Human Preference-Based Learning for High-dimensional Optimization of Exoskeleton Walking Gaits

Maegan Tucker[1], Myra Cheng[2], Ellen Novoseller[2], Richard Cheng[1],
Yisong Yue[2], Joel W. Burdick[1,2], and Aaron D. Ames[1,2]

Abstract—Understanding users' gait preferences of a lower-body exoskeleton requires optimizing over the high-dimensional gait parameter space. However, existing preference-based learning methods have only explored low-dimensional domains due to computational limitations. To learn user preferences in high dimensions, this work presents LINECOSPAR, a human-in-the-loop preference-based framework that enables optimization over many parameters by iteratively exploring one-dimensional subspaces. Additionally, this work identifies gait attributes that characterize broader preferences across users. In simulations and human trials, we empirically verify that LINECOSPAR is a sample-efficient approach for high-dimensional preference optimization. Our analysis of the experimental data reveals a correspondence between human preferences and objective measures of dynamic stability, while also highlighting inconsistencies in the utility functions underlying different users' gait preferences. This has implications for exoskeleton gait synthesis, an active field with applications to clinical use and patient rehabilitation.
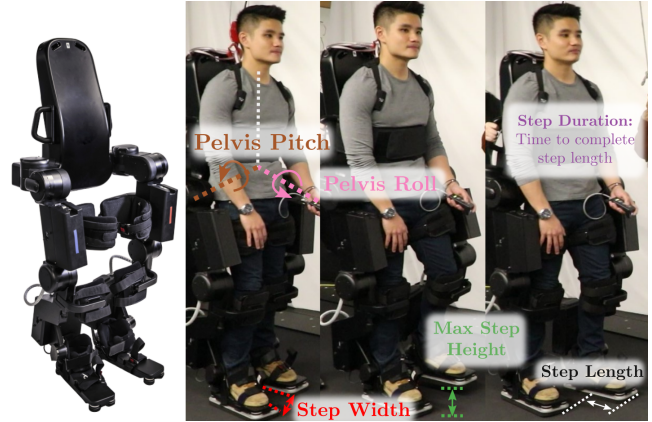
Fig. 1. **Atalante Exoskeleton.** The exoskeleton has 12 actuated joints. The experiments explore six exoskeleton gait parameters: step length, step duration, step width, maximum step height, pelvis roll, and pelvis pitch.

I. INTRODUCTION

Human-in-the-loop online learning techniques have demonstrated significant potential in human-robot interaction tasks [1]–[3], such as in improving the performance of robotic assistive devices. In particular, online learning from human feedback can help to optimize walking gaits for lower-body exoskeletons [4]–[6], which are placed over existing limbs to assist mobility-impaired individuals.

This work focuses on optimizing walking gaits for individual user comfort using the Atalante lower-body exoskeleton developed by Wandercraft. We use a pre-computed gait library, which generates walking gaits offline using optimization-based techniques from nonlinear dynamics [7]–[9]. Gaits are specified by many parameters, ranging from centers of pressure to step dimensions (step length, width, etc.).

Optimizing gait parameters for each exoskeleton user serves two purposes. First, it enables gait personalization to maximize each user's comfort. Second, the relationships among different users' preference functions, in particular their optimal gaits, provide insight into the properties of preferred gaits, which are currently not well-understood. While some gait optimization approaches rely on numeric

metrics such as the user's metabolic expenditure [5], there are no metrics that have established correspondences with user comfort. For example, metabolic expenditure is not an appropriate metric as the exoskeleton does not require the user to expend effort towards walking. A quantitative understanding of human preferences could help generate new, improving gaits beyond the gait library, which represents a small fraction of the rich space of human walking behaviors. This motivates optimizing over the high-dimensional space of exoskeleton gaits to characterize the utility functions governing users' gait preferences.

We rely on users' *relative preferences* to learn exoskeleton gaits optimizing their comfort, as several studies have shown that for subjective human feedback, pairwise preferences are more reliable than numerical scores [10]–[12]. While interactive preference-based learning methods have been applied to robotics [4], [6], existing online preference-based learning methods are restricted to low-dimensional domains due to computational limitations; for example, previous work on preference-based exoskeleton gait optimization either learns over at most two dimensions [4] or utilizes domain knowledge to narrow the search space before performing online learning [6].

We present LINECOSPAR, a high-dimensional human preference-based learning approach that integrates existing techniques for preference learning [4] and high-dimensional optimization [13] into a unified framework. LINECOSPAR relies on preference feedback to iteratively explore one-dimensional subspaces. We demonstrate in simulation that

[1]Authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125

[2]Authors are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125

LINECOSPAR exhibits sample-efficient convergence to user-preferred actions in high-dimensional spaces. The algorithm is then deployed experimentally to optimize exoskeleton walking over six gait parameters for six able-bodied subjects.

Using the preferred gaits identified by LINECOSPAR in the human experiments, we examine the connections among user-preferred gaits to gain understanding of what makes some gaits preferable to others. An analysis using the Zero Moment Point [14] reveals that users' preferences correspond to quantitative metrics of dynamic stability. We observe that while certain users' preferences are remarkably consistent with a metric that prioritizes dynamic stability, other users' preferences are explained by metrics that favor static stability. Based on this analysis, we suggest metrics that should be considered in the gait generation process, such that future exoskeleton gait designs can draw from regions of the gait trajectory space that prioritize user comfort.

II. THE LEARNING ALGORITHM

The LINECOSPAR algorithm (Alg. 1) builds upon COSPAR [4] by learning a Bayesian model over users' preferences in higher-dimensional spaces. Drawing inspiration from the LINEBO algorithm [13], LINECOSPAR exploits low-dimensional structure in the search space by dividing the problem into a series of one-dimensional subproblems. This allows LINECOSPAR to maintain its Bayesian preference relation function over only a subset of the action space in each iteration, reducing the computational complexity of learning the model posterior compared to COSPAR. This section provides background on existing approaches and then describes the LINECOSPAR algorithm, including 1) defining the posterior updating procedure, 2) achieving high-dimensional learning, and 3) incorporating Thompson sampling and coactive feedback.

A. Background

Preference-Based Learning. We learn users' preferred exoskeleton gaits through their relative preferences, which are more reliable than subjective numerical feedback [4], [10]–[12]. To maximize sample efficiency, we adopt the mixed-initiative approach of COSPAR [4], which learns from both pairwise preference and coactive feedback. In coactive learning [15], [16], after each time the algorithm selects an action, the user identifies an improved action. Under both feedback types, the exoskeleton user tests various gaits to specify preferences and suggest gait modifications. COSPAR effectively identifies user-preferred gait parameters across one and two dimensions. However, it is intractable over larger parameter spaces, as it involves manipulating (e.g. inverting) matrices whose sizes grow exponentially in the action space dimension.

