

# Data-Driven Predictive Control for Robust Exoskeleton Locomotion

Kejun Li<sup>1</sup>, Jeeseop Kim<sup>2</sup>, Xiaobin Xiong<sup>3</sup>, Kaveh Akbari Hamed<sup>4</sup>, Yisong Yue<sup>5</sup>, Aaron D. Ames<sup>2,6</sup>

**Abstract**—Exoskeleton locomotion must be robust while being adaptive to different users with and without payloads. To address these challenges, this work introduces a data-driven predictive control (DDPC) framework to synthesize walking gaits for lower-body exoskeletons, employing Hankel matrices and a state transition matrix for its data-driven model. The proposed approach leverages DDPC through a multi-layer architecture. At the top layer, DDPC serves as a planner employing Hankel matrices and a state transition matrix to generate a data-driven model that can learn and adapt to varying users and payloads. At the lower layer, our method incorporates inverse kinematics and passivity-based control to map the planned trajectory from DDPC into the full-order states of the lower-body exoskeleton. We validate the effectiveness of this approach through numerical simulations and hardware experiments conducted on the Atalante lower-body exoskeleton with different payloads. Moreover, we conducted a comparative analysis against the model predictive control (MPC) framework based on the reduced-order linear inverted pendulum (LIP) model. Through this comparison, the paper demonstrates that DDPC enables robust bipedal walking at various velocities while accounting for model uncertainties and unknown perturbations.

## I. INTRODUCTION

Recent research has made substantial advances in lower-body exoskeletons locomotion by enabling stable, dynamic, crutchless walking [1], [2]. However, the effort to synergistically integrate users’ walking experiences with the functionality of robotic assistive devices is still in progress [3]. The primary challenges hindering this progress include the complex nature of robotic bipedal movement—often described as a high degrees of freedom (DoFs), underactuated hybrid system [4]—along with uncertainties in accurately modeling these systems and the unpredictable dynamics of user-exoskeleton interactions due to unmodeled user movements inside the exoskeleton (see Fig. 1).

This research was supported by Wandercraft. Research involving human subjects was conducted under IRB No. 21-0693. The work of Kaveh Akbari Hamed was supported by the NSF under Grants 1923216 and 2024772. The work of Aaron D. Ames was supported by the NSF under Grant 1923239.

<sup>1</sup>K. Li is with the Department of Computation and Neural Systems, Caltech, Pasadena, CA 91125, USA, [kli5@caltech.edu](mailto:kli5@caltech.edu).

<sup>2</sup>J. Kim and A. D. Ames are with the Department of Mechanical and Civil Engineering, Caltech, Pasadena, CA 91125, USA, [{jeeseop, ames}@caltech.edu](mailto).

<sup>3</sup>X. Xiong is with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA, [xiaobin.xiong@wisc.edu](mailto:xiaobin.xiong@wisc.edu).

<sup>4</sup>K. Akbari Hamed is with the Department of Mechanical Engineering, Virginia Tech, VA 24061, USA, [kavehakbarihamed@vt.edu](mailto:kavehakbarihamed@vt.edu).

<sup>5</sup>Y. Yue is with the Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125, USA, [yyue@caltech.edu](mailto:yyue@caltech.edu).

<sup>6</sup>A. D. Ames is with the Department of Control and Dynamical Systems, Caltech, Pasadena, CA 91125, USA, [ames@caltech.edu](mailto:ames@caltech.edu).

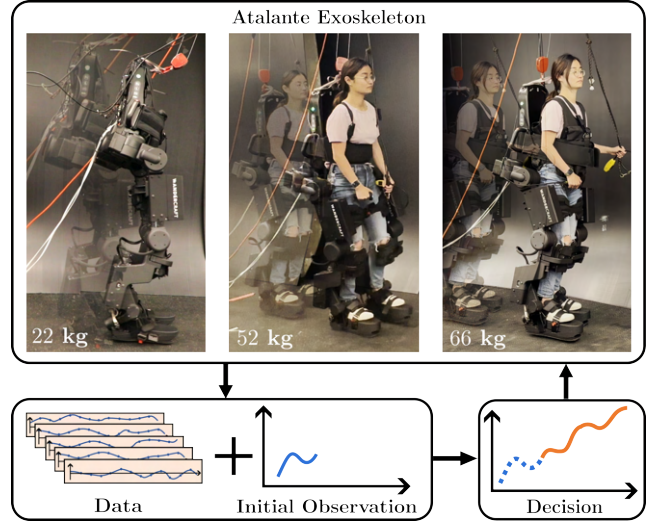


Fig. 1: Illustration of data-driven predictive control for bipedal locomotion on lower-body exoskeleton Atalante with various payloads.

