2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Sample-Based Hybrid Mode Control: Asymptotically Optimal Switching of Algorithmic and Non-Differentiable Control Modes

Yilang Liu[1], Haoxiang You[1], Ian Abraham[1]

Abstract— **This paper investigates a sample-based solution to the hybrid mode control problem across non-differentiable and algorithmic hybrid modes. Our approach reasons about a set of hybrid control modes as an integer-based optimization problem where we select what mode to apply, when to switch to another mode, and the duration for which we are in a given control mode. A sample-based variation is derived to efficiently search the integer domain for optimal solutions. We find our formulation yields strong performance guarantees that can be applied to a number of robotics-related tasks. In addition, our approach is able to synthesize complex algorithms and policies to compound behaviors and achieve challenging tasks. Last, we demonstrate the effectiveness of our approach in a real-world robotic examples that requires reactive switching between long-term planning and high-frequency control. Videos are available on `https://yilangliu.github.io/hybrid_mode_sampling/`**

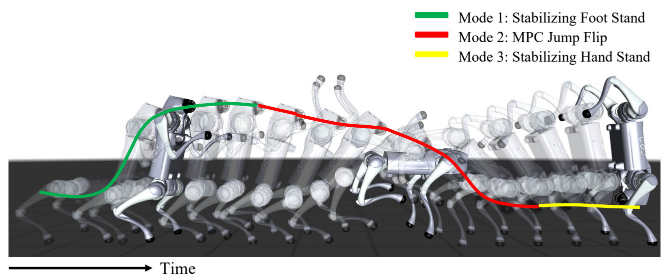

Fig. 1. **Sample-Based Hybrid Mode Switching Enables Extreme Control Variations.** We show the time-elapsed figure of the Unitree Go2 quadruped transitioning across three distinctive control modes (foot stand, jump flipping, and handstand) demonstrating that our method can achieve extremely agile motor skills within sharp transition times. The green, red, and yellow lines denote the head motion during each mode transitions.

I. INTRODUCTION

Modern agile robotic systems must dynamically switch between discrete modes—such as making and breaking contacts—to synthesize complex behaviors like locomotion and manipulation. Traditional continuous control methods struggle with these abrupt mode switches, often resulting in instability or suboptimal performance. These events often cause abrupt changes in dynamics and constraints, requiring either highly reactive control solutions or more algorithmic planning-based controllers that can handle contact-based reasoning. While it is possible to construct controllers within each of these distinct operating conditions, optimally transitioning between modes becomes challenging, especially when the underlying task requires multiple transitions with multiple hybrid modes.

Hybrid control theory offers principled methods that address these challenges by explicitly coordinating discrete mode switches with continuous control inputs. It specifically addresses the fact that many robotic systems undergo mode changes [1]–[3]. For instance, a legged robot system periodically transitions from stance to swing phases [4], while an in-hand manipulation task switches between grasping and free motion [5]. However, adapting hybrid control to arbitrary control modes, e.g., ones that solve algorithmic-based control, or require reasoning of contact dynamics becomes prohibitively challenging due to the combinatorial complexity of switching mode optimization [6], [7].

In this work, we propose a variation to the hybrid control problem that can handle algorithmic and non-differentiable control modes. Our approach formulates the hybrid control problem as a recursive integer-based search problem over the (1) the hybrid mode, (2) the discrete application time of the mode, and (3) the duration of the control mode. Since we operate in discrete-time the underlying search problem can be solved exactly through iterative search. We propose a sample-based version to uniformly sample the exhaustive set without replacement. We find that the sample-based approach has strong convergence guarantees and can find mode switching sequences without needing to reason about the composition of the specific mode. We compare against modern trajectory optimization techniques [8]–[10] and show that the proposed method is able to achieve significantly better performance by synchronizing more complex hybrid modes.

We also validate our approach on a real-world robotic platform, showcasing its practical effectiveness for demanding tasks that transition between stabilizing controllers through bridging of a model-predictive controller. Our contributions can be summarized as below:

1) A novel, iterative sample-based formulation of the hybrid control sequencing problem,
2) provable performance guarantees on optimizing mode sequencing, and
3) demonstration of complex mode switching between stabilizing controllers and mpc-based controllers in the real-world quadruped experiment.

The remainder of the paper is organized as follows: Section II provides related work, Section III introduces the hybrid control problem. Section IV describes our proposed sample-based hybrid mode control approach, Section V provides performance results of our proposed method, and

[1]Yilang Liu, Haoxiang You, and Ian Abraham are with the Department of Mechanical Engineering and Material Science, Yale University, 17 Hillhouse Avenue, New Haven, CT 06520, USA `yilang.liu@yale.edu; haoxiang.you@yale.edu; ian.abraham@sydney.edu.au`

last conclusion and limitation are presented in Section VI.

II. Related Work

A. Hybrid Control

The study of hybrid control systems addresses a canonical problem in robotic systems where dynamic switching between discrete modes, such as making or breaking contacts in locomotion or transitioning between manipulation phases [1], [2], [11], needs to be tightly scheduled. Hybrid control problems are typically solved by optimizing with switched dynamical systems [12], time [13], or controllers [14]. However, they struggle to scale to high-dimensional systems due to two key challenges: (1) the objective landscape is highly nonconvex [3], and (2) computational burden becomes intractable due to an increasing number of mode switches [15].

To mitigate these issues, prior works consider designing multiple simplified dynamics systems for high-dimensional hybrid control tasks [16]. For example, the quadruped dynamical systems are separated into front leg control and back leg control to ensure stable foot placements for the robot [17]. However, using a simplified dynamical system loses the possibility to fully exploit the whole-body capability. Other methods predefine the mode sequence in which contacts are made and broken. In this case, the optimization problem becomes easy to solve [18]. However, predefined modes constrain the robot's behavior to a predetermined trajectory.

In this work, we proposed a sampling-based approach for solving the hybrid control problem. Our method does not rely on pre-sequencing modes or simplified models and can synthesize complex and agile behaviors. Moreover, our method synthesizes extreme motor skills and conducts computations online.

