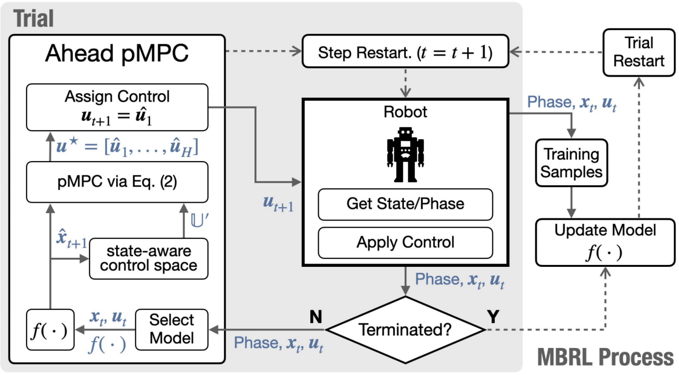
Trial
Ahead pMPC
Assign Control
$\boldsymbol{u}_{t+1} = \hat{\boldsymbol{u}}_1$
$\boldsymbol{u}^\star = [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_H]$
pMPC via Eq. (2)
$\mathbb{U}'$
$\hat{\boldsymbol{x}}_{t+1}$
state-aware control space
$f(\cdot)$
$\boldsymbol{x}_t, \boldsymbol{u}_t$
$f(\cdot)$
Select Model
Step Restart. $(t = t + 1)$
Trial Restart
Robot
$\boldsymbol{u}_{t+1}$
Get State/Phase
Apply Control
Phase, $\boldsymbol{x}_t$, $\boldsymbol{u}_t$
Training Samples
Update Model $f(\cdot)$
Phase, $\boldsymbol{x}_t$, $\boldsymbol{u}_t$
N
Terminated?
Y
Phase, $\boldsymbol{x}_t$, $\boldsymbol{u}_t$
MBRL Process

Fig. 4. MBRL process with ahead pMPC planning.

Based on the theoretical velocities at each time-step, we define the control space for the DS and SS phase as

$$\mathbb{U}'_{DS} = \left\{\dot{\phi}_{sup} \in \left[\dot{\phi}^\star_{sup} - \delta, \dot{\phi}^\star_{sup} + \delta\right]\right\}, \tag{29}$$

$$\mathbb{U}'_{SS} = \left\{\begin{array}{l}\dot{\phi}_{swg} \in \left[\dot{\phi}^\star_{swg} - \delta, \dot{\phi}^\star_{swg} + \delta\right] \\ \dot{\psi}_{swg} \in \left[\dot{\psi}^\star_{swg} - \delta, \dot{\psi}^\star_{swg} + \delta\right]\end{array}\right\}, \tag{30}$$

where δ is defined as 1.2 rd/s for simulated robot and 0.6 rd/s for hardware, which pMPC explores approximate 20% and 10% of the actuator's full capability.

D. MBRL Process

In our experiments, MBRL repeated trials until reaching the target sample duration. In each trial, steps were repeated at a rate of 40 Hz until any termination condition was satisfied. During each step, an ahead pMPC scheme [20] was applied to find the optimal control while alleviating control delay. Specifically, at time step t, the pMPC finds the optimal control sequence $\boldsymbol{u}_\star = [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_H]$ of length $H = 3$ based on the dynamics model of that phase and a predictive state $\hat{\boldsymbol{x}}_{t+1}$, as obtained via (1). The first control signal $\hat{\boldsymbol{u}}_1$ is then assigned as the one-step-ahead control that will be applied to the system at time step $t + 1$. After each trial is completed, the MBRL model is updated using samples collected from all previous trials. The learning process is summarized in Fig. 4.

Termination conditions of for simulation trials included reaching a target distance of 20 m, fall over ($P_z \leq 0.35$ m), or falling back ($\dot{P}_x \leq -0.75$ m/s), where P_z is the z-element of P_C and $\dot{P}_x$ is the x-element of $\dot{P}_C$. For hardware trials, conditions included reaching joint limits ($\theta_1 \notin [36.5°, 143.5°]$ and $\theta_2 \notin [-40.5°, 83.5°]$) or the operator engaging the emergency stop when the robot was unstable or had reached a target distance (≈8.23 m).