Existing approaches of synthesizing motion for bipedal locomotion usually involve solving trajectory optimization problems using either the robot’s full or reduced-order model to find feasible trajectories given the desired walking speed. Methods such as hybrid zero dynamics utilize the full dynamics of the system, offering the advantage of generalizability across platforms and behaviors. It has achieved stable locomotion both on bipedal robots and robotics assistive devices [5], [6], [7], [8]. However, they are computationally demanding, often relying on large-scale offline trajectory optimization, and is prone to convergence issues when seeking stable periodic orbits. For computational benefits, synthesizing motion using a reduced-order model (ROM) such as linear inverted pendulum (LIP) and its variants are appealing since the planning problem depends only on a linear system [9], [10], [11]. Nevertheless, these methods often entail specific model assumptions, such as maintaining a constant center of mass height and zero angular momentum. Choosing the appropriate ROM necessitates careful consideration of the unique characteristics of the robotic platform in use while mitigating any potential discrepancies introduced by these assumptions when designing the desired behaviors.

With the advancement in modern computational power, pure data-driven approaches, such as reinforcement learning, have offered a model-free approach to train controllers by exploiting large amounts of data from simulators [12]. While offering robustness, these approaches often require extensive training data and are sensitive to reward design. This motivates leveraging data-driven approaches to learn a

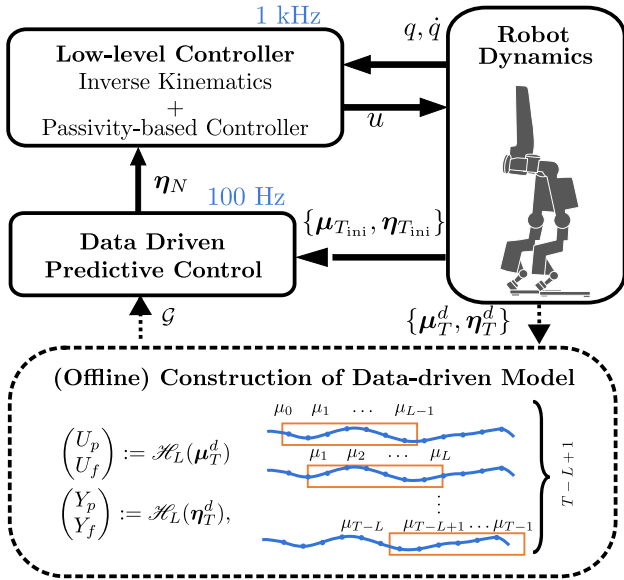


Fig. 2: Overview of the proposed layered control framework composed of the DDPC as a planner with constructed data-driven model and low-level controller.

more accurate representation of the system dynamics, such as learning the residual dynamics either using Gaussian process [13] or deep neural network [14], to use in conjunction with classic control methods. In the context of locomotion, due to the high DoFs, to allow for online planning capability, existing work focuses on achieving robust locomotion [15], [16] via learning a reduced-order representation of the system to mitigate model mismatches.

Another data-driven methodology, representing dynamics system using behavioral systems theory [17], has been introduced and proven to be an effective control method for linear time-invariant (LTI) systems. When this methodology is incorporated with predictive control framework, it is referred to as either data-enabled predictive control (DeePC) [18] or data-driven predictive control (DDPC). This approach not only has relatively mitigated computation cost and but also demonstrated its efficacy in controlling the underactuated quadrupedal robot [19] and the multiple quadrupedal robot systems characterized by highly nonlinear interconnected dynamics [20], [21]. Unlike most modern legged robots that operated effectively under the assumption of having negligible legs [22], [23], Atalante has a significant portion of its mass concentrated in the legs (see Fig. 1). As a result, we are interested in proposing a data-driven dynamic model and utilizing it for trajectory planning to address these challenges.