To learn from preferences, we adopt the dueling bandit setting [17]–[19], in which the algorithm selects actions and receives relative preferences between them. This builds upon SELFSPARRING [17], a state-of-the-art Thompson sampling-based algorithm that iterates through: a) updating a Bayesian posterior over the actions' utilities given the data, b) sampling

---

Algorithm 1 LINECOSPAR

1: **procedure** LINECOSPAR(Utility prior parameters; m = granularity of discretization)
2:    $\mathcal{D} = \emptyset$, $\mathcal{W} = \emptyset$ ▷ $\mathcal{D}$: preference data, $\mathcal{W}$: actions in $\mathcal{D}$
3:    Set p_1, a_0 to uniformly-random actions
4:    **for** t = 1, 2,..., T **do**
5:       $\mathcal{L}_t$ = random line through p_t, discretized via m
6:       $\mathcal{V}_t = \mathcal{L}_t \cup \mathcal{W}$ ▷ Points over which to update posterior
7:       $(\boldsymbol{\mu}_t, \Sigma_t)$ = posterior over points in $\mathcal{V}_t$, given $\mathcal{D}$
8:       Sample utility function $f_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$
9:       Execute action $a_t = \text{argmax}_{a \in \mathcal{V}_t} f_t(a)$
10:      Add pairwise preference between a_t and a_{t-1} to $\mathcal{D}$
11:      Add coactive feedback a_t' to $\mathcal{D}$
12:      Set $\mathcal{W} = \mathcal{W} \cup \{a_t\} \cup \{a_t'\}$ ▷ Update actions in $\mathcal{D}$
13:      Set $p_{t+1} = \text{argmax}_{a \in \mathcal{V}_t} \mu_t(a)$
14:    **end for**
15: **end procedure**

---

utility functions from the posterior, c) executing the actions that maximize the sampled utility functions, and d) observing preferences among the executed actions.

High-Dimensional Bayesian Optimization. Bayesian optimization is a powerful approach for optimizing expensive-to-evaluate black-box functions. It maintains a model posterior over the unknown function, and cycles through a) using the posterior to acquire actions at which to query the function, b) querying the function, and c) updating the posterior using the obtained data. This procedure is challenging in high-dimensional search spaces due to the computational cost of the acquisition step (a), which often requires solving a non-convex optimization problem over the search space, and maintaining the posterior in (c), which can require manipulating matrices that grow exponentially with the input space's dimension. Dimensionality reduction techniques are therefore an area of active interest. Solutions vary from optimizing variable subsets (DROPOUTBO) [20] to projecting into lower-dimensional spaces (REMBO) [21] to sequentially optimizing over one-dimensional subspaces (LINEBO) [13]. We draw upon the approach of LINEBO because of its strong theoretical guarantees, which depend on the objective function's underlying latent dimensionality.

B. The LINECOSPAR Algorithm

Modeling Utilities Using Pairwise Preference Data. LINECOSPAR uses pairwise comparisons to learn a Bayesian model posterior over the utilities of each action (i.e., gait parameter combination) to the user based upon the Gaussian process preference model in [22]. Let $\mathcal{A} \subset \mathbb{R}^d$ be the set of possible actions. In iteration t of the algorithm, we consider a subset of the actions $\mathcal{V}_t \subset \mathcal{A}$, with cardinality V_t (we will define $\mathcal{V}_t$ later). We assume that each action $a \in \mathcal{A}$ has a latent utility to the user, denoted as $f(a)$. Throughout the learning process, LINECOSPAR stores a dataset of all user feedback, $\mathcal{D} = \{a_{k_1} \succ a_{k_2} \mid k = 1, \ldots, N\}$, consisting of N preferences, where $a_{k_1} \succ a_{k_2}$ indicates that the user prefers action a_{k_1} to action a_{k_2}. The preference data $\mathcal{D}$ is used to update the posterior utilities of the actions in $\mathcal{V}_t$. Defining

$\boldsymbol{f} = [f(\boldsymbol{a}_{t_1}), f(\boldsymbol{a}_{t_2}), \ldots, f(\boldsymbol{a}_{t_{V_t}})]^T \in \mathbb{R}^{V_t}$, where $\boldsymbol{a}_{t_i}$ is the ith action in $\mathcal{V}_t$, the utilities $\boldsymbol{f}$ have posterior:

$$\mathcal{P}(\boldsymbol{f}|\mathcal{D}) \propto \mathcal{P}(\mathcal{D}|\boldsymbol{f})\mathcal{P}(\boldsymbol{f}). \tag{1}$$

In iteration t, we define a Gaussian prior over the utilities $\boldsymbol{f}$ of actions in $\mathcal{V}_t$:

$$\mathcal{P}(\boldsymbol{f}) = \frac{1}{(2\pi)^{V_t/2}|\Sigma_t^{\text{pr}}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{f}^T[\Sigma_t^{\text{pr}}]^{-1}\boldsymbol{f}\right),$$

where $\Sigma_t^{\text{pr}} \in \mathbb{R}^{V_t \times V_t}$, $[\Sigma_t^{\text{pr}}]_{ij} = \mathcal{K}(\boldsymbol{a}_{t_i}, \boldsymbol{a}_{t_j})$, and $\mathcal{K}$ is a kernel. Our experiments use the squared exponential kernel. To compute the likelihood $\mathcal{P}(\mathcal{D}|\boldsymbol{f})$, we assume that the preferences may be corrupted by noise, such that:

$$\mathcal{P}(\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}|\boldsymbol{f}) = g\left(\frac{f(\boldsymbol{a}_{k_1}) - f(\boldsymbol{a}_{k_2})}{c}\right),$$

where $g(\cdot) \in [0, 1]$ is a monotonically-increasing link function, and $c > 0$ is a hyperparameter indicating the degree of preference noise. While COSPAR assumes a Gaussian noise model, such that g is the standard normal cumulative distribution function [4], [22], we found the heavier-tailed sigmoid distribution, $g_{\text{sig}}(x) := \sigma(x) = \frac{1}{1+e^{-x}}$, to be a more effective link function. $g_{\text{sig}}(x)$ satisfies the convexity conditions for the Laplace approximation [23] and has been used to model preferences in other contexts [24]. The full likelihood expression becomes:

$$\mathcal{P}(\mathcal{D}|\boldsymbol{f}) = \prod_{k=1}^{N} g_{\text{sig}}\left(\frac{f(\boldsymbol{a}_{k_1}) - f(\boldsymbol{a}_{k_2})}{c}\right).$$

The posterior in (1) is estimated via the Laplace approximation as in [22], yielding a multivariate Gaussian, $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$.

