

# Learning Controller Gains on Bipedal Walking Robots via User Preferences

Noel Csomay-Shanklin<sup>1</sup>, Maegan Tucker<sup>2</sup>, Min Dai<sup>2</sup>, Jenna Reher<sup>2</sup>, Aaron D. Ames<sup>1,2</sup>

**Abstract**—Experimental demonstration of complex robotic behaviors relies heavily on finding the correct controller gains. This painstaking process is often completed by a domain expert, requiring deep knowledge of the relationship between parameter values and the resulting behavior of the system. Even when such knowledge is possessed, it can take significant effort to navigate the nonintuitive landscape of possible parameter combinations. In this work, we explore the extent to which preference-based learning can be used to optimize controller gains online by repeatedly querying the user for their preferences. This general methodology is applied to two variants of control Lyapunov function based nonlinear controllers framed as quadratic programs, which provide theoretical guarantees but are challenging to realize in practice. These controllers are successfully demonstrated both on the planar underactuated biped, AMBER, and on the 3D underactuated biped, Cassie. We experimentally evaluate the performance of the learned controllers and show that the proposed method is repeatably able to learn gains that yield stable and robust locomotion.

## I. INTRODUCTION

Achieving robust and stable performance for physical robotic systems relies heavily on careful gain tuning, regardless of the implemented controller. Navigating the space of possible parameter combinations is a challenging endeavor, even for domain experts. To combat this challenge, researchers have developed systematic ways to tune gains for specific controller types [1]–[4]. For controllers where the input/output relationship between parameters and the resulting behavior is less clear, this can be prohibitively difficult. These difficulties are especially prevalent in the setting of bipedal locomotion, due to the extreme sensitivity of the stability of the system with respect to controller gains.

It was shown in [5] that control Lyapunov functions (CLFs) are capable of stabilizing locomotion through the hybrid zero dynamics (HZD) framework, with [6] demonstrating how this can be implemented as a quadratic program (QP), allowing the problem to be solved in a pointwise-optimal fashion even in the face of feasibility constraints. However, achieving robust walking behavior on physical bipeds can be an arduous process due to complexities such as compliance, under-actuation, and narrow domains of attraction. One such controller that has recently demonstrated stable locomotion on the 22 degree of freedom (DOF) Cassie biped, as shown in Fig. 1, is the ID-CLF-QP<sup>+</sup> [7].

This research was supported by NSF NRI award 1924526, NSF award 1932091, NSF CMMI award 1923239, NSF Graduate Research Fellowship No. DGE-1745301, and the Caltech Big Ideas and ZEITLIN Funds.

<sup>1</sup>Authors are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125.

<sup>2</sup>Authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125.

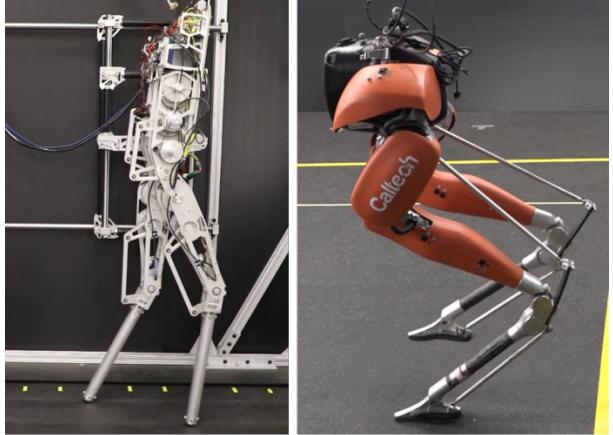


Fig. 1: The two experimental platforms investigated in this work: the planar AMBER-3M point-foot robot [8] (left), and the 3D Cassie robot [9] (right).

Synthesizing a controller capable of accounting for the challenges of underactuated locomotion, such as the ID-CLF-QP<sup>+</sup>, necessitates the addition of numerous control parameters, exacerbating the issue of gain tuning. Moreover, the relationship between the control parameters and the resulting behavior of the robot is extremely nonintuitive and results in a landscape that requires dedicated time to navigate, even for domain experts. Recently, machine learning techniques have been implemented to alleviate the process of hand-tuning gains in a controller agnostic way by systematically navigating the entire parameter space [10]–[12]. More specifically, Bayesian optimization techniques have been applied to learning gait parameters and controller gains for various bipedal systems [13], [14]. However, these techniques rely on a carefully constructed predefined reward function. Furthermore, it is often the case that different desired properties of the robotic behavior are conflicting and therefore can't be simultaneously optimized.

To alleviate the gain tuning process and enable the use of complicated controllers for naïve users, we employ a preference-based learning framework that only relies on subjective user feedback, mainly pairwise preferences, to systematically search the parameter space and realize stable and robust experimental walking. Preferences are a particularly useful feedback mechanism for parameter tuning because they are able to capture the notion of “general goodness” without a predefined reward function. This is particularly important for bipedal locomotion due to the lack of commonly agreed upon numerical metric of good or even stable walking in the community [15]–[18].

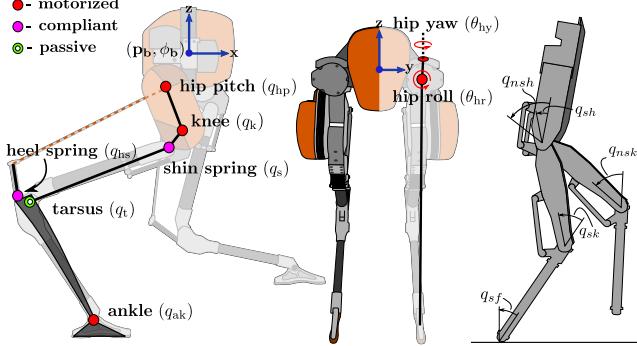


Fig. 2: Configuration of the 22 DOF (using an unpinned model) Cassie robot [9] (left) and configuration of the 5 DOF (using a pinned model) planar robot AMBER-3M [8] (right).