This paper presents a layered framework utilizing data-driven predictive control (DDPC) for trajectory planning and control of the lower-body exoskeleton Atalante as illustrated in Fig. 2. The primary contributions of this study are three-fold. First, we propose a data-driven dynamic model employing behavioral systems theory with time-domain trajectories of the center of mass (CoM) and center of pressure (CoP), inspired by the LIP model for bipedal locomotion. Second, we design a trajectory planner based on convex DDPC with the proposed data-driven model at the higher level

of the proposed layered framework. At the lower level of the framework, the controller incorporates inverse kinematics and passivity-based controllers to translate the planned trajectory from DDPC into the full-order model of the lower-body exoskeleton (Fig. 3a). Third, we conduct experimental evaluations of the proposed data-driven layered framework through numerical simulations and hardware demonstrations. Comparative analysis in simulations demonstrates that the proposed data-driven layered framework effectively stabilizes bipedal gaits at higher speeds, in contrast to the traditional model predictive control (MPC)-based planner for the LIP model. Furthermore, hardware experiments with different kinematics and payloads showcase the ability of this framework to account for user variability as well as the robustness under deviations from nominal data-driven model.

## II. PRELIMINARIES

Considering a robotic system with an  $n$ -dimensional configuration space, we denote the generalized coordinates by  $q = \text{col}(q_b, q_a) \in \mathcal{Q} \subset \mathbb{R}^n$ , where “col” denotes the column operator,  $q_b \in SE(3)$  represents the floating-base coordinates and  $q_a \in \mathbb{R}^m$  represents the actuated DoFs. The control input is denoted by  $u \in \mathbb{R}^m$ . The full-order state is denoted by  $x = \text{col}(q, \dot{q}) \in T\mathcal{Q}$ , where  $T\mathcal{Q}$  represents the tangent bundle of the configuration manifold.

### A. Models of Bipedal Locomotion

*Full-Order Model:* Bipedal locomotion can be described as a hybrid dynamic system consisting of domains with continuous dynamics and discrete transitions due to impact events [5]. In this work, we only consider a single domain with a single support phase where the stance foot is completely flat on the ground. We assume the ground height is always zero. The single support domain  $\mathcal{D}$  and guard  $\mathcal{S}$  of the system is defined as  $\mathcal{D} = \{x \in T\mathcal{Q} : p_{\text{sw}}^z(q) \geq 0\}$  and  $\mathcal{S} = \{x \in T\mathcal{Q} : p_{\text{sw}}^z(q) = 0, \dot{p}_{\text{sw}}^z(q) < 0\}$ , where  $p_{\text{sw}}(q)$  denotes the swing foot position, and the superscript denotes the axis we are considering (i.e., swing foot vertical position). The resulting hybrid system  $\mathcal{H}$  can be defined as follows:

$$\mathcal{H} : \begin{cases} \dot{x} = f(x) + g(x)u & x \in \mathcal{D} \setminus \mathcal{S}, \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S}, \end{cases} \quad (1)$$

where (1) denotes the continuous full-order and Lagrangian dynamics,  $\Delta : \mathcal{S} \rightarrow \mathcal{D}$  represents the discrete event at foot strike, and the superscripts “-” and “+” stand for the instants before and after the impact event, respectively.

*Reduced-Order Inverted Pendulum Model:* Since we only consider flat-footed walking in this work, we can assume all contact points, denoted by  $p_i$  for the contacting index  $i$ , are on the flat ground, that is,  $p_i^z = 0$ . By constraining angular momentum to be constant (i.e.,  $\dot{L}_{\text{com}} = 0$ ), one can simplify the Newton-Euler equation for Centroidal dynamics [24] as follows:

$$a_{\text{com}}^{\{x,y\}} = \frac{a_{\text{com}}^z + g}{P_{\text{com}}^z} \left( p_{\text{com}}^{\{x,y\}} - p_{\text{cop}}^{\{x,y\}} \right), \quad (3)$$