Sampling Approach for Higher Dimensions. Previous work [4] discretizes each of the d action space dimensions into m bins, such that $\mathcal{A}$ is a finite action set with cardinality $A = |\mathcal{A}| = m^d$. Larger m enables finer-grained search at a higher computational cost. This is intractable for higher d since computing the posterior involves inverting $\Sigma_t^{\text{pr}}, \Sigma_t \in \mathbb{R}^{A \times A}$. Inspired by [13], LINECOSPAR overcomes this intractibility by iteratively considering one-dimensional subspaces (i.e. lines), rather than considering the entire discretized action space at once. In each iteration t, LINECOSPAR selects points along a new random line $\mathcal{L}_t$, which is determined by a uniformly-random direction and the action $\boldsymbol{p}_t$ that maximizes the posterior mean. Including $\boldsymbol{p}_t$ in the subspace encourages exploration of higher-utility areas. The posterior $\mathcal{P}(\mathcal{D}|\boldsymbol{f})$ is calculated over $\mathcal{V}_t = \mathcal{L}_t \cup \mathcal{W}$, where $\mathcal{W}$ is the set of actions over which $\mathcal{D}$ has preference feedback.

Critically, this approach shrinks the covariance matrix of the preference relation distribution, Σ_t, from size $A \times A$ to $V_t \times V_t$. Thus, the algorithm's complexity is constant in the dimension d and linear in the iteration t, whereas the baseline algorithm's complexity grows exponentially in d (Fig. 2).

Posterior Sampling Algorithm. Utilities are learned using the SELFSPARRING [17] approach to Thompson sampling
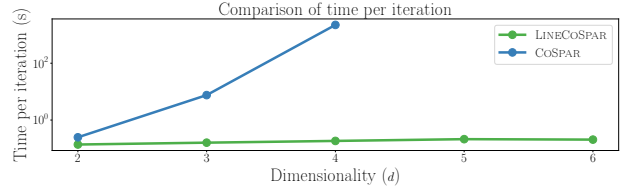

Fig. 2. **Curse of dimensionality for COSPAR.** Average time per iteration of COSPAR vs. LINECOSPAR. The y-axis is on a logarithmic scale. For LINECOSPAR, the time is roughly constant in the number of dimensions d, while the runtime of COSPAR increases exponentially. For $d = 4$, the duration of a COSPAR iteration is inconvenient in the human-in-the-loop learning setting, and for $d \geq 5$, it is intractable.

detailed above. Specifically, in each iteration, we calculate the posterior of the utilities $\boldsymbol{f}$ over the points in $\mathcal{V}_t = \mathcal{L}_t \cup \mathcal{W}$, obtaining the posterior $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ over $\mathcal{V}_t$. The algorithm then samples a utility function f_t from the posterior, which assigns a utility to each action in $\mathcal{V}_t$. LINECOSPAR then executes the action $\boldsymbol{a}_t$ that maximizes f, $\boldsymbol{a}_t = \text{argmax}_{\boldsymbol{a} \in \mathcal{V}_t} f(\boldsymbol{a})$. The user provides a preference (or indicates indifference, i.e. "no preference") between $\boldsymbol{a}_t$ and the preceding action $\boldsymbol{a}_{t-1}$.

In addition, for each executed action $\boldsymbol{a}_t$, the user can provide coactive feedback, specifying the dimension, direction (higher or lower), and degree in which to change $\boldsymbol{a}_t$. The user's suggested action $\boldsymbol{a}_t'$ is added to $\mathcal{W}$, and the feedback is added to $\mathcal{D}$ as $\boldsymbol{a}_t' \succ \boldsymbol{a}_t$. In each iteration, preference and coactive feedback each add at most one action to $\mathcal{W}$. Thus, in iteration t, $\mathcal{V}_t$ contains at most $m + 2(t - 1)$ actions, and so its size is independent of the dimensionality d.

In the subsequent analysis, $\boldsymbol{a}_{\text{max}}$ is defined as the action maximizing the final posterior mean after T iterations, i.e., $\boldsymbol{a}_{\text{max}} := \text{argmax}_{\boldsymbol{a} \in \mathcal{V}_t} \mu_{T+1}(\boldsymbol{a})$. Analogously, $\boldsymbol{a}_{\text{min}} := \text{argmin}_{\boldsymbol{a} \in \mathcal{V}_t} \mu_{T+1}(\boldsymbol{a})$.

III. PERFORMANCE OF LINECOSPAR

A. Simulation Results

We validate the performance of LINECOSPAR in simulation using both standard Bayesian optimization benchmarks and randomly-generated polynomials.[1] The simulations show that LINECOSPAR is sample-efficient, converges to sampling higher-valued actions, and learns a preference relation function such that actions with higher objective values have high posterior utilities.

Standard Bayesian Optimization Benchmarks. We evaluated the performance of LINECOSPAR on the standard Hartmann3 (H3) and Hartmann6 (H6) benchmarks (3 and 6 dimensions, respectively). Preferences are generated in simulation by comparing objective function values. We validate LINECOSPAR with noiseless preferences and then demonstrate its robustness to noisy user preferences.

Under ideal preference feedback, $\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}$ if $f(\boldsymbol{a}_{k_1}) > f(\boldsymbol{a}_{k_2})$. The true objective values f are invisible to the algorithm, which observes only the preference dataset $\mathcal{D}$.

---

[1]All experiments use the squared exponential kernel with lengthscale 0.15 in every dimension, signal variance 1e−4, noise variance 1e−5, and preference noise 0.005.
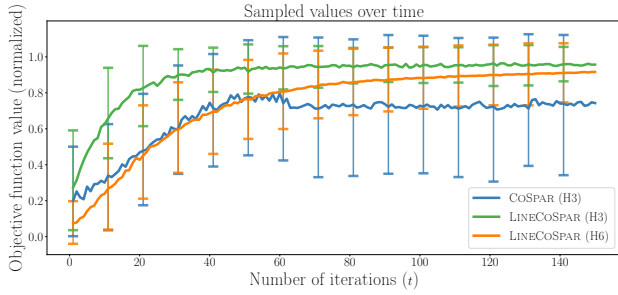
Fig. 3. **Convergence to higher values on standard benchmarks.** Mean objective value $\pm$ SD using H3 and H6, averaged over 100 runs. The sampled actions converge to higher objective values at a faster rate with LINECOSPAR, which has an improved sampling approach and link function. It is intractable to run COSPAR on a 6-dimensional space.
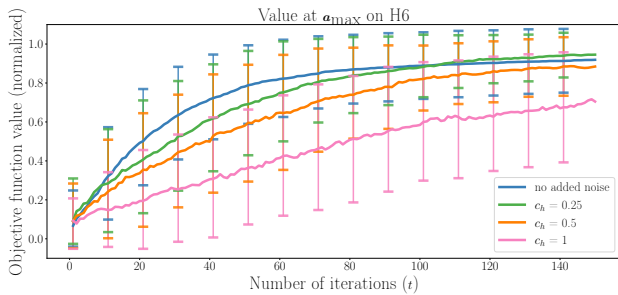

Fig. 4. **Robustness to noisy preferences.** Mean objective value $\pm$ SD of the action $\boldsymbol{a}_{\max}$ with the highest posterior utility. This is averaged over 100 runs using LINECOSPAR on H6 with varying preference noise, as quantified by c_h. Higher performance correlates with less noise (lower c_h). The algorithm is robust to noise to a certain degree.