B. Sample-based Control

Sample-based control methods [19]–[21] have recently emerged as a simple yet effective approach for solving high-dimensional robotics tasks such as legged locomotion [22]–[24] and manipulation [25]–[29]. Rather than relying on explicit gradient-based techniques, sample-based controls begin by sampling a control sequence from an initial distribution, executing a forward rollout of that sequence, and then adjusting the sampling distribution based on the resulting costs. This gradient-free nature makes sample-based control particularly well-suited for complex, non-differentiable systems where traditional optimization methods may struggle. Recent studies also show that stronger derivative modeling and stable value-learning formulations can directly improve control optimization behavior [30]–[32].

Despite the effectiveness, sample-based control methods exhibit two limitations. First, they often treat control at each timestep as an independent variable, ignoring the inherent hybrid structure of robotic tasks [33]. Second, the number of samples required to adequately explore the control space scales exponentially with the planning horizon. This exponential growth makes sample-based methods computationally infeasible for long-horizon tasks.

Our work unifies the strengths of hybrid control and sample-based optimization. By reparameterizing the key decision variables, i.e., *when* and *how long* to apply control, our approach can discover complex behaviors in contact-rich scenarios Additionally, the number of decision variables in our formulation is independent of time horizon, effectively reducing the search space for long-horizon tasks like in-hand manipulation. As a result, we effectively mitigate the exponential growth in sample requirements, making it feasible to handle both high-dimensional and long-horizon robotic tasks.

III. Problem Formulation: The (Indefinite) Hybrid Control Problem

In this section, we present a general formulation of the *indefinite* hybrid-mode switching control problem.

Definition 1: (Hybrid Mode System) Let the hybrid (autonomous) dynamical system with $M \in \mathbb{Z}_+$ modes be defined as

$$\dot{x} = f_m(x(t), t),\, m \in \mathcal{M} \tag{1}$$

where $\mathcal{M} = \{1, 2, \ldots, M\}$, $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, and $f_m : \mathcal{X} \times \mathbb{R}_+ \to \mathcal{T}_\mathcal{X}$ are (potentially non-differentiable or algorithmic) hybrid modes.

An example of non-differentiable and algorithmic hybrid modes are state transitions subject to open-loop model-predictive controller (e.g., model-predictive path integral control [8]), learned policies [34], or high-frequency whole-body controllers [35].

We interested in minimizing a performance metric $\mathcal{J} : \mathcal{X} \times [0, t_f] \to \mathbb{R}$ for some time window $t \in [0, t_f]$ by selecting a sequence of modes $\{m_1, \ldots m_i, \ldots m_I\}$ for $I < \infty$ instances that are applied at non-overlapping time $\tau_i \in [0, t_f]$ for a time duration of $\lambda_i \leq t_f - \tau_i$.

Definition 2: (Continuous-Time Hybrid Mode Switching Set) The set of non-overlapping switches is given as

$$\mathcal{T} = \Big\{ (m, \tau, \lambda)_i \mid \forall i \in \{1, \ldots, I\}, \\ \tau_i \in [0, t_f], \lambda_i \leq t_f - \tau_i, \tau_i + \lambda_i = \tau_{i+1} \Big\} \in \mathbb{M} \tag{2}$$

where the tuple $(m, \tau, \lambda)_i$ represent the i^{th} mode transition at time τ for a duration of λ, and $\mathbb{M}$ is the set of all possible mode transitions.

Given the definition for the set of non-overlapping switches and hybrid modes, we formalize the optimization problem to find the optimal switching set $\mathcal{T}^\star$.

Problem 1: (The Indefinite Hybrid Mode Switching Problem) Let $\mathcal{J} : \mathbb{M} \to \mathbb{R}$ be a performance metric and $\mathcal{M}$ be a set of valid hybrid modes. Then the set of mode transitions

$\mathcal{T} \in \mathbb{M}$ that minimizing $\mathcal{J}$ is given as

$$\min_{\mathcal{T}} \mathcal{J}(\mathcal{T}) = \int_{t=0}^{t_f} \ell(x(s))ds$$
$$\text{subject to}$$
$$\mathcal{T} = \Big\{(m, \tau, \lambda)_i \mid \forall i \in \{1, \dots, I\}, \quad \tau_i \in [0, t_f], \lambda_i \leq t_f - \tau_i, \tau_i + \lambda_i = \tau_{i+1}\Big\} \tag{3}$$
$$\dot{x} = f_m(x(t), t), x(0) = x_0, m_i \in \mathcal{M}$$

where $\ell : \mathcal{X} \to \mathbb{R}$ is a cost function that defines a high-level objective.

In general (3) is challenging to solve as (1) f_m needs to be at least continuous and differentiable[1], and (2) there can be an *indefinite* number of switches I which makes the problem intractable. Existing hybrid control methods generally focus on solving the indefinite switching through formulating variations of (3) by either defining a finite set of mode transitions (i.e., I is given) [37], or iteratively solving for single mode transitions one at a time and "stitching" together the switching mode set. More specifically, we define the single-mode hybrid control problem as a means to reconcile infinitely many switches into a single mode switching problem.

Problem 2: (The Single Switch Hybrid Mode Control Problem) Let $\mathcal{J} : \mathbb{M} \to \mathbb{R}$ be a performance metric and $\mathcal{T} \in \mathbb{M}$ be a nominal switching sequence. Then the set of single mode transition tuple (m, τ, λ) that minimizing $\mathcal{J}$ is given as

$$\min_{(m,\tau,\lambda)} \mathcal{J}((m, \tau, \lambda) \cup \mathcal{T}) = \int_{t=0}^{t_f} \ell(x(s))ds$$
$$\text{subject to}$$
$$\mathcal{T} = \Big\{(m, \tau, \lambda)_i \mid \forall i \in \{1, \dots, I\}, \quad \tau_i \in [0, t_f], \lambda_i \leq t_f - \tau_i, \tau_i + \lambda_i = \tau_{i+1}\Big\} \tag{4}$$
$$\dot{x} = f_m(x(t), t), x(0) = x_0, m_i \in \mathcal{M}$$

where the operation $(m, \tau, \lambda) \cup \mathcal{T}$ stitches the single mode transition tuple (m, τ, λ) with a given default switching mode set $\mathcal{T}$.