where  $g$  is the gravitational constant,  $\mathbf{p}_{\text{cop}}^{\{x,y\}} := \frac{\sum_i f_i^z p_i^{\{x,y\}}}{\sum_i f_i^z}$  represents the CoP position,  $\mathbf{p}_{\text{com}}$  denotes the CoM position, and  $a_{\text{com}}$  represents the CoM acceleration. Since the contact forces  $f_i$  are unilateral with  $f_i^z \geq 0$ , the CoP has to lie inside the convex hull of the contact points, that is,  $\mathbf{p}_{\text{cop}} \in \text{conv}\{p_i\}$ . If we additionally assume constant vertical CoM position  $\mathbf{p}_{\text{com}}^z$ , one could simplify (3) to obtain the classic LIP model that is widely used for motion synthesis in bipedal locomotion [25]. Motivated by the relation between the CoP inputs and CoM outputs in the LIP model, we will develop a data-driven model in the next sections.

### B. Behavioral Systems Theory

This section briefly reviews some of the fundamental results of the behavioral systems theory. Let us consider a state representation of a discrete-time LTI system as follows:

$$\begin{aligned} \theta(t+1) &= A\theta(t) + B\mu(t) \\ \eta(t) &= C\theta(t) + D\mu(t), \end{aligned} \quad (4)$$

where  $\theta(t) \in \mathbb{R}^\beta$ ,  $\mu(t) \in \mathbb{R}^\kappa$ ,  $\eta(t) \in \mathbb{R}^\nu$  represent the state vector, control inputs, and outputs, respectively, at time  $t \in \mathbb{Z}_{\geq 0} := \{0, 1, \dots\}$ , and  $A \in \mathbb{R}^{\beta \times \beta}$ ,  $B \in \mathbb{R}^{\beta \times \kappa}$ ,  $C \in \mathbb{R}^{\nu \times \beta}$ ,  $D \in \mathbb{R}^{\nu \times \kappa}$  denote the unknown state matrices. In behavioral systems theory, a dynamical system is defined as a 3-tuple  $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathcal{B})$ , where  $\mathbb{W}$  is a signal space and  $\mathcal{B} \in \mathbb{W}^{\mathbb{Z}_{\geq 0}}$  is the behavior. In contrast with classical systems theory with a particular parametric system representation such as that of (4), behavioral systems theory focuses on the subspace of the signal space where system trajectories live. Let  $\boldsymbol{\mu}_T := \text{col}(\mu_0, \mu_1, \dots, \mu_{T-1})$  be an input trajectory with length  $T \in \mathbb{N} := \{1, 2, \dots\}$  applied in  $\mathcal{B}$  with the corresponding output trajectory  $\boldsymbol{\eta}_T := \text{col}(\eta_0, \eta_1, \dots, \eta_{T-1})$ . We can construct the Hankel matrix with  $\boldsymbol{\mu}_T$  by concatenating trajectory with length  $L \in \mathbb{N}$  and  $T > L$  as follows:

$$\mathcal{H}_L(\boldsymbol{\mu}_T) := \begin{bmatrix} \mu_0 & \cdots & \mu_{T-L} \\ \vdots & \ddots & \vdots \\ \mu_{L-1} & \cdots & \mu_{T-1} \end{bmatrix} \in \mathbb{R}^{\kappa L \times (T-L+1)}.$$

Here we note that the Hankel matrix with  $\boldsymbol{\eta}_T$ ,  $\mathcal{H}_L(\boldsymbol{\eta}_T)$ , can be constructed analogously. The signal  $\boldsymbol{\mu}_T$  is said to be persistently exciting of order  $L$  if  $\mathcal{H}_L(\boldsymbol{\mu}_T)$  is of full row rank [26]. If a given data input-output (I-O) trajectory of the system, denoted by the pair  $(\boldsymbol{\mu}_T^d, \boldsymbol{\eta}_T^d)$ , is persistently exciting of order  $L+\beta$ , from Fundamental Lemma [26], [27], any trajectory of the LTI system can be constructed using a linear combination of the columns of the Hankel matrices. In particular, one can use the columns of the Hankel matrices to develop a data-driven model to predict the system's future behavior.