For all experiments described below, samples are collected on-site and dynamics are learned from scratch. This means that no skill or model is transferred between experiments.

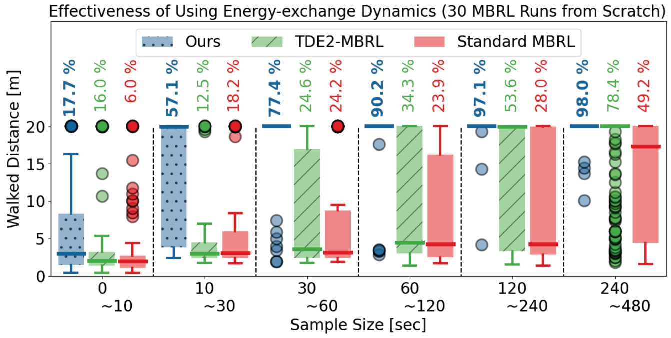

Fig. 5. **Simulation** results of utilizing energy for learning the walking task. All three methods are applied with the proposed state-aware control space. The success rates of achieving a 20 m walk are displayed above the bar chart. Even with a reduced dimension dynamics model, MBRL with EED model is thus shown to be sufficient for walking task learning.

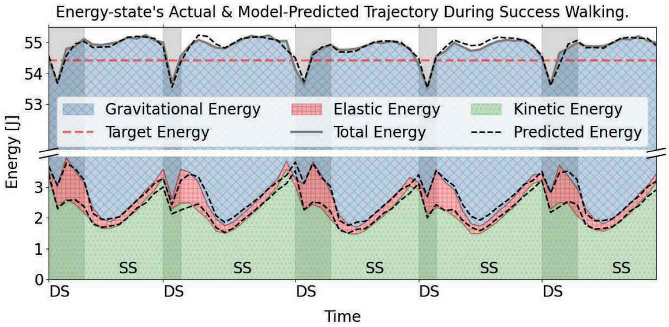

Fig. 6. Energy-state's actual and model-predicted (predicted mean) trajectory during successful walking in simulation, showing the exchange between gravitational, elastic, and kinetic energy, as well as the energy loss during the DS phase due to joint friction and damping. Our approach captures this energy loss, as shown in the model-predicted trajectory.

TABLE II
STATE DIMENSIONS (D) AND AVERAGE TIME PER PMPC OPTIMIZATION ITERATION FOR EACH BASELINE, WITH STANDARD DEVIATIONS IN PERCENTAGES

Instance	D	Average Time per pMPC Iteration	Speed Improvement
Ours	**9**	**0.58 ms** $\pm$**4.61%**	**1.71×**
TDE2-MBRL [19]	13	0.83 ms $\pm 2.73\%$	1.19×
Standard MBRL	16	0.99 ms $\pm 2.84\%$	1×

One iteration includes one H-horizon state prediction for optimiztion in (2).

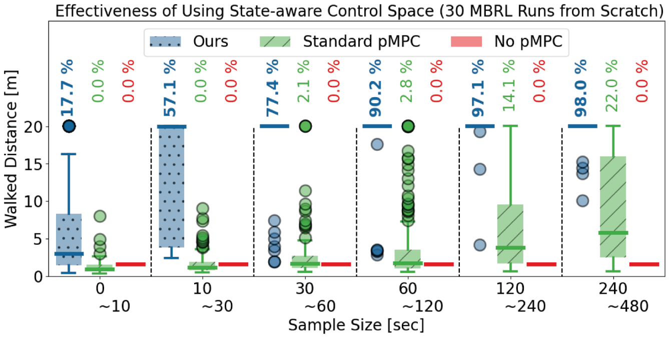

Fig. 7. **Simulation** results of constraining pMPC control space for learning the walking task. The success rates of achieving a 20 m walk are displayed above the bar chart. The increase in both learning efficiency and success rate confirms our approach's effectiveness in enhancing planning reliability.