Iteratively solving (4) can eventually lead to solving (3) [36]; however, existing solutions are limited by the choice of hybrid mode which leads to performance gaps, minimal solution guarantees, and often requires heuristic-based mode switching [36]. Integrating and scheduling non-differentiable, and algorithmic hybrid modes has the potential to utilize more complex behaviors and reduce the performance gap, but scheduling remains an open challenge. In the following section, we present a potential solution to this problem via sample-based optimization.

IV. Sample-Based Hybrid Mode Control

We propose to solve (3) by first formulating a discrete-time variation of the indefinite hybrid mode-switching problem. Then, we show that one can iteratively solve for singular switching modes via sample-based optimization that yield strong asymptotic convergence guarantees.

A. The Discrete (Definite) Hybrid Control Problem

Let us define a discrete time index k where $t_k = k\Delta t$ is the time at the k^{th} index, and $\Delta t \in \mathbb{R}_+$ is a time discretization. Here, we choose Δt to be the slowest hybrid state mode transition f_m and

$$\begin{aligned} x_{k+1} &= F_m(x_k, k) \\ &= x(t_k + \Delta t) = x(t_k) + \int_{s=t_k}^{t_k+\Delta t} f_m(x(s), s)ds \end{aligned} \tag{5}$$

is the mode transition function. In general, f_m may not be continuous and F_m is solved via an algorithmic optimization (e.g., contact-based dynamics [38]). The idea is that the underlying control problem with state feedback will operate on some *fixed* frequency (i.e., discretized time). Thus, we abandon continuous-time formulation (which is limited in runtime computation) for a simpler formulation of mode scheduling in discrete-time. Next, we define the set of mode transition in discrete-time,

Definition 3: (Discrete-Time Hybrid Mode Switching Set) Let $T\Delta t = t_f$ be the discrete planning time and $\mu_i, \nu_i \in \mathbb{Z}_+$ are the discrete-time mode application and duration times respectively. Then the discrete-time hybrid mode switching set is given as

$$\mathcal{K} = \Big\{(m, \mu, \nu)_i \mid \forall i \in \{1, \dots, I\}, \quad \mu_i \in [0, T], \nu_i \in [0, T - \mu_i], \mu_i + \nu_i = \mu_{i+1}\Big\} \in \mathcal{M}^T \tag{6}$$

where $\mathcal{M}^T \subset \mathbb{Z}_+^T$ is the space of T hybrid mode sequences. [2]

Discretizing the performance metric $\mathcal{J}$ leads to the following mode-scheduling optimization problem.

Problem 3: (The Discrete-Time Hybrid Mode Switching Problem) Let $\mathcal{K} \in \mathcal{M}^T$ and $\mathcal{J} : \mathcal{M}^T \to \mathbb{R}$ be a discretized performance metric with $T\Delta t = t_f$ as the discretized planning time. Then an optimal mode sequence $\mathcal{K}^\star$ can be obtained by solving the following problem

$$\min_{\mathcal{K}} \mathcal{J}(\mathcal{K}) = \sum_{k=0}^{T} \ell(x_k)\Delta t$$
$$\text{subject to}$$
$$\mathcal{K} = \Big\{(m, \mu, \nu)_i \mid \forall i \in \{1, \dots, I\}, \quad \mu_i \in [0, T], \nu_i \in [0, T - \mu_i], \mu_i + \nu_i = \mu_{i+1}\Big\} . \tag{7}$$
$$x_{k+1} = F_m(x_k, k), x(0) = x_0, m_i \in \mathcal{M}$$

Note that there are at most T hybrid mode transitions which makes this problem definite with $\mathcal{K} = \{m_0, m_1, \dots, m_{T-1}\} \in \mathcal{M}^T$ being a vector containing T transition modes (that may repeat according to the discrete

[1] [36] presents a solutions for impacting systems, but does require gradient knowledge before and after the impact.

[2] The set here in discrete-time reduces to a sequence of modes m_i that starts at discrete time μ_i repeat according to ν_i or a vector of integer values, e.g., $\mathcal{K} = \{1, 4, 4, 4, 5, 2, 1, 1\}$ for $\mathcal{M} = [1, \dots, 5]$.

duration). In fact, since m, μ and ν are all *discrete integers*, the problem can be *exactly* solved through brute-force search that scales $\mathcal{O}(M^T)$. Exact solutions to (7) can be obtained with modern GPUs; however, a simpler recursive formulation can be derived which is used to inform us of formal performance guarantees.

B. Iterative Hybrid Mode Control Sequencing

In order to solve (7), we propose a iterative method rather than a brute-force approach that provides formal guarantees.

Problem 4: (Discrete-Time Single Switch Hybrid Mode Control Problem) Let us define a discrete default mode transition set $\mathcal{K}_{\text{def}} \in \mathcal{M}^T$. Then, the solution $(m, \mu, \nu)^\star$ that minimizes $\mathcal{J}$ is given as

$$(m, \mu, \nu)^\star = \underset{(m,\mu,\nu)}{\arg\min}\ \mathcal{J}((m, \mu, \nu) \cup \mathcal{K}_{\text{def}}) = \sum_{k=0}^{T} \ell(x_k)\Delta t$$

subject to

$$\mathcal{K}_{\text{def}} = \Big\{ m_i \forall i \in [0, T-1] \Big\}$$
$$x_{k+1} = F_m(x_k, k), x(0) = x_0, m_i \in \mathcal{M} \tag{8}$$

where $(m, \mu, \eta) \cup \mathcal{K}_{\text{def}}$ produces an updated default schedule $\mathcal{K}_{\text{def}}(k) = m \forall k \in [\mu, \mu + \eta]$.

Problem (7) can be solved through iterative refinement of $\mathcal{K}_{\text{def}}$ to obtain the optimal discrete-time mode transition set $\mathcal{K}^\star$. Here, (8) can be exactly solved via brute-force search with complexity $\mathcal{O}(MT(T+1)/2)$ which is much faster than $\mathcal{O}(M^T)$. We outline the iterative approach in Algorithm 1.