Compared to COSPAR, LINECOSPAR converges to sampling actions with higher objective values at a faster rate (Fig. 3). Thus, LINECOSPAR not only enables higher-dimensional optimization, but also improves speed and accuracy of learning.

Since human preferences may be noisy, we tested the algorithm's robustness to noisy preference feedback. In simulation, this is modeled via $\mathcal{P}(\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}) = \frac{1}{1+e^{-\frac{s_k}{c_h}}}$, where $s_k = f(\boldsymbol{a}_{k_1}) - f(\boldsymbol{a}_{k_2})$ and c_h is a hyperparameter controlling the noise level. As $c_h \to \infty$, the preferences approach uniform randomness (i.e. become noisier). Also, actions become less distinguishable when the distance between $f(\boldsymbol{a}_{k_1})$ and $f(\boldsymbol{a}_{k_2})$ decreases. This reflects human preference generation since it is more difficult to give consistent preferences between actions with similar utilities. By simulating noisy preferences, we demonstrate that LINECOSPAR is robust to noisy feedback to a certain degree (Fig. 4).

Randomly-Generated Functions. We also tested LINECOSPAR using randomly-generated d-dimensional polynomials (for $d = 6$) as objective functions: $p(\boldsymbol{a}) = \sum_{i=1}^{d} \alpha_i \sum_{j=1}^{d} \beta_j a_j$, where a_j denotes the jth element of $\boldsymbol{a}$, and $\alpha_i, \beta_i, i \in \{1, \ldots, d\}$ are sampled independently from the uniform distribution $\mathcal{U}(-1, 1)$. The dimensions' ranges and discretizations match those in the exoskeleton experiments, so that these simulations approximate the number of human trials needed to find optimal gaits.

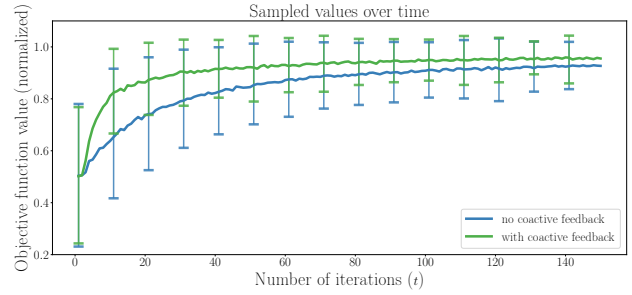Coactive feedback was simulated for each sampled action

Fig. 5. **Coactive feedback improves convergence.** Mean objective value $\pm$ SD of the sampled actions using random functions. This is averaged over 1000 runs using LINECOSPAR on 100 randomly-generated six-dimensional functions ($d = 6$). The sampled actions converge to high objective values in relatively few iterations, and coactive feedback accelerates this process.

$\boldsymbol{a}_t$ by finding an action $\boldsymbol{a}_t'$ with a higher objective value that differs from $\boldsymbol{a}_t$ along only one dimension. The action $\boldsymbol{a}_t'$ is determined by randomly choosing a dimension in $\{1, \ldots, d\}$ and direction (positive or negative), and taking a step from $\boldsymbol{a}_t$ along this vector. If the resulting action $\boldsymbol{a}_t'$ has a higher objective value, it is added to the dataset $\mathcal{D}$ as $\boldsymbol{a}_t' \succ \boldsymbol{a}_t$. This is a proxy for the human coactive feedback acquired in the exoskeleton experiments described below, in which the user can suggest a dimension and direction in which to modify an action to obtain an improved gait.

Fig. 5 displays LINECOSPAR's performance over 100 randomly-generated polynomials (10 repetitions each) with computation time shown in Fig. 2. The results demonstrate that LINECOSPAR samples high-valued actions within relatively few iterations (≈ 20 with coactive feedback).

B. Human Subject Experiments

Our simulations demonstrate that LINECOSPAR efficiently locates optimal actions in high-dimensional spaces, suggesting that it can optimize parameters of assistive devices in human trials. Next, the algorithm is experimentally deployed on the lower-body exoskeleton Atalante (Fig. 1) to optimize six gait parameters for six able-bodied users. A video illustrating the experimental results is provided [25].