Preference-based learning has been previously used towards selecting essential constraints of an HZD gait generation framework which resulted in stable and robust experimental walking on a planar biped with unmodeled compliance at the ankle [19]. In this paper, we build on the previous work by exploring the application of preference-based learning towards implementing optimization-based controllers on multiple bipedal platforms. Specifically, we demonstrate the framework towards tuning gains of a CLF-QP<sup>+</sup> controller on the AMBER bipedal robot, as well as an ID-CLF-QP<sup>+</sup> controller on the Cassie bipedal robot, requiring the learning framework to operate in a much higher-dimensional space.

## II. PRELIMINARIES ON DYNAMICS AND CONTROL

### A. Modeling and Gait Generation

Following a floating-base convention [18], we define the configuration space as  $\mathcal{Q} \subset \mathbb{R}^n$ , where  $n$  is the unconstrained DOF (degrees of freedom). Let  $q = (p_b, \phi_b, q_l) \in \mathcal{Q} := \mathbb{R}^3 \times SO(3) \times \mathcal{Q}_l$ , where  $p_b$  is the global Cartesian position of the body fixed frame attached to the base linkage (the pelvis),  $\phi_b$  is its global orientation, and  $q_l \in \mathcal{Q}_l \subset \mathbb{R}^{n_l}$  are the local coordinates representing rotational joint angles. Further, the state space  $\mathcal{X} = T\mathcal{Q} \subset \mathbb{R}^{2n}$  has coordinates  $x = (q^\top, \dot{q}^\top)^\top$ . The robot is subject to various *holonomic constraints*, which can be summarized by an equality constraint  $h(q) \equiv 0$  where  $h(q) \in \mathbb{R}^h$ . Differentiating  $h(q)$  twice and applying D'Alembert's principle to the Euler-Lagrange equations for the constrained system, the dynamics can be written as:

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J(q)^\top \lambda \quad (1)$$

$$J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} = 0 \quad (2)$$

where  $D(q) \in \mathbb{R}^{n \times n}$  is the mass-inertia matrix,  $H(q, \dot{q})$  contains the Coriolis, gravity, and additional non-conservative forces,  $B \in \mathbb{R}^{n \times m}$  is the actuation matrix,  $J(q) \in \mathbb{R}^{h \times n}$  is the Jacobian matrix of the holonomic constraint, and  $\lambda \in \mathbb{R}^h$  is the constraint wrench. The system of equations (1) for the dynamics can also be written in control-affine form:

$$\dot{x} = \underbrace{\begin{bmatrix} \dot{q} \\ -D(q)^{-1}(H(q, \dot{q}) - J(q)^\top \lambda) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ D(q)^{-1}B \end{bmatrix}}_{g(x)} u.$$

The mappings  $f : T\mathcal{Q} \rightarrow \mathbb{R}^n$  and  $g : T\mathcal{Q} \rightarrow \mathbb{R}^{n \times m}$  are assumed to be locally Lipschitz continuous.

Dynamic and underactuated walking consists of periods of continuous motion followed by discrete impacts, which can be accurately modeled within a hybrid framework [20]. If we consider a bipedal robot undergoing domains of motion with only one foot in contact (either the left ( $L$ ) or right ( $R$ )), and domain transition triggered at footstrike, then we can define:

$$\mathcal{D}_{SS}^{\{L,R\}} = \{(q, \dot{q}) : p_{swf}^z(q) \geq 0\},$$

$$\mathcal{S}_{L \rightarrow R, R \rightarrow L} = \{(q, \dot{q}) : p_{swf}^z(q) = 0, \dot{p}_{swf}^z(q, \dot{q}) < 0\},$$

where  $p_{swf}^z : \mathcal{Q} \rightarrow \mathbb{R}$  is the vertical position of the swing foot, and  $\mathcal{D}_{SS}^{\{L,R\}}$  is the continuous domain on which our dynamics (1) evolve with a transition from one stance leg to the next triggered by the switching surface  $\mathcal{S}_{L \rightarrow R, R \rightarrow L}$ . When this domain transition is triggered, the robot undergoes an impact with the ground, yielding a hybrid model:

$$\mathcal{HC} = \begin{cases} \dot{x} = f(x) + g(x)u & x \notin \mathcal{S}_{L \rightarrow R, R \rightarrow L} \\ \dot{x}^+ = \Delta(x^-) & x \in \mathcal{S}_{L \rightarrow R, R \rightarrow L} \end{cases} \quad (3)$$

where  $\Delta$  is a plastic impact model [18] applied to the pre-impact states,  $x^-$ , such that the post-impact states,  $x^+$ , respect the holonomic constraints of the subsequent domain.

In this work, we design locomotion using the *hybrid zero dynamics* (HZD) framework [20] in order to generate stable periodic walking for underactuated bipeds. At the core of this method is the regulation of *virtual constraints*, or outputs:

$$y(x, \tau, \alpha) = y_a(x) - y_d(\tau, \alpha), \quad (4)$$

with the goal of driving  $y \rightarrow 0$  where  $y_a : T\mathcal{Q} \rightarrow \mathbb{R}^p$  and  $y_d : T\mathcal{Q} \times \mathbb{R} \times \mathbb{R}^a \rightarrow \mathbb{R}^p$  are smooth functions representing the actual and desired outputs, respectively,  $\tau$  is a phasing variable, and  $\alpha$  is a set of Bezièr polynomial coefficients that can be shaped to encode stable locomotion.