Assumption 1: There exists a sequence of modes modes $\mathcal{K}^\star = \{m_i\}_{i=1}^M$ such that $\mathcal{J}(\mathcal{K}^\star)$ is minimized.

Proposition 1: (Locally Optimal Schedules) Let $f_m \forall m \in \mathcal{M}$ be deterministic, $\mathcal{K}^\star$ the optimal switching mode schedule, and $\mathcal{J}(\mathcal{K}^\star)$ a local optima. Then, $\mathcal{K}$ is optimal if $\forall \mu \in [0, T-1]$ and $\forall m \in \mathcal{M}$, $\nexists \nu \in [1, T-\mu]$ such that

$$\mathcal{J}((m, \mu, \nu) \cup \mathcal{K}) \leq \mathcal{J}(\mathcal{K}), \tag{9}$$

$\mathcal{J}(\mathcal{K}) = \mathcal{J}(\mathcal{K}^\star)$, and $\mathcal{K} = \mathcal{K}^\star$ is a local optima.

Proof: Enumerating through each mode m and discrete application time μ and checking for all $\nu \in [1, T-\mu]$, if the cost does not reduce, then the only viable solution is to apply all modes for a duration $\nu = 0$ resulting in no reduction of the cost, and the schedule $\mathcal{K}$ is at a local optima. ■

Now that we know of the existence of local optima and a condition for which to check. We seek to prove that iteratively solving (8) leads to the local optimal solution.

Theorem 1: (Asymptotic Convergence) Let $\mathcal{J} : \mathcal{M}^T \to \mathbb{R}$, and $V_{(m,\mu,\eta)^\star} : \mathcal{M}^T \to \mathcal{M}^T$ where $V_{(m,\mu,\eta)^\star}(\mathcal{K}^k) = (m, \mu, \nu)^\star \cup \mathcal{K}^k = \mathcal{K}^{k+1}$ is an update equation to the k^{th} hybrid transition set with solution (8). Then,

$$\mathcal{J}(V_{(m,\mu,\eta)^\star}(\mathcal{K})) \leq J(\mathcal{K}) \tag{10}$$

and $\exists K \in \mathbb{Z}_+$ such that

$$V_{(m,\mu,\eta)^\star}(\mathcal{K}^{K+1}) = V_{(m,\mu,\eta)^\star}(\mathcal{K}^K) \tag{11}$$

is a fixed-point for $k \in [0, \ldots, K]$.

Proof: Using Proposition 1, if $\exists m, \mu$ where $\nu \neq 0$ in (8), then $\mathcal{J}(V_{(m,\mu,\eta)^\star}(\mathcal{K}^{k+1})) < J(\mathcal{K}^k)$. If $\nu = 0 \forall m, \mu$ is the minimizer to (8), then $(m, \mu, 0) \cup \mathcal{K} = \mathcal{K}$ and $V_{(m,\mu,\eta)^\star}(\mathcal{K}^{k+1}) = V_{(m,\mu,\eta)^\star}(\mathcal{K}^k)$ is a fixed point. ■

Because we discretized the underlying mode sequencing problem, Theorem 1 becomes possible to state as a performance guarantee. The main challenge is efficiently iterating through the solution space of (8) to exactly find the minimum $(m, \mu, \nu)^\star$. With a GPU, it is possible to iterate through the solution set; however, we present a more efficient sample-based approach that allows to iterate through the solution set without a GPU.

Algorithm 1 Iterative Discrete Hybrid Mode Sequencing

Initialize: control modes F_m for $m \in \mathcal{M}$, default mode sequence $\mathcal{K}_{\text{def}} = \{m_i \forall i \in [0, T-1]\}$, with m_i is any arbitrary mode, initial condition x_0, planning time T, cost ℓ, max iterations iter_{max},
iter $\leftarrow 0$
while iter $< \text{iter}_{\text{max}}$ or $\mathcal{J}((m, \mu, \nu) \cup \mathcal{K}_{\text{def}}) \leq \mathcal{J}(\mathcal{K}_{\text{def}})$ **do**
    $(m, \mu, \nu) \leftarrow$ Solve Problem 8
    if $\nu = 0$ **then**
        Terminate while loop
    end if
    $\mathcal{K}_{\text{def}}(k) = m \forall k \in [\mu, \mu + \nu]$
    iter $\leftarrow$ iter $+ 1$
end while

C. Sample-Based Hybrid Mode Control

Here, we present a variation of the solution to Problem (7) via a sample-based iteration of Problem (8). The rationale for a sampling method is that the search space can be significantly reduced from $\mathcal{O}(M \cdot T \cdot (T+1))$ to only requiring $\mathcal{O}(N)$ samples for N is the number of uniform samples to check a solution.

Assumption 2: (Uniqueness of Optimal Mode Transitions) Given a default mode transition set $\mathcal{K}$, there exists a unique single mode transition (m, μ, ν) that minimizes $\mathcal{J}((m, \mu, \nu) \cup \mathcal{K})$.

For real-valued states $x(t)$ with algorithmic modes F_m, Assumption 2 is valid so long as $\nu \neq 0$, in which case the solver is at a local optima.

Theorem 2: (Convergence of Sample-Based Single-Mode Hybrid Mode Control) Let $\Omega = \{(m, \mu, \eta) \forall m \in \mathcal{M}, \mu \in [0, T-1], \nu \in [0, T-\mu]\}$ be the set of all possible $Z = M \cdot T \cdot (T+1)/2$ single mode transitions. In addition, let $\mathcal{U}_N(\Omega)$ be a uniform distribution that draws $N < Z$ samples without replacement from the set Ω. Then the optimal mode transition tuple $(m, \mu, \nu)^\star$ to Problem (8) is found with probability $\mathcal{P}((m, \mu, \nu)^\star) = N/Z$ with at most $\frac{Z}{N}$ draws.