To make this notion more precise, let us take  $L$  as  $L = T_{\text{ini}} + N$ , where  $T_{\text{ini}}$  and  $N$  denote the estimation horizon and control horizon, respectively. The estimation horizon is used to estimate the system's initial state from past I-O measurements. The control horizon is used for the predictive controller. We can partition the Hankel matrices into the past

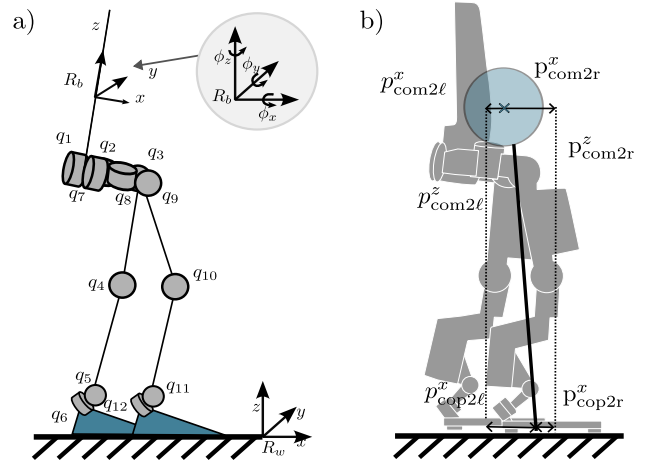


Fig. 3: a) Generalized coordinates for the lower-body exoskeleton Atalante. b) The input and output variables in the x-direction to be used for the Hankel matrix construction.

and future portions accordingly as follows:

$$\begin{bmatrix} U_p \\ U_f \end{bmatrix} := \mathcal{H}_L(\boldsymbol{\mu}_T^d) \quad \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} := \mathcal{H}_L(\boldsymbol{\eta}_T^d),$$

where  $U_p \in \mathbb{R}^{\kappa T_{\text{ini}} \times (T-L+1)}$ ,  $U_f \in \mathbb{R}^{\kappa N \times (T-L+1)}$ ,  $Y_p \in \mathbb{R}^{\nu T_{\text{ini}} \times (T-L+1)}$ ,  $Y_f \in \mathbb{R}^{\nu N \times (T-L+1)}$ . From the Fundamental Lemma [18], [26], [27], any new trajectory lies in the range space of the Hankel matrices, or equivalently, there exists  $\gamma \in \mathbb{R}^{T-L+1}$  such that

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} \gamma = \begin{bmatrix} \boldsymbol{\mu}_{T_{\text{ini}}} \\ \boldsymbol{\eta}_{T_{\text{ini}}} \\ \boldsymbol{\mu}_N \\ \boldsymbol{\eta}_N \end{bmatrix}, \quad (5)$$

where  $\boldsymbol{\mu}_{T_{\text{ini}}}$  and  $\boldsymbol{\eta}_{T_{\text{ini}}}$  denote the past portions of the I-O trajectories over the estimation horizon of  $T_{\text{ini}}$ . Similarly,  $\boldsymbol{\mu}_N$  and  $\boldsymbol{\eta}_N$  represent the predicted (i.e., future) portions of the I-O trajectories over the control horizon of  $N$ . Since the dimensionality of the  $\gamma$  vector in formulating predictive controllers is huge, we aim to remove  $\gamma$  from (5). One way is to have an offline approximation for  $\gamma$  using least squares similar to [19], [28] to obtain the data-driven model as

$$\boldsymbol{\eta}_N = Y_f \gamma = Y_f \underbrace{\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^\dagger}_{\mathcal{G}} \begin{bmatrix} \boldsymbol{\mu}_{T_{\text{ini}}} \\ \boldsymbol{\eta}_{T_{\text{ini}}} \\ \boldsymbol{\mu}_N \end{bmatrix}, \quad (6)$$

where  $(\cdot)^\dagger$  represents the pseudo inverse and  $\mathcal{G}$  denotes the data-driven state transition matrix over  $N$ -steps.

### III. DATA-DRIVEN MOTION PLANNER

This section aims to present the proposed DDPC-based trajectory planner at the high level of the control scheme for bipedal locomotion (see Fig. 2).