The desired outputs were optimized using the FROST toolbox [21], where stability of the gait was ensured in the sense of Poincaré via HZD theory [22]. This was done first for AMBER, in which one walking gait was designed using a pinned model of the robot [8], and then on Cassie for 3D locomotion using the motion library found in [23] consisting of 171 walking gaits for speeds in 0.1 m/s intervals on a grid for sagittal speeds of  $v_x \in [-0.6, 1.2]$  m/s and coronal speeds of  $v_y \in [-0.4, 0.4]$  m/s.

### B. Control Lyapunov Functions

Control Lyapunov functions (CLFs), and specifically rapidly exponentially stabilizing control Lyapunov functions (RES-CLFs), were introduced as methods for achieving (rapidly) exponential stability on walking robots [24]. This control approach has the benefit of yielding a control framework that can provably stabilize periodic orbits for hybrid system models of walking robots, and can be realized in a pointwise optimal fashion. In this work, we consider only outputs which are *vector relative degree* 2. Thus, differentiating (4) twice with respect to the dynamics results in:

$$\ddot{y}(x) = L_f^2 y(x) + L_g L_f y(x) u.$$

where  $L_f^2 y(x) : T\mathcal{Q} \rightarrow \mathbb{R}^p$  and  $L_g L_f y(x) : T\mathcal{Q} \rightarrow \mathbb{R}^p$  represent the Lie derivatives of the outputs with respect to the vector fields  $f(x)$  and  $g(x)$ . Assuming that the system is feedback linearizable, we can invert the decoupling matrix,  $L_g L_f y(x)$ , to construct a preliminary control input:

$$u = (L_g L_f y(x))^{-1} (\nu - L_f^2 y(x)), \quad (5)$$

which renders the output dynamics to be  $\ddot{y} = \nu$ . With the auxiliary input  $\nu$  appropriately chosen, the nonlinear system can be made exponentially stable. Assuming the preliminary controller (5) has been applied to our system, and defining  $\eta = [y, \dot{y}]^\top$  we have the following output dynamics [25]:

$$\dot{\eta} = \underbrace{\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}}_F \eta + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_G v. \quad (6)$$

With the goal of constructing a CLF using (6), we evaluate the continuous time algebraic Riccati equation (CARE):

$$F^\top P + PF + PGR^{-1}G^\top P + Q = 0, \quad (\text{CARE})$$

which has a solution  $P \succ 0$  for any  $Q = Q^\top \succ 0$  and  $R = R^\top \succ 0$ . From the solution of (CARE), we can construct a rapidly exponentially stabilizing CLF (RES-CLF) [24]:

$$V(\eta) = \eta^\top I_\varepsilon PI_\varepsilon \eta, \quad I_\varepsilon = \begin{bmatrix} \frac{1}{\varepsilon} I & 0 \\ 0 & I \end{bmatrix}, \quad (7)$$

where  $0 < \varepsilon < 1$  is a tunable parameter that drives the (rapidly) exponential convergence. Any feedback controller,  $u$ , which can satisfy the convergence condition:

$$\dot{V}(\eta) = L_f V(\eta) + L_g V(\eta) u \leq -\frac{1}{\varepsilon} \underbrace{\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}}_\gamma V(\eta), \quad (8)$$

will then render rapidly exponential stability for the output dynamics (4). To enforce (8), a quadratic program (CLF-QP) [6], with (8) as an inequality constraint can be posed.

Implementing this controller on physical systems, which are often subject to additional constraints such as torque bounds or friction limits, suggests that relaxation for the inequality constraint should be used. The introduction of relaxation and the need to reduce torque chatter on physical hardware lead to the following relaxed (CLF-QP) with incentivized convergence in the cost [26]:

#### CLF-QP<sup>+</sup>:

$$\begin{aligned} u^* &= \underset{u \in \mathbb{R}^m}{\operatorname{argmin}} \|L_f^2 y(x) + L_g L_f y(x) u\|^2 + w_V \dot{V}(x, u) \\ \text{s.t. } & u_{\min} \preceq u \preceq u_{\max} \end{aligned} \quad (9)$$

In order to avoid computationally expensive inversions of the model sensitive mass-inertia matrix, and to allow for a variety of costs and constraints to be implemented, a variant of the (CLF-QP) termed the (ID-CLF-QP) was introduced in [26]. This controller is used on the Cassie biped, with the decision variables  $\mathcal{X} = [\ddot{q}^\top, u^\top, \lambda^\top]^\top \in \mathbb{R}^{39}$ :

#### ID-CLF-QP<sup>+</sup>:

$$\mathcal{X}^* = \underset{\mathcal{X} \in \mathbb{X}_{ext}}{\operatorname{argmin}} \|A(x)\mathcal{X} - b(x)\|^2 + \dot{V}(q, \dot{q}, \ddot{q}) \quad (10)$$

$$\text{s.t. } D(q)\ddot{q} + H(q, \dot{q}) = Bu + J(q)^\top \lambda$$

$$u_{\min} \preceq u \preceq u_{\max}$$

$$\lambda \in \mathcal{AC}(\mathcal{X}) \quad (11)$$

where (2) has been moved into the cost terms  $A(x)$  and  $b(x)$  as a weighted soft constraint, in addition to a feedback linearizing cost, and a regularization for the nominal  $\mathcal{X}^*(\tau)$  from the HZD optimization. Interested readers are referred to [7], [26] for the full (ID-CLF-QP+) formulation.