Proof: The probability of not finding the optimal single-mode transition is given by $\mathcal{P}(\neg(m, \mu, \nu)^\star) = \prod_{i=1}^{N} \frac{Z-i}{Z-i+1} = \frac{Z-N}{Z}$. Thus, $\mathcal{P}((m, \mu, \nu)^\star) = 1 - \frac{Z-N}{Z} = \frac{N}{Z}$ of drawing the optimal mode transition with at most $\frac{Z}{N}$ draws until $\mathcal{P}((m, \mu, \nu)^\star) = 1$. ■

In essence, the proposed sample-based approach takes advantage of the integer-based optimization that amounts to evaluating a list of tuples (the mode transition set). Indeed, there exists many ways to approach this problem that can be done by evaluating the list of mode transitions. Our proposed approach was motivated by the need to only evaluate a subsample of the mode transition set as needed. An outline of the sample-based single-mode hybrid control approach is provided in Algorithm 2.

Algorithm 2 Sample-Based Single-Mode Hybrid Mode Control

Initialize: nominal mode sequence $\mathcal{K}$, sample set Ω, N total number of samples, control modes F_m for $m \in \mathcal{M}$, objective $\mathcal{J}$, planning time T, initial condition x_0, max iterations $\text{iter}_{\text{max}} = Z/N$, $\mathcal{J}_{\text{best}} = \mathcal{J}(\mathcal{K}_{\text{def}})$.
iter $\leftarrow 0$
while iter $<$ iter_{max} **do**
  $\{(m, \mu, \nu)\}_{i=1}^{N} \sim \mathcal{U}_N(\Omega)$
  for $i = 1, \ldots, N$ **do**
    $\mathcal{K}_{\text{eval}} = \mathcal{K}_{\text{def}} \cup (m, \mu, \nu)$
    if $\mathcal{J}(\mathcal{K}_{\text{eval}}) \leq \mathcal{J}_{\text{best}}$ **then**
      return (m, μ, ν)
    end if
  end for
  $\Omega = \Omega \setminus \{(m, \mu, \nu)\}_{i=1}^{N}$
  iter $\leftarrow$ iter $+ 1$
end while

Algorithm 2 effectively solves Problem (7) by drawing N uniform samples from the set of single mode transitions Ω and evaluating the performance of the mode sequence stitched with $\mathcal{K}_{\text{def}}$. If a mode transition reduces the cost from the default, then we return the mode transition. Otherwise, we remove the drawn samples from the set Ω and resample. There is a scenario where we return a sampled mode (m, μ, ν) that reduces the cost, but another mode transition (m, μ, ν') for $\nu' \in [0, T - \mu]$ can further reduce the cost. In this case, we rely on Algorithm 1 to reinitialize Algorithm 2 to randomly search the set Ω once more to the improve the cost until $\nu = 0$ is the only valid option (i.e., the only mode transition that maintains the cost as stationary is applying no additional mode transition).

Through this sample-based approach, we are able to more efficiently search for hybrid mode sequences. As a result, we can optimize mode-sequences of complex behaviors on systems that experience non-differentiable contact, require mixing of algorithmic modes, and require composition of low-level behaviors to achieve far greater global behaviors than a single mode could. We illustrate our results in the following Section V.

V. Experiments and Results

We present both simulation and real-world results for our method, along with comparisons to other sampling-based approaches.

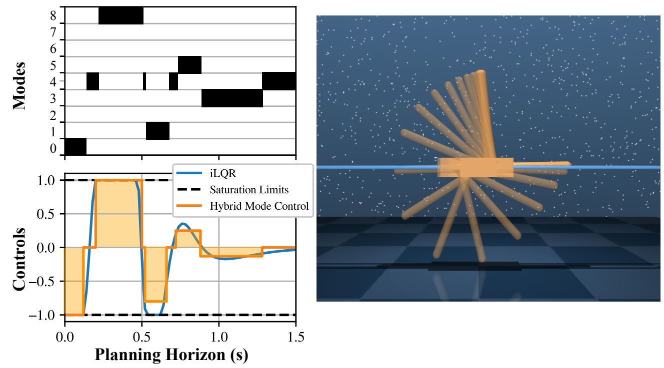

Fig. 2. **Top Left**: The optimal control mode sequences for the cartpole system, where the Y axis denotes the index for each corresponding sampled modes. **Bottom Left**: The resulting control applied to the actuators for the cartpole system are shown. The blue curve represents the control generated by the iLQR algorithm. The orange curve represents the executed control found using our sample-based hybrid scheduler, where the shaded areas denote the resulting distinguish modes. **Right**: The visualization of the cartpole trajectory with controls found by our method.

Toy Example: Cartpole Swing Up Here, we consider a cartpole swing-up task. We set the total horizon $T = 100$. The state is defined as pole angle θ, cart position p, pole angular velocity $\dot{\theta}$, and cart linear velocity $\dot{p}$. The control is the one-dimensional force applied horizontally to the cart. The pole is initialized in up-down $\theta = \frac{1}{2}\pi$, and the goal is to swing the pole up-right and maintain balance. Hence, the stage cost is defined as $c(\mathbf{x}_t, \mathbf{u}_t) = 4.0(\cos(\theta) - 1)^2 + 0.1p^2 + 0.1(\dot{\theta}^2 + \dot{p}^2) + \mathbf{u}^2$ and the terminal cost is defined as $c_f(\mathbf{x}_T) = 4.0(\cos(\theta) - 1)^2$. We choose the simple hybrid mode $\mathcal{M}$ as a set of the discrete controls bounded by the saturation limits. Specifically, controls are uniformly spaced between u_{min} and u_{max}

Figure 3 demonstrates the results for our method and other sampling approaches. We observe that our approach consistently finds optimal solutions across different horizons by actively considering time as a decision variable. In contrast, other sampling-based methods fail to identify good optima as the horizon increases, owing to the rapidly expanding search space. Therefore the performance of other methods further deteriorates with longer horizons, as their limited sample size prevents them from effectively solving the control problem. By comparison, our method continues to improve performance as the horizon increases.

We also compare the control sequence found by our method to the gradient-based iLQR method in Figure 2. The control sequence found by our method closely resembles the optimal iLQR sequence, even with a few modes.