#### A. Construction of the Data-Driven Model

Assuming reasonable behavior for the actuated coordinates, the difficulty and complexity of the bipedal locomotion usually lie in the control and planning for the weakly

actuated or underactuated Centroidal states. In this context, designing the CoM trajectory encapsulated all the requisite DoFs' information, although their individual dynamics are not described explicitly. Hence instead of using the full-order states  $x$  for trajectory planning, we are focusing on the Centroidal states to construct the Hankel matrices. Specifically, we draw inspiration from the LIP model that considers the CoM and CoP of the robot. We consider a local representation of these states with respect to either the left or right foot frame as  $p_{\text{cop}2\ell/r}^{x,y} := p_{\text{cop}}^{x,y} - p_{\ell/r}^{x,y}$  and  $p_{\text{com}2\ell/r}^{x,y,z} := p_{\text{com}}^{x,y,z} - p_{\ell/r}^{x,y,z}$  as shown in Fig. 3b.

An intuitive choice would be using the CoM and CoP trajectories in the stance-foot frame. However, this choice would create discontinuity during domain switches, posing challenges for planning through impact. To prevent the state space from monotonically increasing during forward walking and to maintain continuity in the trajectory planning, we adopt a redundant representation where the trajectory of the CoM and CoP with respect to both stance and swing feet are being considered. Hence, the input  $\mu \in \mathbb{R}^4$  and output  $\eta \in \mathbb{R}^6$  of the data-driven model are chosen as follows:

$$\eta := \text{col}(p_{\text{com}2\ell}^{x,y,z}, p_{\text{com}2r}^{x,y,z}) \quad \mu := \text{col}(p_{\text{cop}2\ell}^{x,y}, p_{\text{cop}2r}^{x,y}).$$

Notably, with this redundant formulation, we also provide trajectories in the following domain throughout the prediction horizon for the low-level controller to track in case of unexpected early or late impact. This is especially advantageous in scenarios involving uncertain impact timings or unexpected communication delays, significantly enhancing the framework's practical applicability. We do not include velocity terms in our representation as the velocity and acceleration information is implicitly captured in the position trajectory in the Hankel matrix. We remark that the proposed data-driven model, inspired by the LIP model, captures more comprehensive information about the system with implicit consideration of swing foot trajectory and the effect of the low-level whole-body controller on the system dynamics. A comparative analysis of the performance between these two models and their corresponding planner architectures will be presented in Section V.

### B. DDPC Algorithm for Trajectory Optimization

We are now positioned to present the DDPC-based trajectory planner for optimizing the I-O (i.e., CoP and CoM) trajectories. The real-time DDPC planner is formulated as the following strictly convex quadratic program (QP)

$$\begin{aligned} \min_{(\mu_N, \eta_N)} \quad & \sum_{k=0}^{N-1} (\|\eta_k - r_k^\eta\|_Q^2 + \|\mu_k - r_k^\mu\|_R^2) \\ \text{subject to} \quad & \eta_N = \mathcal{G} \begin{bmatrix} \mu_{T_{\text{ini}}} \\ \eta_{T_{\text{ini}}} \\ \mu_N \end{bmatrix} \quad (\text{Data-Driven Model}) \quad (7) \\ & \mu_k \in \mathcal{U}, \quad \eta_k \in \mathcal{P}, \quad k = 0, \dots, N-1, \end{aligned}$$

where  $r^\eta := \text{col}(r_0^\eta, \dots, r_{N-1}^\eta)$  represents a reference output (i.e., CoM) trajectory,  $r^\mu := \text{col}(r_0^\mu, \dots, r_{N-1}^\mu)$  denotes

a reference input (i.e., CoP) trajectory, and  $(\mu_{T_{\text{ini}}}, \eta_{T_{\text{ini}}})$  represents the pair of past actual I-O trajectories (i.e., feedback to the DDPC). In addition,  $\mathcal{U} \subseteq \mathbb{R}^4$  and  $\mathcal{P} \subseteq \mathbb{R}^6$  denotes the input and output feasibility sets, respectively. Finally,  $Q \in \mathbb{R}^{6 \times 6}$  and  $R \in \mathbb{R}^{4 \times 4}$  are chosen as positive definite weighting matrices.

## IV. LAYERED CONTROL FRAMEWORK FOR ATALANTE

In this section, we introduce the remaining components and details for practically implementing the entire layered control framework used to realize locomotion on the lower-body exoskeleton Atalante (see Fig. 1). Atalante contains 12 actuated DoFs as illustrated in Fig. 3a, with three actuators on the hip, one on the knee, and two on the ankle for each leg. The device weighs 82 kg and has adjustable thigh and shin length to account for the varying heights of users. Depending on the physical parameters of the users, the largest kinematically feasible step length in  $x$  direction that the device is capable to achieve for flat-footed walking is less than 0.2 m.