#### C. Parameterization of CLF-QP

For the following discussion, let  $\mathbf{a} = [a_1, \dots, a_v] \in \mathbf{A} \subset \mathbb{R}^v$  be an element of a  $v$ -dimensional parameter space, termed an *action*. We let  $Q = Q(\mathbf{a})$ ,  $\varepsilon = \varepsilon(\mathbf{a})$ , and  $w_V = w_V(\mathbf{a})$  denote a parameterization of our control tuning variables, which will subsequently be learned. Each gain  $a_i$  for  $i = 1, \dots, v$  is discretized into  $d_i$  values, leading to an overall search space of actions given by the set  $\mathbf{A}$  with cardinality  $|\mathbf{A}| = \prod_{i=1}^v d_i$ . For the AMBER robot,  $v$  is taken to be 6 with discretizations  $d = [4, 4, 5, 5, 4, 5]$ , resulting in the following parameterization:

$$Q(\mathbf{a}) = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}, \quad Q_1 = \operatorname{diag}([a_1, a_2, a_2, a_1]), \\ \varepsilon(\mathbf{a}) = a_5, \quad w_V(\mathbf{a}) = a_6,$$

which satisfies  $Q(\mathbf{a}) \succ 0$ ,  $0 < \varepsilon(\mathbf{a}) < 1$ , and  $w_V(\mathbf{a}) > 0$  for the choice of bounds, as summarized in Table I. Because of the simplicity of AMBER, we were able to tune all associated gains for the CLF-QP<sup>+</sup> controller. For Cassie, however, the complexity of the ID-CLF-QP<sup>+</sup> controller warranted only a subset of parameters to be selected. Namely,  $v$  is taken to be 12 and  $d_i$  to be 8, resulting in:

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}, \quad Q_1 = \operatorname{diag}([a_1, \dots, a_{12}]), \\ Q_2 = \bar{Q},$$

with  $\bar{Q}$ ,  $\varepsilon$ , and  $w_V$  remaining fixed and predetermined by a domain expert. From this definition of  $Q$ , we can split our output coordinates  $\eta = (\eta_t, \eta_{nt})$  into *tuned* and *not-tuned* components, where  $\eta_t \in \mathbb{R}^{12}$  and  $\eta_{nt} \in \mathbb{R}^6$  correspond to the  $Q_1$  and  $Q_2$  blocks in  $Q$ .

TABLE I: Learned Parameters

CASSIE		
	Pos. Bounds	Vel. Bounds
$Q$ Pelvis Roll ( $\phi_x$ )	$a_1:[2000, 12000]$	$a_7:[5, 200]$
$Q$ Pelvis Pitch ( $\phi_y$ )	$a_2:[2000, 12000]$	$a_8:[5, 200]$
$Q$ Stance Leg Length ( $\ \phi^{st}\ _2$ )	$a_3:[4000, 15000]$	$a_9:[50, 500]$
$Q$ Swing Leg Length ( $\ \phi^{sw}\ _2$ )	$a_4:[4000, 20000]$	$a_{10}:[50, 500]$
$Q$ Swing Leg Angle ( $\theta_{hp}^{sw}$ )	$a_5:[1000, 10000]$	$a_{11}:[10, 200]$
$Q$ Swing Leg Roll ( $\theta_{hr}^{sw}$ )	$a_6:[1000, 8000]$	$a_{12}:[5, 150]$

AMBER			
	Pos. Bounds	Vel. Bounds	Bounds
$Q$ Knees	$a_1:[100, 1500]$	$a_3:[10, 300]$	$\varepsilon$
$Q$ Hips	$a_2:[100, 1500]$	$a_4:[10, 300]$	$w_V$
			$a_5:[0.08, 0.2]$
			$a_6:[1, 5]$

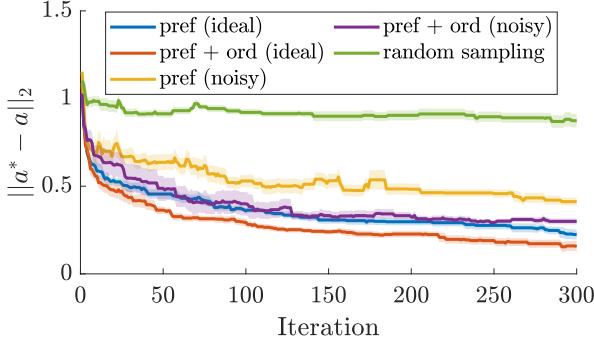


Fig. 3: Simulated results averaged over 10 runs, demonstrating the capability of preference-based learning to optimize over large action spaces, specifically the one used for experiments with Cassie. Shaded region depicts standard error.

### III. LEARNING FRAMEWORK

The preference-based learning framework leveraged in this work is a slight extension of that presented in [27]. Specifically, this work implements ordinal labels as an additional feedback mechanism to improve sample-efficiency. As in [27], the algorithm is aimed at regret minimization, defined as sampling  $N$  actions  $\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$  such that:

$$\{\mathbf{a}_1, \dots, \mathbf{a}_N\} = \operatorname{argmin}_{\mathbf{a} \in \mathbf{A}} \sum_{i=1}^N (U(\mathbf{a}^*) - U(\mathbf{a}_i)),$$

where  $\mathbf{A} \subset \mathbb{R}^{|\mathbf{a}|}$  is the discretized set of all possible actions,  $U : \mathbf{A} \rightarrow \mathbb{R}$  is the underlying utility function of the human operator mapping each action to a subjective measure of “good”, and  $\mathbf{a}^*$  is the action maximizing  $U$ . This objective of regret minimization can be equivalently interpreted as trying to sample actions with high utilities,  $\bar{\mathbf{a}}^* = \operatorname{argmax}_{\mathbf{a} \in \mathbf{A}} U(\mathbf{a})$ , in as few iterations as possible. In this section, we briefly outline the learning framework and how it was modified for our application.