V. EXPERIMENTAL RESULTS

A. Simulation Results

The target sample duration is set to 480 seconds, and we evaluate our proposed method by studying its 1) effectiveness of leveraging EED for learning biped walking, 2) effectiveness of constraining pMPC exploration with state-aware control space, 3) generalizability when walking at different velocities, and walking on uneven terrains.

1) Leveraging EED for Biped Walking: This section demonstrates the effectiveness of utilizing EED via comparing:

1) **Standard MBRL**, which learns the standard dynamics without energy terms, (31).
2) **TDE2-MBRL** [19], which only considered spring's elastic energy as energy-state, (32).
3) **Ours**, which considered both spring and actuator's motion as energy-state, (7).

All three methods are applied with the proposed state-aware control space. The state of standard MBRL is

$$\boldsymbol{x} = \left[\boldsymbol{\Theta}^\top, \theta_T, \dot{\boldsymbol{\Theta}}^\top, P_z, \dot{P}_R, \dot{P}_T, \boldsymbol{\Theta}_S\right]^\top \in \mathbb{R}^{16}, \tag{31}$$

where P_z is the z-element of robot's CoM position $\boldsymbol{P}_C$; and $\boldsymbol{\Theta}_S \in \mathbb{R}^4$ contains all four spring's deflection. Furthermore, the state of TDE2-MBRL is

$$\boldsymbol{x} = \left[\boldsymbol{\Theta}^\top, \theta_T, \dot{\boldsymbol{\Theta}}^\top, \boldsymbol{E}^\top\right]^\top \in \mathbb{R}^{13}. \tag{32}$$

Fig. 5 and Table II demonstrate that learning EED not only does not impede learning performance, but also reduces the time cost per pMPC optimization iterations, leading to higher success rates and highlighting the importance of increasing optimization iterations. In addition, Fig. 6 shows an example of energy-state trajectory and energy-exchange during successful walking with our approach.

2) Utilizing State-Aware Control Space: This section presents the effectiveness of constraining pMPC exploration. We compare the following three approaches:

1) **No pMPC**, where the theoretical velocities, obtained via (26)–(28), are directly taken as control input.
2) **Standard pMPC**, where the pMPC control space is set to the actuator's performance limit.
3) **Ours**, the proposed approach that pMPC explores an established control space around the theoretical velocities.

Fig. 7 shows that the "No pMPC" was unable to execute the walking task due to its inability to counteract joint elasticity. The "Standard pMPC" had a slow learning speed, as a result of a wide pMPC exploration range. Our approach, however, exhibited a remarkable improvement in both learning efficiency and walking reliability; 97.1% success rate was achieved after only two minutes of training samples. Also, Fig. 8 illustrates the swing leg motion of the proposed approach, demonstrating improved tracking capability in the later-stage ($\geq$240 s) with a much more concentrated trajectory compared to the early-stage ($\leq$30 s) results, which contributes to walking stability and a higher success rate shown in Fig. 7.

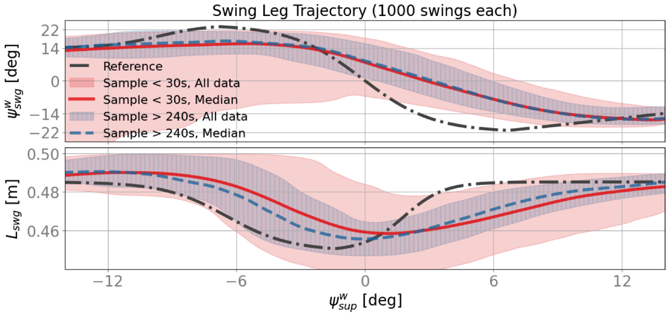

Fig. 8. Swing leg trajectory obtained through our approach in simulation (including fails), indicating that the tracking performance is enhanced with an increased number of samples collected.