### A. Trajectory Planner with DDPC

We generate a reference trajectory with the open-source toolbox FROST [29] using the full-order system to account for kinematics and dynamics feasibility. This trajectory is described by 7-th order Bézier polynomials with the coefficient matrix of  $\alpha = [\alpha_{\text{com}}, \alpha_\phi, \alpha_{\text{sw}}]$ , where  $\alpha_{\text{com}}$  describes the CoM trajectory,  $\alpha_\phi$  described the pelvis orientation and  $\alpha_{\text{sw}}$  described the swing foot position and orientation. The Bézier polynomials are evaluated based on a time-based phase variable. Specifically, considering a desired step duration  $t_d$  and the initial time at the beginning of the domain  $t_0$ , we can calculate the phase variable  $\tau_k = \frac{t_k - t_0}{t_d}$  at time point  $t_k$ . The reference for the CoM trajectory is then determined by  $r_k^\eta = \text{col}(p_{\text{com}}^{x,y,z}(\alpha, \tau_k) - p_\ell^{x,y,z}(\alpha, \tau_k), p_{\text{com}}^{x,y,z}(\alpha, \tau_k) - p_r^{x,y,z}(\alpha, \tau_k))$ . In addition, the reference CoP trajectory is generated via  $r_k^\mu = \text{col}(-p_{\text{sw}}^{x,y}(\alpha, \tau_k), 0, 0)$  for the right stance and  $r_k^\mu = \text{col}(0, 0, -p_{\text{sw}}^{x,y}(\alpha, \tau_k))$  for the left stance.

Without explicit guidelines to construct the Hankel matrix for nonlinear systems, we empirically determine the hyperparameters of the DDPC algorithm via a grid search over the space of the discrete-time interval between nearby points  $\delta_t \in [0.01, 0.03]$ , the trajectory length of  $T \in [50, 600]$ , the initial trajectory length of  $T_{\text{ini}} \in [5, 50]$ , and the control horizon of  $N \in [10, 300]$ . This search space is constructed considering the computation speed, low-level controller frequency, noisy level of the data, and the amount of data required. We choose  $T = 400$ ,  $T_{\text{ini}} = 10$ ,  $N = 20$ , and  $\delta_t = 0.02$  with a selection criteria on the accuracy of the least-square approximation over some unseen trajectory. In total, 8 s of data are used for the Hankel matrix construction, and a trajectory for 0.4 s is planned. This means our DDPC-based trajectory planner primarily acts as a short-term regulator to stabilize the system. We remark that even though  $p_{\text{com}}^z$  is planned, it is not being used by the low-level controller but only used as part of the states to determine system dynamics. The trajectories are planned at 100 Hz, which