#### A. Summary of Learning Method

In each iteration, the user is queried for their preference between the most recently sampled action,  $\mathbf{a}_i$ , and the previous action,  $\mathbf{a}_{i-1}$ , denoted as  $\mathbf{a}_i \succ \mathbf{a}_{i-1}$  if action  $\mathbf{a}_i$  is preferred. This preference is modeled as:

$$\mathcal{P}(\mathbf{a}_i \succ \mathbf{a}_{i-1} | U(\mathbf{a}_i), U(\mathbf{a}_{i-1})) = \phi\left(\frac{U(\mathbf{a}_i) - U(\mathbf{a}_{i-1})}{c_p}\right),$$

where  $\phi : \mathbb{R} \rightarrow [0, 1]$  is a monotonically-increasing link function, and  $c_p > 0$  represents the amount of noise expected in the preferences. In this work, we select the heavy-tailed sigmoid distribution  $\phi(x) := \frac{1}{1+e^{-x}}$ .

Inspired by [28], we supplement preference feedback with ordinal labels. Because ordinal labels are expected to be noisy, the ordinal categories are limited to only “very bad”, “neutral”, and “very good”. Ordinal labels are obtained each iteration for the corresponding action  $\mathbf{a}_i$  and are assumed to be assigned based on  $U(\mathbf{a}_i)$ . Similar to preferences, these ordinal labels are modeled using a likelihood function:

$$\mathcal{P}(o = r | U(\mathbf{a}_i)) = \phi\left(\frac{b_r - U(\mathbf{a}_i)}{c_o}\right) - \phi\left(\frac{b_{r-1} - U(\mathbf{a}_i)}{c_o}\right),$$

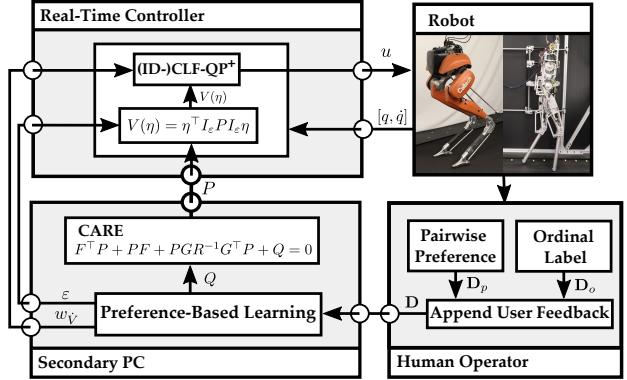


Fig. 4: The experimental procedure, notably the communication between the controller, physical robot, human operator, and learning framework.

where  $o$  denotes the ordinal label provided by the user with a corresponding ordered ranking  $r \in \{1, 2, 3\}$ ,  $c_0 > 0$  denotes expected noise in the ordinal labels, and  $\{b_0, \dots, b_3\}$  are arbitrary thresholds that dictate which latent utility ranges correspond to which ordinal label.

In each iteration, operator feedback is obtained and appended to the preference and ordinal label datasets  $\mathbf{D}_p$  and  $\mathbf{D}_o$ , with all feedback denoted as  $\mathbf{D} = \mathbf{D}_p \cup \mathbf{D}_o$ . This feedback is then used to approximate the posterior distribution  $\mathcal{P}(\mathbf{U} | \mathbf{D})$  as a multivariate Gaussian  $\mathcal{N}(\mu, \Sigma)$  via the Laplace approximation as in [29, Sec. 2.3]. To remain tractable in high-dimensions,  $\mathbf{U}$  is a restriction of  $U$  as defined in [27]. The predictive distribution at location  $\hat{\mathbf{a}}$  is described by a univariate Gaussian  $U(\hat{\mathbf{a}}) \sim \mathcal{N}(\mu(\hat{\mathbf{a}}), \Sigma(\hat{\mathbf{a}}))$ , whose equations can be found in [29, Sec. 2.3].

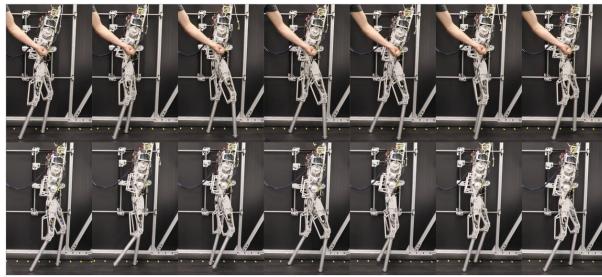
To select new actions to query in each iteration, Thompson sampling [30] is used. Specifically, at each iteration, a function  $\hat{U}$  randomly drawn from the Gaussian process is maximized. This iterative process of (1) querying the operator for feedback, (2) modeling the underlying utility function, and (3) sampling new actions, is repeated in each subsequent iteration. Finally, the best action after the completion of the experiment is given by  $\hat{\mathbf{a}}^* = \operatorname{argmax}_{\mathbf{a} \in \mathbf{A}} \mu(\mathbf{a})$ .