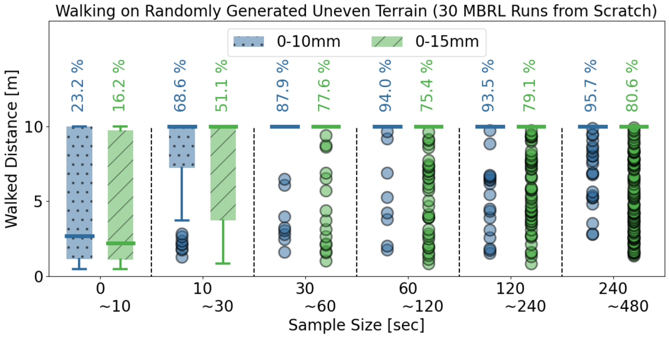

Fig. 9. **Simulation** results of of 30 MBRL attempts to walk on uneven terrains, showcasing the robustness of our approach. The success rates of achieving a 10 m walk are displayed above the bar chart.

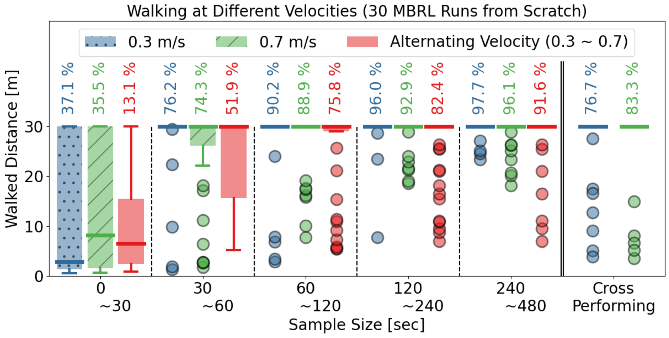

Fig. 10. **Simulation** results of 30 MBRL attempts to walk at different CoM's peak velocities. The "alternating velocity" switches between 0.3 and 0.7 m/s every five meters of walking. The "cross-performing" shows the average walking distance over five trials of using all 30 models learned with 0.3 m/s to perform 0.7 m/s walking and vice versa. The success rates of achieving a 30 m walk are displayed above the bar chart.

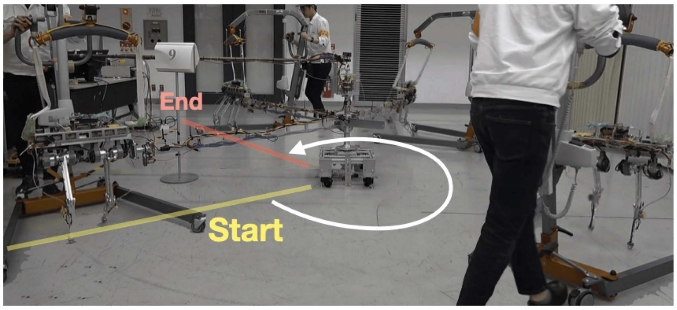

Fig. 11. Footage shots of hardware experiment.

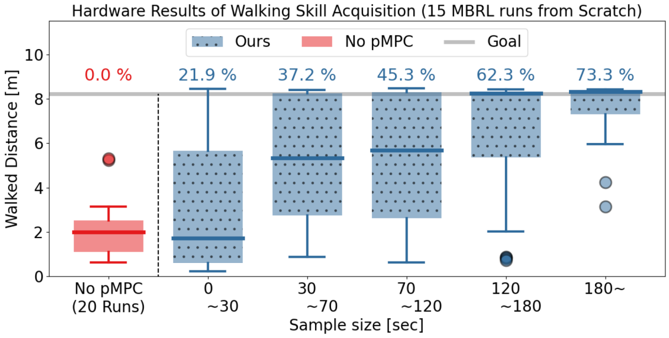

Fig. 12. **Hardware** results of our approach learning the walking task. The walked distance was obtained by combining the IMU's z-rotation readouts and the boom length. The success rate of achieving the 7.5 m goal distance (accounting for sensor errors) is displayed in percentages above each bar. The pMPC averaged 2.44 ms per iterations with the on-robot CPU.