Atalante Exoskeleton. Atalante (Fig. 1) [8], [26], [27], developed by Wandercraft, has 12 actuated joints: three at each hip, one at each knee, and two in each ankle. [9] describes the device's mechanical components and control architecture in detail. Exoskeleton walking is achieved using pre-computed walking gaits, generated using the partial hybrid zero dynamics framework [28] and a nonlinear constrained optimization process that utilizes direct collocation. The configuration space of the human-exoskeleton system is constructed as $q = (p, \phi, q_b) \in Q \in \mathbb{R}^{18}$, where $p \in \mathbb{R}^3$ and $\phi \in \mathbb{SO}^3$ denote the position and orientation of the exoskeleton floating base frame with respect to the world frame, and $q_b \in \mathbb{R}^{12}$ denotes the relative angles of the actuated joints. The generated gaits are realized on the exoskeleton using PD control at the joint level and a high-level controller adjusting joint targets based on state feedback. The controller is executed by an embedded computer unit running a real-time operating system. Gaits
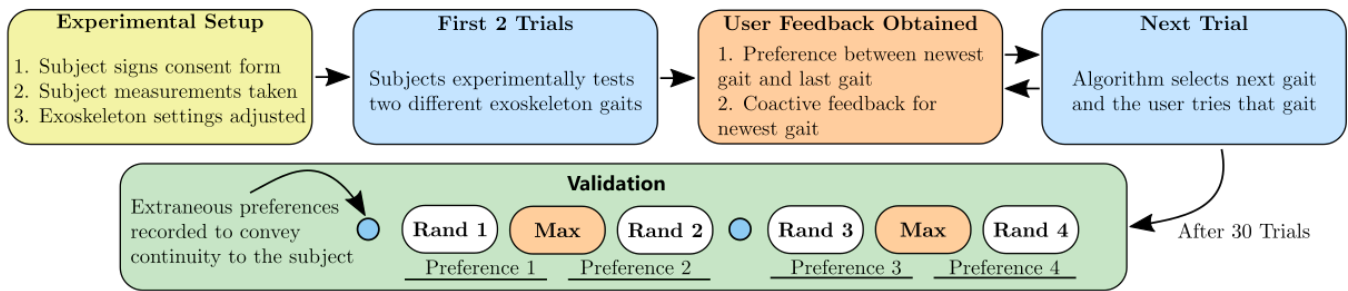
Fig. 6. **Experimental Procedure.** Procedure for human experiments on the exoskeleton. After setup of the subject-exoskeleton system, subjects were queried for preferences between all consecutive pairs of gaits, along with coactive feedback, in 30 gait trials (for a total of at most 29 pairwise preferences and 30 pieces of coactive feedback). After these 30 trials, the subject unknowingly entered the validation portion of the experiment, in which he/she validated the posterior-maximizing gait, a_{max}, against four randomly-selected gaits.
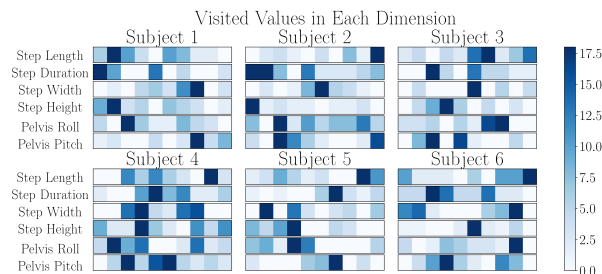
(a) Each row depicts the distribution of particular gait parameter's values across all gaits that the subject tested. Each dimension is discretized into 10 bins. Note that the algorithm explores different parts of the action space for each subject.
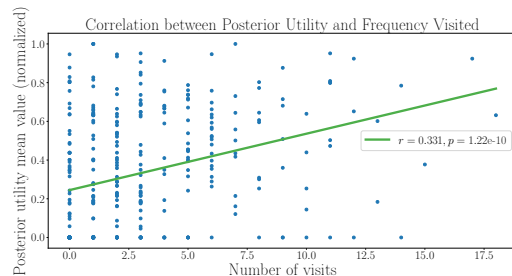
(b) As the algorithm learns users' preferences, it exploits higher-utility parts of the action space. There is a significant positive correlation between the average posterior utility of the actions in each bin and the frequency that the bin is visited (bins as depicted in (a)).

Fig. 7. **Exploration vs. exploitation in human trials**.

are sent to the exoskeleton over a wireless connection via a custom graphical user interface.

Experimental Procedure. LINECOSPAR optimized exoskeleton gaits for six able-bodied subjects over six gait parameters (Fig. 1): step length, step duration, step width, maximum step height, pelvis roll, and pelvis pitch. The parameter ranges, respectively, were: 0.08-0.18 meters, 0.85-1.15 seconds, 0.25-0.3 meters, 0.065-0.075 meters, 5.5-9.5 degrees, and 10.5-14.5 degrees. Fig. 6 illustrates the experimental procedure for testing and validating LINECOSPAR.

All subjects were volunteers without prior exoskeleton exposure. For each subject, the testing procedure lasted approximately two hours, with one hour of setup and explanation followed by one hour of exoskeleton testing. Before testing, the subject was informed of the procedure for providing feedback after each trial, including how to give coactive feedback for each of the six gait parameters. The subjects had control over initiating and terminating each instance of exoskeleton walking and were instructed to try each walking gait until they felt comfortable giving a preference. The subjects could choose to test each gait multiple times to confirm their preference. They could also specify "no preference" between two gait trials, in which case no new information was added to the dataset $\mathcal{D}$.

After completing 30 trials (including trials with no preference, but not including voluntary gait repetitions), the subject began a set of "validation" trials; for consistency, the

subject was not informed of the start of the validation phase. Validation consisted of six additional trials and yielded four pairwise preferences, each between the posterior-maximizing action a_{max} and a randomly-generated action. This validation step verifies that a_{max} is preferred over other parameter combinations across the search space.

Gait Optimization Results. Fig. 7 illustrates that the LINECOSPAR algorithm both explores across the gait parameter space and exploits regions with higher posterior utility. As LINECOSPAR performs more iterations, it samples actions concentrated in regions of the search space that are preferred based on previous feedback. This results in a significant correlation between visitation frequencies and posterior utilities across these regions (Fig. 7b).

For each subject, Table I lists the parameters of the predicted optimal gaits, a_{max}, identified by LINECOSPAR. Table I also illustrates the results of the validation trials for each subject. These results show that a_{max} was predominantly preferred over the randomly-selected actions during validation. For four of the six subjects, all four validation preferences matched the posterior, while the other subjects matched three and one of the four preferences, respectively. Outliers may be due to noisy preference feedback that did not reflect the users' true utilities.

Subject	Height (m)	Mass (kg)	Step Length (m)	Step Duration (s)	Step Width (m)	Max Step Height (m)	Pelvis Roll (deg)	Pelvis Pitch (deg)	Validation Accuracy (%)
1	1.85	89.9	0.0835	0.943	0.278	0.0674	6.38	10.9	75
2	1.668	69.2	0.136	1.04	0.285	0.0679	6.41	12.4	100
3	1.635	51.2	0.137	0.922	0.279	0.0688	8.56	11.4	100
4	1.795	73.6	0.127	0.989	0.268	0.065	6.68	12.7	25
5	1.625	55.9	0.161	1.05	0.258	0.0689	7.32	13.2	100
6	1.66	65	0.177	1.11	0.256	0.0663	7.71	13.5	100

IV. ANALYSIS OF PREFERENCE FEEDBACK AND IMPLICATIONS FOR GAIT SYNTHESIS

In addition to optimizing exoskeleton walking gaits for individual users, we aim to understand the utility functions underlying human preferences and apply this knowledge towards improving gait synthesis. As discussed in [4], exoskeleton gaits are generated using the partial hybrid zero dynamics framework, which is formulated by the nonlinear optimization problem [8]:

$$\alpha^* = \underset{\alpha}{\arg\min} \quad \mathcal{J}(\alpha)$$
$$\text{s.t.} \quad \Delta(\mathcal{S} \cap \mathcal{PZ}_\alpha) \subset \mathcal{PZ}_\alpha$$
$$\mathcal{W}_i x \leq b_i$$
$$\dot{\eta}_\alpha = A_{cl}\eta_\alpha,$$

where α are coefficients of Bézier polynomials that yield impact-invariant periodic orbits, $\mathcal{J}(\alpha)$ is a user-determined cost, $\Delta(\mathcal{S} \cap \mathcal{PZ}_\alpha) \subset \mathcal{PZ}_\alpha$ is the impact invariance condition, $\mathcal{W}_i x \leq b_i$ are other physical constraints, and $\dot{\eta}_\alpha = A_{cl}\eta_\alpha$ is the output dynamics condition. For more details on these constraints, refer to [28].