#### B. Expected Learning Behavior

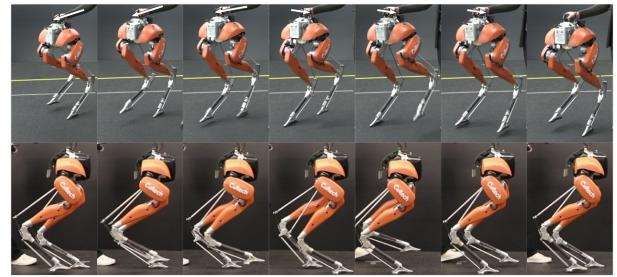
To demonstrate the learning, a simple example was constructed of the same dimensionality as the parameter space being investigated on Cassie ( $v = 12, d = 8$ ), where the utility was modeled as  $U(\mathbf{a}) = \|\mathbf{a} - \mathbf{a}^*\|_2$  for some  $\mathbf{a}^*$ . Feedback was automatically generated for both ideal noise-free feedback as well as for noisy feedback (correct feedback given with probability 0.9). The results of the simulated algorithm, illustrated in Fig. 3, show that the learning framework quickly samples actions near  $\mathbf{a}^*$ , even for an action space as large as the one used in the experiments with Cassie. The simulated results also show that ordinal labels improve convergence, motivating their use in the final experiment.

TABLE II: Learned Parameters

AMBER	[750, 100, 300, 100, 0.125, 2]
Cassie	[2400, 1700, 4200, 5600, 1700, 1200, 27, 40, 120, 56, 17, 7]

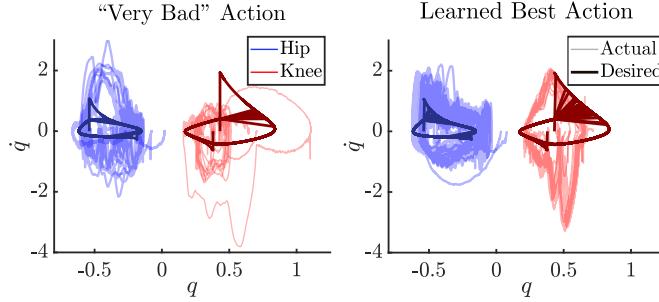


(a) Very low utility (top) where the robot was unable to walk unassisted and maximum posterior utility (bottom) where stable walking was achieved.



(b) The robustness (top) and tracking (bottom) of the walking with the learned optimal gains, enabling unassisted, stable walking and good tracking performance.

Fig. 5: Gait tiles for AMBER (left) and Cassie (right).



(a) Phase portraits for AMBER experiments.

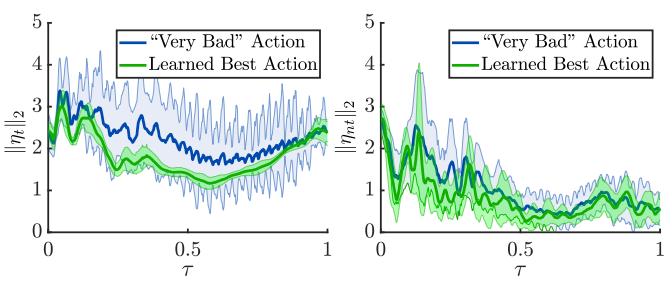
Fig. 6: Experimental walking behavior of the CLF-QP<sup>+</sup>

#### IV. LEARNING TO WALK IN EXPERIMENTS

Preference-based learning was applied to the realization of optimization-based control on two separate robotic platforms: the 5 DOF planar biped AMBER, and the 22 DOF 3D biped Cassie, as can be seen in the video [31]. As illustrated in Fig. 4, the experimental procedure had four main components: the physical robot (either AMBER or Cassie), the controller running on a real-time PC, a human operator providing feedback, and a secondary PC running the learning algorithm. Each action was tested for approximately one minute, during which the behavior of the robot was evaluated in terms of both performance and robustness. User feedback in the form of pairwise preferences and ordinal labels was obtained after testing each action via the respective questions: “Do you prefer this behavior more or less than the last behavior”, and “Would you give this gait a label of very bad, neutral, or very good”. After user feedback was collected for the sampled controller gains, the posterior was inferred over all of the uniquely sampled actions, which took up to 0.5 seconds. The experiment with AMBER was conducted for 50 iterations, lasting one hour, and the experiment with Cassie was conducted for 100 iterations, lasting two hours. The duration of the experiments was scaled based on the size of the respective action spaces, and trials were terminated when satisfactory behaviors had been sampled.

##### A. Results with AMBER – CLF-QP<sup>+</sup>

The CLF-QP<sup>+</sup> controller was implemented on an off-board i7-6700HQ CPU @ 2.6GHz with 16 GB RAM, which solved for desired torques and communicated them with the



(b) Output Error of  $\eta_t$  (left) and  $\eta_{nt}$  (right) for Cassie experiment.

(left) and the ID-CLF-QP<sup>+</sup> (right) with the learned gains.

ELMO motor drivers on the AMBER robot at 2kHz. During the first half of the experiment, the algorithm sampled a variety of gains causing behavior ranging from instantaneous torque chatter to induced tripping due to inferior output tracking. It is important to note that none of the initial sampled values led to unassisted walking. By the end of the experiment however, the algorithm had sampled 3 gains which were deemed “very good”, and which resulted in stable walking. The final learned best actions found by the algorithm are reported in Table II. Gait tiles for an action deemed “very bad”, as well as the learned best action are shown in Fig. 5a. Additionally, tracking performance for the two sets of gains is seen in Fig. 6a, where the learned best action tracks the desired behavior to a better degree.