A. Scalability to High-dimensional Task

Next, we investigate the scalability of our proposed approach to more complex and higher-dimensional control tasks. We start with simulation experiments to demonstrate scalability to high-dimensional tasks shown in Figure 1. The robot is tasked to stand on its hind legs and perform a flip directly to foreleg stance and balance. The task can be performed in three phases of locomotion, each of which requires distinct control strategies: stabilizing hind leg control policy,

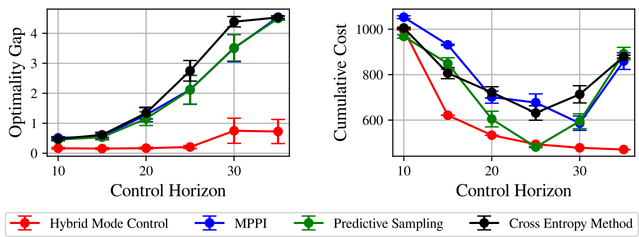

Fig. 3. **Top**: Results for the short-horizon subproblem of cartpole system. The Y-axis quantifies the optimality gap, defined as the difference between the objective value and a locally optimal solution obtained via iterative LQR. To ensure scale invariance across tasks, we normalize this gap by the planning horizon length H. Here error-bar denotes standard deviation across 5 random seeds. The performance of classical sampling methods deteriorates as the horizon length increases, whereas our method consistently finds good minima, even with a limited sample size. **Bottom**: Overall performance under a predictive control framework. Both methods show a decrease in cumulative cost as the horizon increases, as a longer horizon reduces myopic decision-making. However, as the horizon continues to grow, classical methods experience an increase in total cost due to their inability to effectively optimize the sub-problem, whereas our method continues to decrease the overall cost.

high speed predictive forward momentum swing towards the foreleg stance, and a stabilizing foreleg balancing control policy. We specifically choose this experiment because those three stages requires distinct control strategies and whole-body coordination.

Task Design: The quadruped example uses the Unitree Go2 robot for both simulation and hardware experiment. The dimension for quadruped state space $x_k \in \mathbb{R}^{48}$ and positional control inputs $u_k \in \mathbb{R}^{12}$. Here, state x_k contains the gravity projected in the body frame, base position, base linear and angular velocity, joint positions and velocities, and previous action. Control inputs u_k contain desired robot motor target angles in radians.

Hybrid Mode Design: We design three hybrid modes consisting of learning-based foot stand and hand stand controller, and a sampling-based model predictive controller (MPC). More modes can be added as long as they generate distinct motor skills. All three controllers share the same observation and action space. Specifically, our hybrid modes are defined as the following:

$$\mathcal{M} = \{F_f(x_k, k), F_h(x_k, k), F_j(x_k, k)\} \tag{12}$$

where F_f is the closed loop system under the foot stand policy $\pi_f(x_k)$, F_h is the closed loop system subject to the hand stand policy π_h, and $\pi_j(x_k)$ is the jump forward $F_j(x_k, k)$ mode is the closed-loop system driven by a model predictive controller. The loss function is the sum of the individual subgoals $\ell(x_t, u_t) = \sum_i \omega_i l_i$ defined in Table I:

For policy training process, we use PPO [39] as training backbone and adopt an asymmetric actor-critic settings [40], where the policy network and value network accept different observation inputs. Specifically, the policy network received the inputs mentioned earlier while the value network receive additional robot actuator forces and torso height. To effectively deploy the learned policy and model predictive control in simulation, we use Brax [41] since it can effectively combine policy inference and online sampling in the same control framework without converting data type. **Policy and Value network design**: The objective of the policy to maximize the expected return of a policy. Here, we flip the sign of the cost function to represent reward function for learning. Both networks use a three-layer Multilayer Perceptron (MLP) with hidden sizes of 512, 256, and 128. In between each hidden layers, we use Swish [42] as activation function. **MPC Design**: We adopt MPPI as backbone for sample-based model predictive control. The goal of the MPPI is to find best control perturbations of a future sequence of actions that best minimize the loss. We set planning horizon to be 20 steps forward and sample size to be 25. The temperature coefficient of MPPI is fixed to 0.1.

TABLE I

LOSS TERMS

Loss Term	Expression
Height Tracking	$l_{\text{h}} = 1.5 \cdot \|\|z^{tar} - z\|\|^2$
Orientation Tracking	$l_{\text{ori}} = 1.0 \cdot \|\|\phi_{\text{body,xy}} - \phi^{\text{tar}}_{\text{body,xy}}\|\|^2$
Joint Position	$l_q = 0.5 \cdot \|\|q - q_{\text{nominal}}\|\|^2$
Action Rate	$l_{\text{rate}} = 0.001 \cdot \|\|u_t - u_{t-1}\|\|^2$
Energy Consumption	$l_{\text{energy}} = 0.003 \cdot \|\|\dot{q} \cdot \tau\|\|$
Pose Deviation	$l_{\text{pose}} = 2.5e-7 \cdot \exp\left(-\|\|q - q_{\text{default}}\|\|^2\right)$

TABLE II

COMPARISON WITH EXISTING METHODS

	FootStand	JumpFlip	HandStand	Cost ↓
PPO-only	✓	×	×	34.03
MPPI-Only	✓	×	×	46.73
CEM-Only	✓	×	×	55.68
PS-Only	✓	×	×	48.39
Fixed Seq	✓	✓	×	22.24
Ours	✓	✓	✓	**13.519**

B. Comparison to baseline control methods.

We compare our approach against four baselines: (1) a unified policy network trained with PPO, referred to as PPO-Only; (2) pure model predictive control methods, including MPPI-Only [8], CEM-Only [10], and PS-Only (predictive sampling) [9]; (3) predefined hybrid mode sequences, denoted as Predefined Mode Seq; and (4) our sample-based hybrid mode control combined with MPPI, denoted as Hybrid Mode Control. We choose these baselines because alternative hybrid-control and hierarchical reinforcement-learning approaches, such as the Logical Options Framework (LOF) [43], are computationally expensive and require substantially more data, whereas our method composes complex locomotion behaviors from a simple policy-learning framework. For evaluation, we assess whether each method can solve the task fully or partially, as well as the cumulative cost, which each method aims to minimize. As shown in Table II, we conducted five trials for each methods and our methods can sccuessfully perform complete flip and balance