The cost function $\mathcal{J}(\alpha)$ largely influences the behavior of the walking gaits that it generates; however, the user's cost function $\mathcal{J}_{\text{human}}$ underlying their preferences is poorly-understood. This section aims to describe the relationship between gaits and user preferences through the underlying cost function $\mathcal{J}_{\text{human}}$, so that future gait synthesis can be streamlined towards user-preferred walking. Thus, we aim to identify key terms in $\mathcal{J}_{\text{human}}$ that numerically account for the preferences captured by LINECOSPAR.

All walking gaits on the exoskeleton are flat-footed. Thus, by analyzing the center of mass (CoM) and center of pressure (CoP), we can treat the patient-exoskeleton system as a Linear Inverted Pendulum Model (LIPM). This allows us to analyze the underlying utility function $\mathcal{J}_{\text{human}}$ using the cost structure from [29]. We first introduce Zero Moment Point (ZMP) and LIPM, and then discuss correspondences between metrics of dynamic stability and user comfort.

Zero Moment Point. The Zero Moment Point (ZMP) is a widely-used notion of stability for bipedal robots that is defined as *the point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes* [14]. When the ZMP exists outside of the "support polygon," i.e. the convex hull of the stance foot (or stance feet in the double-support domain), the robot experiences foot roll.
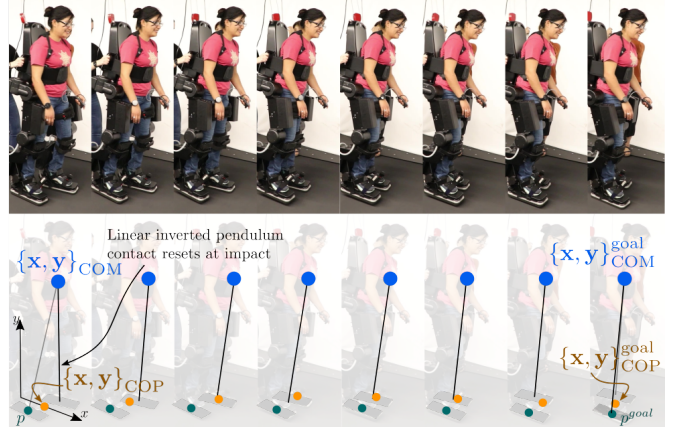

Fig. 8. Illustration of one step with the overlayed LIPM model.

Static and Dynamic Stability. For static stability, the CoM never leaves the support polygon. For dynamic stability, the CoM leaves the support polygon during the single-support phase but returns to the support polygon during the double-support phase when the swing leg is placed on the ground. For both types of stability, the ZMP always remains inside the support polygon.

Linear Inverted Pendulum Model (LIPM). The LIPM is a low-dimensional dynamical system for reduced-order gait generation. The LIPM model assumes constant height of the center of mass, as well as zero angular momentum. The dynamics of the LIPM are:

$$m\ddot{x}_{\text{CoM}} = \frac{mg}{z_0}(x_{\text{CoM}} - x_{\text{CoP}})$$
$$m\ddot{y}_{\text{CoM}} = \frac{mg}{z_0}(y_{\text{CoM}} - y_{\text{CoP}}),$$

where $\{x, y\}_{\text{CoM}}$ are the x and y positions of the CoM at constant height z_0, and $\{x, y\}_{\text{CoP}}$ denote the x and y positions of the CoP. For planar horizontal ground walking, the ZMP is mathematically equivalent to the CoP. The CoP was experimentally obtained using the four 3-axis force sensors on the bottom of the exoskeleton's feet.

Fitting the LIPM Cost Function to User Preferences. Since flat-foot level-ground walking is well captured by the LIPM model, the cost function used in the LIPM to generate desirable walking behavior may explain the users' utility functions underlying their exoskeleton gait preferences. As
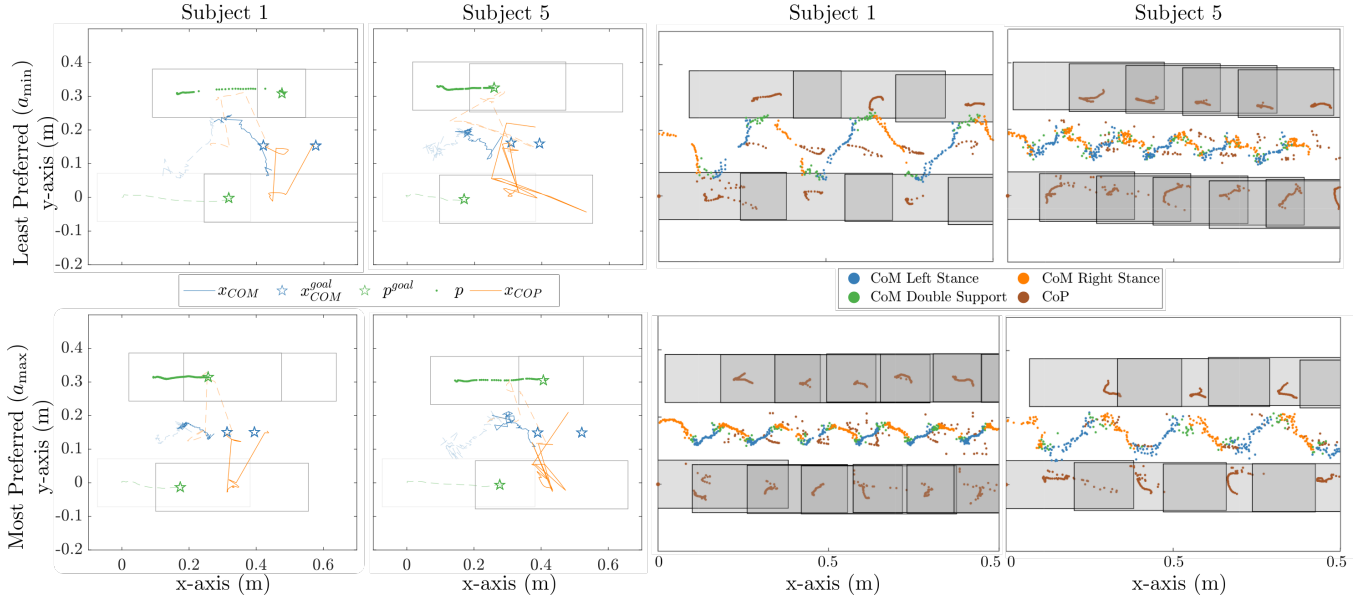
Fig. 9. **Comparison of Preferences**. This figure illustrates the trade-off between more and less dynamically stable gaits as well as the contrasting preferences among different subjects. While all of the exoskeleton gaits are dynamically-stable, both the least preferred gait ($a_{\min}$) of subject 1 and the most preferred gait ($a_{\max}$) of subject 5 exhibit behavior closer to statically-stable gaits. Subject 1 preferred dynamic gaits with a large difference between x_{CoP} and x_{CoM}; in contrast, subject 5 preferred gaits in which x_{CoP} closely followed the center of mass. Rectangles represent the exoskeleton's feet.