3) Generalizability: Using the same framework and settings as in previous experiments, we tested our approach's generalizability of learning different tasks. Footage of the below experiments is provided in the supplementary movie.

First, we evaluate our approach's ability to learn robot dynamics under unknown terrains that have random ground heights ranging from 0–10 mm and 0–15 mm (2% and 3% leg length). Results in Fig. 9 shows our approached achieved over 95% and 80% success on the 10 mm and 15 mm terrains, respectively, demonstrating its ability to handle ground uncertainties without knowing the ground's status.

Second, we tested our approach's ability to walk at different speeds by changing the velocity of the CoM at peak $\boldsymbol{v}_p$ to 0.3 m/s, 0.7 m/s, and alternating between these two velocities every five meters to handle continuous speed changes. All other settings remained unchanged. The results (see Fig. 10) showed over 90% success across all three tests, demonstrating our approach's ability to generalize over different and changing walking speeds.

The results of the "cross-performing" test in Fig. 10 show that over 75% success was achieved when using all 30 models learned with 0.3 m/s walking speed to perform 0.7 m/s walking, and vice versa. This highlights the advantages of learning robot dynamics over learning robot tasks.

B. Hardware Results

We compared our approach with "No pMPC," which applies theoretical control input. Our goal was to learn the robot's dynamics from scratch and perform walking with 40 Hz online planning until the boom reaches 270° rotation ($\approx$8.23 m). The target sample duration for hardware is set to 180 seconds. As shown in Fig. 12, our approach achieved a 73.3% success rate in this walking task with only $\approx$180 seconds of samples, while "No pMPC" failed. This result demonstrates the real-world capability of our approach, even with a relatively low-spec CPU (i7-4700EQ) used for this experiment.

VI. Discussions

Despite the demonstrated success, our approach has limitations. The state-aware control space requires user adjustment to filter out unreliable controls without limiting the robot's capabilities, and control problems requiring both actuator positions and velocities cannot be applied if either is concealed. Nevertheless, this work provides insight into utilizing EED for compliant biped locomotion.

We listed the following future studies to improve this work. 1) Additional hardware experiments to validate that conditions in simulation experiment is reproducible in the real world. 2) While energy is a physical quantity that exists in all systems, we need to address the computational burden when implementing it in high-dimensional bipedal robots. 3) An adaptive model is required to handle scenarios where the zero point of potential energy changes, such as slops. 4) Developing a reference trajectory generator that considers stability will be helpful in achieving various dynamic maneuvers. 5) Investigating the ability to transfer models can improve the generalizability of handling changing environments.

VII. Conclusion

We propose an MBRL approach for learning the EED of a compliant biped robot and using pMPC to identify an optimal control for walking. The EED reduces dimensionality and effectively expresses the dynamics of a spring-loaded biped robot. Simulation results demonstrate our approach's ability to generalize across different speeds, changing speeds, and uneven terrains while learning quickly. Real-world feasibility is also demonstrated with hardware. Results show successful on-site learning using a compact dynamics model, 40 Hz real-time planning, and on-site learning within a few minutes.

Appendix A
Joint/Task/World Space Conversion

The following equations derive the conversion between local-(θ_1, θ_2), task- (ϕ, ψ, L) and world-space (ψ^w, θ_T):

$$\psi^w = \psi + \theta_T, \tag{33}$$

$$\phi = 0.5\left(\theta_{knee} - \theta_1 + \theta_2\right), \tag{34}$$

$$\psi = 0.5\left(\pi - \theta_{knee} - \theta_1 - \theta_2\right), \tag{35}$$

$$L = \left(l_{thigh} + l_{shin}\right)\cos\phi, \tag{36}$$

$$\dot{L} = \left(l_{thigh} + l_{shin}\right)\dot{\phi}\sin\phi, \tag{37}$$

$$\theta_i = \theta_i^{act} + \theta_i^{spr},\ \forall i = 1, 2, \tag{38}$$

where θ^{act} and θ^{spr} are the readouts of actuator and spring.