We observe that the robot falls quickly during the first transition period suggesting that solely using one control method is not enough to execute all motions. While training a single policy can successfully perform foot stand, the

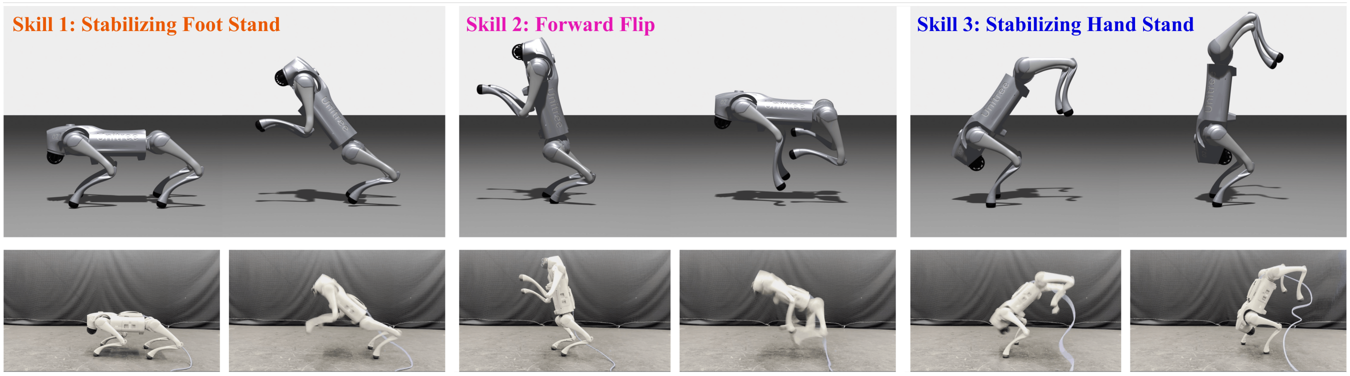

Fig. 4. We verify our method on the real Unitree Go2 quadruped. From left to right, the robot successfully transitions through foot-stand, jump-flip, and hand-stand behaviors. We update the control sequence in real time with onboard sensing and computation, demonstrating the efficiency of our method.

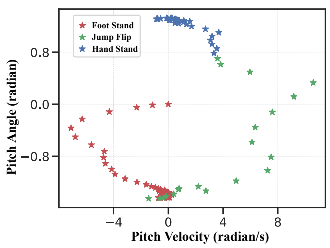

Fig. 5. **State Visitation of Robot Pitch Rotation:** We collect the trajectory robot motion and visualize the distribution of robot pitch angular position and velocity in the base frame. We show that our method can effective solving for control solutions that resides in distinct areas by optimizing control mode sequences.

policy quickly fall during the flip phase. We hypothesis that a single MLP policy network cannot capture this multi-modal behaviors. We also implement a hybrid mode control method similar to ours, but with predefined modes. The robot can safely transition to the flip phase but fails to adjust its pose into a handstand. In contrast, our approach is able to reason about the multi-modal nature of the objective and sequence the appropriate control modes. As shown in Fig. 5, each single-strategy controller learns to operate effectively only within a limited region of the state space. However, our method actively considers all single-strategy controllers and provides an effective way to generate more complex behaviors.

C. Hardware Experiments

We further validate our method in the real-world experiment using the quadrupedal robot Unitree Go2. As shown in Fig. 4, we successfully deploy our hybrid mode control method for agile foot stand, jump flip, and hand stand. The policy is converted to CPU runtime using Open Neural Network Exchange (ONNX) [44] to ensure real-time performance. We run the proposed method at 50 Hz on a single Intel i7-12700H CPU with 32 GB of memory. Additionally, we implement state estimation based on an extended Kalman filter using onboard sensing. In comparison to existing sample-based control framework [22], [45] relying on high accurate mocap system, our method only uses onboard sensing, demonstrating robustness under noisy state measurements.

VI. Conclusions

This work presents a sample-based hybrid mode control method inspired by hybrid control theory. Our approach offers an alternative modeling scheme for sampling within a predictive-control framework. By actively searching for control modes, application time, and duration as integer-based optimization, our method can effectively be applied to tasks that require complex control compositions. Moreover, by reasoning over algorithmic and learned control modes, our method can synthesize complex behavior for high-dimensional systems.

There are several limitations and future directions to consider. Sample-based control typically requires an accurate contact model—one can only simulate what is well-represented in that model. This reliance on a precise model constrains real-world applications, particularly in unstructured environments or scenarios where obtaining a reliable model is challenging. Future research directions include integrating our methods with data-driven approaches that do not require explicit modeling.

VII. Acknowledgments

This work is supported by the National Science Foundation under award NSF FRR 2238066. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. H. Park and H. Chung, "Hybrid control for biped robots using impedance control and computed-torque control," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, pp. 1365–1370 vol.2.

[2] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[3] A. R. Ansari and T. D. Murphey, "Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems," *IEEE Transactions on Robotics*, vol. 32, no. 5, p. 1196–1214, Oct. 2016. [Online]. Available: http://dx.doi.org/10.1109/TRO.2016.2596768

[4] S. Katayama and T. Ohtsuka, "Whole-body model predictive control with rigid contacts via online switching time optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8858–8865.

[5] R. S. Zarrin, R. Jitosho, and K. Yamane, "Hybrid learning-and model-based planning and control of in-hand manipulation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8720–8726.

[6] J. Carpentier and N. Mansard, “Multicontact locomotion of legged robots,” *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.

[7] Y. Liu, Y. Chen, H. Li, and J. Dong, “Design an augmentation exoskeleton to enhance lifting strength,” in *ASME International Mechanical Engineering Congress and Exposition*, vol. 84539. American Society of Mechanical Engineers, 2020, p. V006T06A022.