defined in [29], the LIPM cost function is:

$$\mathcal{J}_{\text{LIPM}} = w_1||x^{goal}_{\text{CoM}} - x_{\text{CoM}}||^2 + w_2||\dot{x}_{\text{CoM}}||^2 +$$
$$w_3||\dot{x}_{\text{CoP}}||^2 + w_4||p^{goal}_x - p_x||^2 +$$
$$w_1||y^{goal}_{\text{CoM}} - y_{\text{CoM}}||^2 + w_2||\dot{y}_{\text{CoM}}||^2 +$$
$$w_3||\dot{y}_{\text{CoP}}||^2 + w_4||p^{goal}_y - p_y||^2,$$

where $\{x, y\}^{goal}_{\text{CoM}}$ denotes the CoM goal position in the x and the y directions, $\{\dot{x}, \dot{y}\}_{\text{CoP}}$ denotes the velocity of the CoP in the x and y directions, $\{\dot{x}, \dot{y}\}_{\text{CoM}}$ is the velocity of the CoM, $p^{goal}_{\{x,y\}}$ denotes the next stance foot position in the x and y directions, and $p_{\{x,y\}}$ denotes the x and y positions of the swing foot (Fig. 8).

The hypothesis is that we can capture $\mathcal{J}_{\text{human}}(w)$ as a function of the weights w_i. Therefore, we fit weights w of $\mathcal{J}_{\text{LIPM}}$ to the validation-stage preference data, i.e., the preferences between the most-preferred gaits (gaits with parameters $a_{\max}$) and each of the random gaits presented during the validation phase. The weights w were optimized via the quatratic program:

$$w* = \operatorname*{argmin}_{w} ||w||$$

$$\text{s.t. } \begin{bmatrix} \delta^{(1)}_1 & \delta^{(1)}_2 & \delta^{(1)}_3 & \delta^{(1)}_4 \\ & & \vdots & \\ \delta^{(n)}_1 & \delta^{(n)}_2 & \delta^{(n)}_3 & \delta^{(n)}_4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} < 0,$$

where n denotes the number of pairwise preferences, and:

$$\delta_i = \left(||x^{\text{pref}}_{(i,x)}||^2 + ||x^{\text{pref}}_{(i,y)}||^2\right) - \left(||x^{\text{not pref}}_{(i,x)}||^2 + ||x^{\text{not pref}}_{(i,y)}||^2\right)$$

$$\begin{aligned} x_{(1,x)} &= x^{goal}_{\text{CoM}} - x_{\text{CoM}} & x_{(1,y)} &= y^{goal}_{\text{CoM}} - y_{\text{CoM}} \\ x_{(2,x)} &= \dot{x}_{\text{CoM}} & x_{(2,y)} &= \dot{y}_{\text{CoM}} \\ x_{(3,x)} &= \dot{x}_{\text{CoP}} & x_{(3,y)} &= \dot{y}_{\text{CoP}} \\ x_{(4,x)} &= p^{goal}_x - p_x & x_{(4,y)} &= p^{goal}_y - p_y. \end{aligned}$$

We use subject-wise holdout (leave-one-out) cross-validation across the subjects to verify the reliability of the fit. The weights across all six subjects are: $w_1 = -0.1595$, $w_2 = 0.1506$, $w_3 = -0.1198$, and $w_4 = 1.1461$.

We quantify the predictive power of each fitted cost function on the users' utility functions using the rank consistency between the cost function values and the preference data. Table II shows the predictive power of $\mathcal{J}_{\text{LIPM}}$ on the preferences, as well as the predictive power of two other cost functions, $\mathcal{J}_{\text{static}}$ and $\mathcal{J}_{\text{dynamic}}$, respectively defined as:

$$\mathcal{J}_{\text{static}} = ||\{x, y\}_{\text{CoM}} - \{x, y\}_{\text{CoP}}||^2,$$
$$\mathcal{J}_{\text{dynamic}} = ||p^{goal}_{\{x,y\}} - p_{\{x,y\}}||^2.$$

These two metrics are directly opposed: while $\mathcal{J}_{\text{dynamic}}$ is the term from $\mathcal{J}_{\text{LIPM}}$ that promotes dynamic stability, $\mathcal{J}_{\text{static}}$ penalizes dynamic stability in favor of static stability. This is because in the LIPM dynamics, the acceleration of $\{x, y\}_{\text{CoM}}$ approaches zero as $\mathcal{J}_{\text{static}}$ approaches zero. We find that $\mathcal{J}_{\text{LIPM}}$ and $\mathcal{J}_{\text{dynamic}}$ capture the preferences of five of the six subjects, while $\mathcal{J}_{\text{static}}$ completely predicts the preferences of the single outlier, subject 5.

Fig. 9 further illustrates this difference. Not only do subject 1's and subject 5's preferences lean toward dynamic and static stability (as captured by $\mathcal{J}_{\text{dynamic}}$ and $\mathcal{J}_{\text{static}}$), respectively, but the gait tiles for these subjects' most and least-preferred gaits also exhibit these opposing styles of walking. The diametric opposition between the cost function terms predicting these users' preferences reflects inconsistencies across users' gait utility functions. This suggests that there is most likely no single metric that entirely captures all users' underlying utilities. Thus, it is important to generate a variety of gaits that satisfy the cost functions reflecting different users' preferences.

Cost Function	Correctly predicted preferences per subject (%)					
	1	2	3	4	5	6
$\mathcal{J}_{\textbf{LIPM}}$ (holdout)	75	100	62.5	75	37.5	100
$\mathcal{J}_{\textbf{LIPM}}$	75	87.5	72.5	75	62.5	100
$\mathcal{J}_{\textbf{dynamic}}$	100	100	50	75	12.5	37.5
$\mathcal{J}_{\textbf{static}}$	50	75	37.5	50	100	75

V. CONCLUSION

This work presents two main contributions: 1) the LINECOSPAR algorithm to efficiently learn personalized, user-preferred gaits in high dimensions, and 2) insight into the utilities underlying individual users' gait preferences. LINECOSPAR identifies preferred actions in high dimensions, both in simulation and experiments with six able-bodied subjects using the Atalante lower-body exoskeleton.