References

[1] T. Kamioka, H. Shin, R. Yamaguchi, and M. Muromachi, "Development and analysis of a biped robot with prismatic compliance," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 10398–10404.

[2] J.-K. Huang and J. W. Grizzle, "Efficient anytime CLF reactive planning system for a bipedal robot on undulating terrain," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2093–2110, Jun. 2023.

[3] R. Batke et al., "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots*, 2022, pp. 714–721.

[4] F. Yu, R. Batke, J. Dao, J. Hurst, K. Green, and A. Fern, "Dynamic bipedal turning through sim-to-real reinforcement learning," in *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots*, 2022, pp. 903–910.

[5] N. Csomay-Shanklin, M. Tucker, M. Dai, J. Reher, and A. D. Ames, "Learning controller gains on bipedal walking robots via user preferences," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 10405–10411.

[6] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Ann. Math. Artif. Intell.*, vol. 76, no. 1, pp. 5–23, 2016.

[7] K. Green, Y. Godse, J. Dao, R. L. Hatton, A. Fern, and J. Hurst, "Learning spring mass locomotion: Guiding policies with a reduced-order model," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3926–3932, Apr. 2021.

[8] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7309–7315.

[9] L. Yang, Z. Li, J. Zeng, and K. Sreenath, "Bayesian optimization meets hybrid zero dynamics: Safe parameter learning for bipedal locomotion control," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 10456–10462.

[10] Z. Li et al., "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 2811–2817.

[11] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based gaussian process filtering," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 225–232.

[12] S. Kamthe and M. P. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 1701–1710.

[13] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," in *Mach. Learn.* Norwell, MA, USA: Foundations and Trends, 2023, pp. 1–118.

[14] S. Levine and V. Koltun, "Learning complex neural network policies with trajectory optimization," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn.*, 2014, pp. II–829–II–837.

[15] J. Morimoto and C. Atkeson, "Minimax differential dynamic programming: An application to robust biped walking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 1539–1546.

[16] M. P. Deisenroth, R. Calandra, A. Seyfarth, and J. Peters, "Toward fast policy search for learning legged locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1787–1792.

[17] Y. Cui, S. Osaki, and T. Matsubara, "Autonomous boat driving system using sample-efficient model predictive control-based reinforcement learning approach," *J. Field Robot.*, vol. 38, no. 3, pp. 331–354, 2020.

[18] C.-Y. Kuo, Y. Cui, and T. Matsubara, "Sample-and-computation-efficient probabilistic model predictive control with random features," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 307–313.

[19] C.-Y. Kuo, H. Shin, T. Kamioka, and T. Matsubara, "TDE2-MBRL: Energy-exchange dynamics learning with task decomposition for spring-loaded bipedal robot locomotion," in *Proc. IEEE-RAS 22th Int. Conf. Humanoid Robots*, 2022, pp. 550–557.

[20] C.-Y. Kuo, A. Schaarschmidt, Y. Cui, T. Asfour, and T. Matsubara, "Uncertainty-aware contact-safe model-based reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3918–3925, Apr. 2021.

[21] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 465–472.

[22] G. Folkertsma and S. Stramigioli, "Energy in robotics," *Found. Trends Robot.*, vol. 6, no. 3, pp. 140–210, 2017.

[23] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, MA, USA: MIT Press, 2003.

[24] G. Zhao, M. Grimmer, and A. Seyfarth, "The mechanisms and mechanical energy of human gait initiation from the lower-limb joint level perspective," *Sci. Rep.*, vol. 11, Nov. 2021, Art. no. 22473.