[8] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, “Information theoretic model predictive control: Theory and applications to autonomous driving,” 2017. [Online]. Available: https://arxiv.org/abs/1707.02342

[9] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, “Predictive sampling: Real-time behaviour synthesis with mujoco,” 2022. [Online]. Available: https://arxiv.org/abs/2212.00541

[10] S. Mannor, R. Rubinstein, and Y. Gat, “The cross entropy method for fast policy search,” *Proceedings, Twentieth International Conference on Machine Learning*, vol. 2, 07 2003.

[11] H. You, Y. Liu, and I. Abraham, “Accelerating visual-policy learning through parallel differentiable simulation,” 2025. [Online]. Available: https://arxiv.org/abs/2505.10646

[12] M. Egerstedt, Y. Wardi, and F. Delmotte, “Optimal control of switching times in switched dynamical systems,” in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 3, 2003, pp. 2138–2143 Vol.3.

[13] K. Flaßkamp, T. Murphey, and S. Ober-Blöbaum, “Discretized switching time optimization problems,” in *2013 European Control Conference (ECC)*, 2013, pp. 3179–3184.

[14] M. Egerstedt, X. Hu, and A. Stotsky, “A hybrid control approach to action coordination for mobile robots,” *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 617–622, 1999, 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017561054

[15] T. M. Caldwell and T. D. Murphey, “Projection-based switched system optimization: Absolute continuity of the line search,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 699–706.

[16] P. Wensing and D. Orin, “Improved computation of the humanoid centroidal dynamics and application for whole-body control,” *International Journal of Humanoid Robotics*, vol. 13, p. 1550039, 09 2015.

[17] R. Chipalkatty, H. Daepp, M. Egerstedt, and W. Book, “Human-in-the-loop: Mpc for shared control of a quadruped rescue robot,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4556–4561.

[18] P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Taïx, “C-croc: Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios,” *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 676–691, 2020.

[19] G. Williams, A. Aldrich, and E. Theodorou, “Model predictive path integral control using covariance variable importance sampling,” 2015. [Online]. Available: https://arxiv.org/abs/1509.01149

[20] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber, “Natural evolution strategies,” in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 3381–3387.

[21] S. Sirigiri, C. Hughes, I. Abraham, and F. Ramos, “Diversifying parallel ergodic search: A signature kernel evolution strategy,” in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=3XuUnUEI7e

[22] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, “Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing,” 2024. [Online]. Available: https://arxiv.org/abs/2409.15610

[23] H. You, Y. Liu, and I. Abraham, “Accelerating visual-policy learning through parallel differentiable simulation,” 2025. [Online]. Available: https://arxiv.org/abs/2505.10646

[24] Y. Liu and A. Barati Farimani, “An energy-saving snake locomotion pattern learned in a physically constrained environment with online model-based policy gradient method,” *Journal of Mechanisms and Robotics*, vol. 15, no. 4, p. 041007, 11 2022. [Online]. Available: https://doi.org/10.1115/1.4055167

[25] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, “Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.

[26] G. Rizzi, J. J. Chung, A. Gawel, L. Ott, M. Tognon, and R. Siegwart, “Robust sampling-based control of mobile manipulators for interaction with articulated objects,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1929–1946, 2023.

[27] I. Abraham, A. Handa, N. D. Ratliff, K. Lowrey, T. D. Murphey, and D. Fox, “Model-based generalization under parameter uncertainty using path integral control,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 2864–2871, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:212635299

[28] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. Ratliff, D. Fox, F. Ramos, and B. Boots, “Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation,” 2021. [Online]. Available: https://arxiv.org/abs/2104.13542

[29] Q. Wang, O. Abdellall, T. Gao, X. Sun, and D. Rakita, “Subsecond 3d mesh generation for robot manipulation,” 2025. [Online]. Available: https://arxiv.org/abs/2512.24428

[30] C. Liang, Q. Wang, A. Xu, and D. Rakita, “ad-trait: A fast and flexible automatic differentiation library in rust,” in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2025, pp. 1320–1326.

[31] D. Rakita, C. Liang, and Q. Wang, “Coherence-based approximate derivatives via web of affine spaces optimization,” 2025. [Online]. Available: https://arxiv.org/abs/2504.18790

[32] H. You, L. Molu, and I. Abraham, “Is bellman equation enough for learning control?” 2025. [Online]. Available: https://arxiv.org/abs/2503.02171

[33] L. Liu, K. Yin, M. Van de Panne, T. Shao, and W. Xu, “Sampling-based contact-rich motion control,” in *ACM SIGGRAPH 2010 papers*. Association for Computing Machinery, 2010, pp. 1–10.

[34] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” in *Conference on robot learning*. PMLR, 2022, pp. 91–100.

[35] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” 2019. [Online]. Available: https://arxiv.org/abs/1909.06586

[36] A. Ansari, K. Flaßkamp, and T. D. Murphey, “Sequential action control for tracking of free invariant manifolds,” pp. 335–342, 2015, analysis and Design of Hybrid Systems ADHS. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896315024556

[37] H. Chen, Z. Hong, S. Yang, P. M. Wensing, and W. Zhang, “Quadruped capturability and push recovery via a switched-systems characterization of dynamic balance,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2111–2130, 2023.

[38] N. J. Kong, C. Li, G. Council, and A. M. Johnson, “Hybrid ilqr model predictive control for contact implicit stabilization on legged robots,” *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4712–4727, 2023.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[40] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, “Asymmetric actor critic for image-based robot learning,” *Robotics: Science and Systems XIV*, 2018.

[41] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, “Brax - a differentiable physics engine for large scale rigid body simulation,” 2021. [Online]. Available: http://github.com/google/brax

[42] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” 2017. [Online]. Available: https://arxiv.org/abs/1710.05941

[43] B. Araki, X. Li, K. Vodrahalli, J. Decastro, M. Fry, and D. Rus, “The logical options framework,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 307–317. [Online]. Available: https://proceedings.mlr.press/v139/araki21a.html

[44] O. R. developers, “Onnx runtime,” https://onnxruntime.ai/, 2021, version: x.y.z.

[45] J. Alvarez-Padilla, J. Z. Zhang, S. Kwok, J. M. Dolan, and Z. Manchester, “Real-time whole-body control of legged robots with model-predictive path integral control,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 14 721–14 727.