We then examine the experimentally-obtained gait preferences to gain insight into the utility functions underlying users' gait preferences. We identify opposing measures of stability that have predictive power for different users' preferences, implying that each user consistently prefers walking gaits that are either more dynamically or statically stable. These considerations may inform the synthesis of new exoskeleton gaits that maximize user comfort.

In the future, we plan to conduct additional exoskeleton trials to further characterize the structure of human gait preferences. Given more preference data, one could apply tensor decomposition techniques to discover invariant subspaces among the gait parameters. Such knowledge could accelerate learning of personalized gaits by guiding exploration.

The high-dimensional learning capabilities of LINECOSPAR provide insight into the exoskeleton walking gaits that maximize user comfort, paving the way for generating entirely new gaits beyond the gait library. This presents promising advancements for clinical trials and the broader rehabilitation community.

ACKNOWLEDGMENT

REFERENCES

[1] A. Bajcsy, D. P. Losey, M. K. OMalley, and A. D. Dragan, "Learning robot objectives from physical human interaction," in *Conf. on Robot Learning*, 2017, pp. 217–226.

[2] S. Christen, S. Stevšić, and O. Hilliges, "Guided deep reinforcement learning of control policies for dexterous human-robot interaction," in *Int. Conf. on Robotics and Automation*. IEEE, 2019, pp. 2161–2167.

[3] S. Cremer, S. K. Das, I. B. Wijayasinghe, D. O. Popa, and F. L. Lewis, "Model-free online neuroadaptive controller with intent estimation for physical human–robot interaction," *IEEE Trans. on Robotics*, 2019.

[4] M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. Burdick, and A. D. Ames, "Preference-based learning for exoskeleton gait optimization," *arXiv preprint arXiv:1909.12316*, 2019.

[5] J. Zhang, P. Fiers, K. A. Witte, R. W. Jackson, K. L. Poggensee, C. G. Atkeson, and S. H. Collins, "Human-in-the-loop optimization of exoskeleton assistance during walking," *Science*, vol. 356, no. 6344, pp. 1280–1284, 2017.

[6] N. Thatte, H. Duan, and H. Geyer, "A method for online optimization of lower limb assistive devices with high dimensional parameter spaces," in *Int. Conf. on Robotics and Automation*. IEEE, 2018, pp. 1–6.

[7] A. Agrawal, O. Harib, A. Hereid, S. Finet, M. Masselin, L. Praly, A. D. Ames, K. Sreenath, and J. W. Grizzle, "First steps towards translating hzd control of bipedal robots to decentralized control of exoskeletons," *IEEE Access*, vol. 5, pp. 9919–9934, 2017.

[8] O. Harib, A. Hereid, A. Agrawal, T. Gurriet, S. Finet, G. Boeris, A. Duburcq, M. E. Mungai, M. Masselin, A. D. Ames, *et al.*, "Feedback control of an exoskeleton for paraplegics: Toward robustly stable, hands-free dynamic walking," *IEEE Control Systems Magazine*, vol. 38, no. 6, pp. 61–87, 2018.

[9] T. Gurriet, S. Finet, G. Boeris, A. Duburcq, A. Hereid, O. Harib, M. Masselin, J. Grizzle, and A. D. Ames, "Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons," in *Int. Conf. on Robotics and Automation*. IEEE, 2018, pp. 2804–2811.

[10] C. Basu, Q. Yang, D. Hungerman, M. Sinahal, and A. D. Dragan, "Do you want your autonomous car to drive like you?" in *Int. Conf. on Human-Robot Interaction*. IEEE, 2017, pp. 417–425.

[11] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *SIGIR*, vol. 5, 2005, pp. 154–161.

[12] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue, "Large-scale validation and analysis of interleaved search evaluation," *ACM Transactions on Information Systems*, vol. 30, no. 1, p. 6, 2012.

[13] J. Kirschner, M. Mutny, N. Hiller, R. Ischebeck, and A. Krause, "Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces," in *Int. Conf. on Machine Learning*, 2019, pp. 3429–3438.

[14] M. Vukobratović and B. Borovac, "Zero-moment pointthirty five years of its life," *Int. journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[15] P. Shivaswamy and T. Joachims, "Online structured prediction via coactive learning," in *Int. Conf. on Machine Learning*. Omnipress, 2012, pp. 59–66.

[16] ——, "Coactive learning," *Journal of Artificial Intelligence Research*, vol. 53, pp. 1–40, 2015.

[17] Y. Sui, V. Zhuang, J. W. Burdick, and Y. Yue, "Multi-dueling bandits with dependent arms," in *Conf. on Uncertainty in Artificial Intelligence*, 2017.

[18] Y. Sui, M. Zoghi, K. Hofmann, and Y. Yue, "Advancements in dueling bandits," in *IJCAI*, 2018, pp. 5502–5510.

[19] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The k-armed dueling bandits problem," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1538–1556, 2012.

[20] C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton, "High dimensional Bayesian optimization using dropout," *Int. Joint Conf. on Artificial Intelligence*, 2017.

[21] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. D. Feitas, "Bayesian optimization in a billion dimensions via random embeddings," *J. of Artificial Intelligence Research*, vol. 55, p. 361387, 2016.

[22] W. Chu and Z. Ghahramani, "Preference learning with Gaussian processes," in *Int. Conf. on Machine Learning*. ACM, 2005, pp. 137–144.

[23] U. Ezeafulukwe, M. Darus, and O. Fadipe-Joseph, "On analytic properties of a sigmoid function," *Int. Journal of Mathematics and Computer Science*, vol. 13, no. 2, pp. 171–178, 1 2018.

[24] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A survey of preference-based reinforcement learning methods," *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.

[25] "Video of the experimental results." https://vimeo.com/394608113.

[26] A. Duburcq, Y. Chevaleyre, N. Bredech, and G. Boéris, "Online trajectory planning through combined trajectory optimization and function approximation: Application to the exoskeleton Atalante," *arXiv preprint arXiv:1910.00514*, 2019.

[27] T. Gurriet, M. Tucker, A. Duburcq, G. Boeris, and A. D. Ames, "Towards variable assistance for lower body exoskeletons," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 266–273, 2019.

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

[29] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *Int. Conf. on humanoid robots*. IEEE, 2010, pp. 52–59.