##### B. Results with Cassie – ID-CLF-QP<sup>+</sup>

The ID-CLF-QP<sup>+</sup> controller was implemented on the on-board Intel NUC computer, which was running a PRE-EMPT\_RT kernel. The software runs on two ROS nodes, one of which communicate state information and joint torques over UDP to the Simulink Real-Time xPC, and one of which runs the controller. Each node is given a separate core on the CPU, and is elevated to real-time priority. Preference-based learning was run on an external computer and was connected to the ROS master over WiFi. Actions were updated in real-time; once an action was selected, it was sent to Cassie via a rosservice call, where, upon receipt, the robot immediately updated the corresponding gains. As rosservice calls are blocking, multithreading their receipt and parsing was necessary in order to maintain real-time performance.

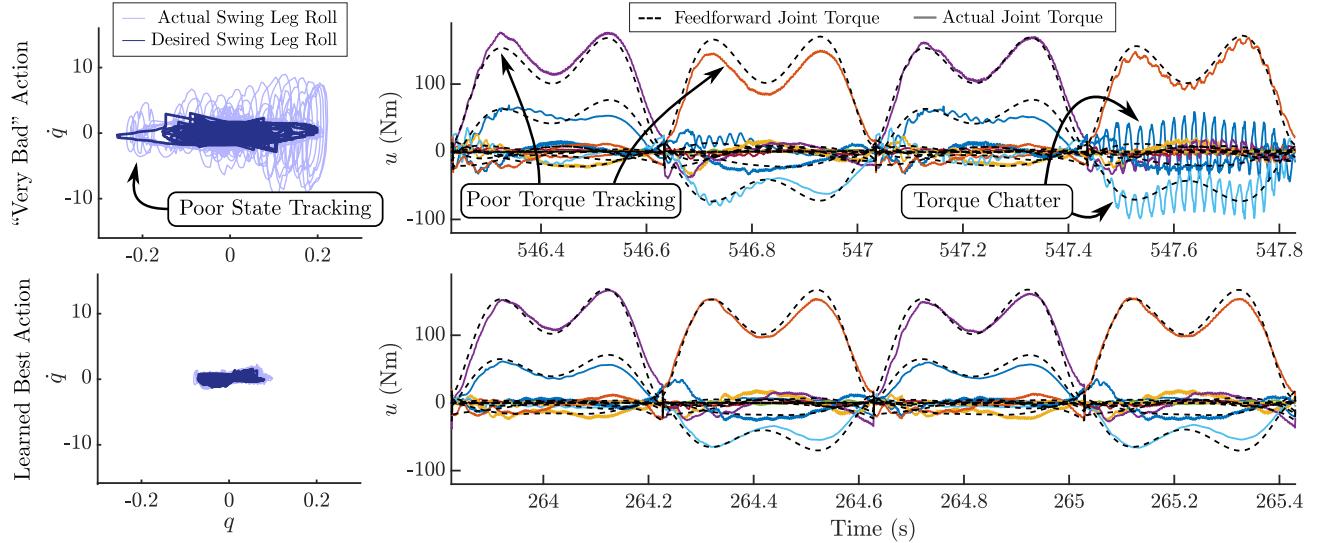


Fig. 7: Phase plots and torques commanded by the ID-CLF-QP<sup>+</sup> in the naïve user experiments with Cassie. For torques, each colored line corresponds to a different joint, with the black dotted lines being the feedforward torque. The gains corresponding to a “very bad” action (top) yield torques that exhibit poor tracking on joints and torque chatter. On the other hand, the gains corresponding to the learned optimal action (bottom) exhibit much better tracking and no torque chatter.

To demonstrate repeatability, the experiment was conducted twice on Cassie: once with a domain expert, and once with a naïve user. In both experiments, a subset of the  $Q$  matrix from (CARE) was tuned with coarse bounds given by a domain export, as reported in Table I. These specific outputs were chosen because they were deemed to have a large impact on the performance of the controller. Some metrics used to determine preferences were the following: no torque chatter, no drift in the floating base frame, responsiveness to desired directional input, no violent impacts, no harsh noise, and naturalness of walking. At the start of the experiments, there was significant torque chatter and wandering, with the user having to regularly intervene to recenter the global frame. As the experiments continued, the walking noticeably improved. At the conclusion of 100 iterations, the posterior was inferred over all uniquely visited actions. The action corresponding with the maximum utility – believed by the algorithm to result in the most user preferred walking behavior – was further evaluated for tracking and robustness. In the end, this learned best action coincided with the walking behavior that the user preferred the most.

Features of this optimal action, compared to a worse action sampled in the beginning of the experiments, are outlined in Fig. 6. In terms of quantifiable improvement, the difference in tracking performance is shown in Fig. 6b. The magnitude of the tuned parameters,  $\eta_t$ , illustrates the improvement that preference-based learning attained in tracking the outputs it intended to. At the same time, the tracking error of the constant parameters,  $\eta_{nt}$ , shows that the outputs that were not tuned remained unaffected by the learning process. This quantifiable improvement is further illustrated by the commanded torques in Fig. 7, which show that the optimal gains result in much less torque chatter and better tracking as compared to the other gains.

### C. Limitations and Future Work

The main limitation of the current formulation of preference-based learning is that the action space must be predefined with set bounds. In the context of controller gains, these bounds are difficult to know *a priori* since the relationship between the gains and the resulting behavior is unpredictable. Future work to address this problem involves modifications to the learning framework to shift action space based on the user’s preferences. Furthermore, the current framework limits the set of potential new actions to the set of actions discretized by  $d_i$  for each dimension  $i$ . As such, future work also includes adapting the granularity of the action space based on the uncertainty in specific regions.

### V. CONCLUSION

Navigating the complex landscape of controller gains is a challenging process that often requires significant knowledge and expertise. In this work, we demonstrated that preference-based learning is an effective mechanism towards systematically exploring a high-dimensional controller parameter space, without needing to define an objective function. Furthermore, we experimentally demonstrated the power of this method on two different platforms with two different controllers, showing the application agnostic nature of the framework. In all experiments, the robots went from stumbling to walking in a matter of hours. Additionally, the learned best gains in both experiments corresponded with the walking trials most preferred by the human operator. In the end, the robots had improved tracking performance, and were robust to external disturbance. Future work includes addressing the aforementioned limitations, extending this methodology to other robotic platforms, coupling preference-based learning with metric-based optimization techniques, and addressing multi-layered parameter tuning tasks.

## REFERENCES

- [1] W. K. Ho, C. C. Hang, and L. S. Cao, “Tuning of PID controllers based on gain and phase margin specifications,” *Automatica*, vol. 31, no. 3, pp. 497–502, 1995.
- [2] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta, “Practical approach to tuning MPC,” *ISA transactions*, vol. 42, no. 1, pp. 149–162, 2003.
- [3] L. Zheng, “A practical guide to tune of proportional and integral (PI) like fuzzy controllers,” in *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*. IEEE, 1992, pp. 633–640.
- [4] H. Hjalmarsson and T. Birkeland, “Iterative feedback tuning of linear time-invariant MIMO systems,” in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, vol. 4. IEEE, 1998, pp. 3893–3898.
- [5] A. D. Ames and M. Powell, “Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs,” in *Control of Cyber-Physical Systems*. Springer, 2013, pp. 219–240.
- [6] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, “Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs,” *IEEE Access*, vol. 3, pp. 323–332, 2015.
- [7] J. Reher and A. Ames, “Control Lyapunov functions for compliant hybrid zero dynamic walking,” *ArXiv, Submitted to IEEE Transactions on Robotics*, vol. abs/2107.04241, 2021.
- [8] E. Ambrose, W.-L. Ma, C. Hubicki, and A. D. Ames, “Toward benchmarking locomotion economy across design configurations on the modular robot: AMBER-3M,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 1270–1276.
- [9] A. Robotics, <https://www.agilityrobotics.com/robots#cassie>, Last accessed on 2021-09-14.
- [10] M. Birattari and J. Kacprzyk, *Tuning metaheuristics: a machine learning perspective*. Springer, 2009, vol. 197.
- [11] M. Jun and M. G. Safonov, “Automatic PID tuning: An application of unfalsified control,” in *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No. 99TH8404)*. IEEE, 1999, pp. 328–333.
- [12] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, “Automatic LQR tuning based on Gaussian process global optimization,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 270–277.
- [13] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, “Bayesian optimization for learning gaits under uncertainty,” *Annals of Mathematics and Artificial Intelligence*, vol. 76, no. 1, pp. 5–23, 2016.
- [14] A. Rai, R. Antonova, S. Song, W. Martin, H. Geyer, and C. Atkeson, “Bayesian optimization using domain knowledge on the ATRIAS biped,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1771–1778.
- [15] P.-B. Wieber, “On the stability of walking systems,” in *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002.
- [16] M. Vukobratović and B. Borovac, “Zero-moment point—thirty five years of its life,” *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.
- [17] J. E. Pratt and R. Tedrake, “Velocity-based stability margins for fast bipedal walking,” in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 299–324.
- [18] J. W. Grizzle, C. Chevallereau, R. W. Sinnott, and A. D. Ames, “Models, feedback control, and open problems of 3D bipedal robotic walking,” *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.
- [19] M. Tucker, N. Csomay-Shanklin, W.-L. Ma, and A. D. Ames, “Preference-based learning for user-guided HZD gait generation on bipedal walking robots,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2804–2810.
- [20] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.
- [21] A. Hereid and A. D. Ames, “FROST: Fast robot optimization and simulation toolkit,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 719–726.
- [22] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, “Hybrid zero dynamics of planar biped walkers,” *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 42–56, 2003.
- [23] J. Reher and A. D. Ames, “Inverse dynamics control of compliant hybrid zero dynamic walking,” 2020.
- [24] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.
- [25] A. Isidori, *Nonlinear Control Systems, Third Edition*, ser. Communications and Control Engineering. Springer, 1995. [Online]. Available: <https://doi.org/10.1007/978-1-84628-615-5>
- [26] J. Reher, C. Kann, and A. D. Ames, “An inverse dynamics approach to control Lyapunov functions,” 2020.
- [27] M. Tucker, M. Cheng, E. Novoseller, R. Cheng, Y. Yue, J. W. Burdick, and A. D. Ames, “Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3423–3430.
- [28] K. Li, M. Tucker, E. Biyik, E. Novoseller, J. W. Burdick, Y. Sui, D. Sadigh, Y. Yue, and A. D. Ames, “ROIAL: Region of interest active learning for characterizing exoskeleton gait preference landscapes,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3212–3218.
- [29] W. Chu and Z. Ghahramani, “Preference learning with Gaussian processes,” in *Proceedings of the 22nd International Conference on Machine learning (ICML)*, 2005, pp. 137–144.
- [30] O. Chapelle and L. Li, “An empirical evaluation of Thompson sampling,” *Advances in neural information processing systems*, vol. 24, pp. 2249–2257, 2011.
- [31] “Video of the experimental results.” [https://youtu.be/jMX5a\\_6Xcuw](https://youtu.be/jMX5a_6Xcuw).