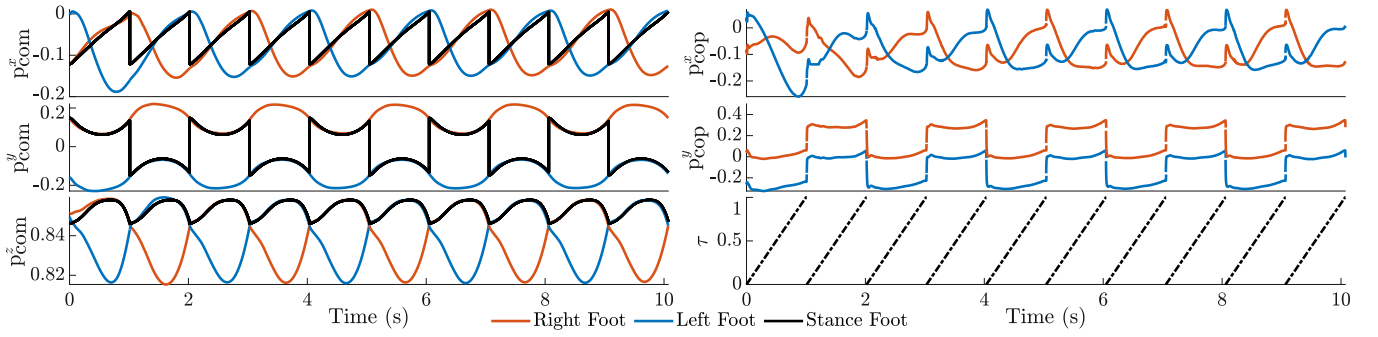


Fig. 4: One set of the planned CoM and CoP trajectories from the DDPC planner and the tracked trajectory in simulation in right foot frame, the left foot frame trajectories, and the Stance foot. The stance foot frame trajectory in black is generated from DDPC. The corresponding phase variables are plotted in the dashed line.

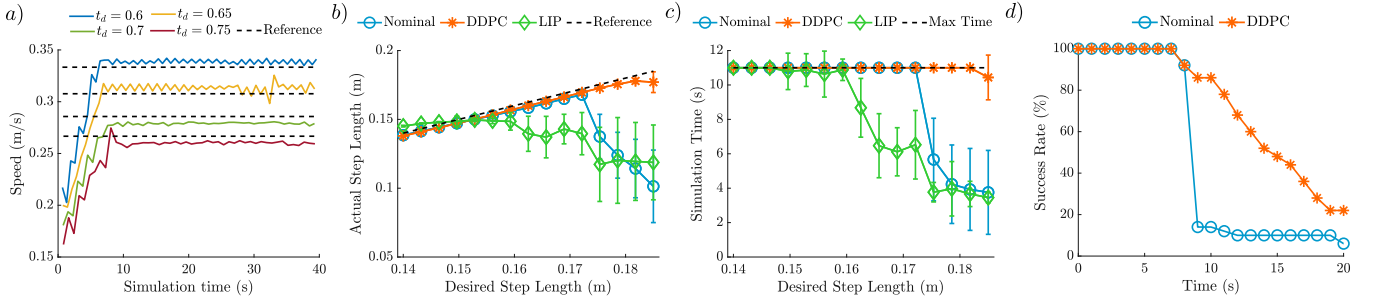


Fig. 5: Simulation comparison over nominal indicated by blue circles, DDPC indicated by orange stars, and MPC indicated by green diamonds. a) DDPC planner planning trajectories for increasing desired speed, capped at maximum step length 0.2 m at different step duration  $t_d$ . b) tracking performance over different desired step length with the same step duration. The dashed line indicate the ideal performance. Error bar indicates the standard deviation over 50 models. c) Simulation time before robot falling for the tracking performance comparison. Error bar indicates the standard deviation over 50 models. The maximum simulation time is 11 s, indicated by the horizontal dash line. d) Comparison of Nominal and DDPC under time-varying perturbation applied on the negative  $x$  direction.

is faster than the interval  $\delta_t$  specified. However, since the reference generation is based on the current domain and phase variable, the trajectory planner still has the ability to regulate the system behavior during this interval till the next time sample at which  $\mu_{T_{ini}}$  and  $\eta_{T_{ini}}$  are updated.

### B. Output Synthesis

The desired walking behavior is encoded by the task space output  $\mathbf{y} = \mathbf{y}^{\text{act}} - \mathbf{y}^{\text{des}}$ , where  $\mathbf{y}^{\text{act}} \in \mathbb{R}^{12}$  and  $\mathbf{y}^{\text{des}} \in \mathbb{R}^{12}$  are the actual and desired outputs, respectively. In particular, we choose the following outputs for the system:

$$\mathbf{y}^{\text{act}} = \begin{bmatrix} \mathbf{p}_{\text{com2st}}^{x,y,z}(q) & \phi_{\text{pelv}}^{x,y,z}(q) & \mathbf{p}_{\text{sw}}^{x,y,z}(q) & \phi_{\text{sw}}^{x,y,z}(q) \end{bmatrix}$$

$$\mathbf{y}^{\text{des}} = \begin{bmatrix} \mathbf{p}_{\text{com2st}}^{x,y,z} & \phi_{\text{pelv}}^{x,y,z}(\alpha) & \mathbf{p}_{\text{sw}}^{x,y,z}(\alpha, \lambda^{x,y}) & \phi_{\text{sw}}^{x,y,z}(\alpha) \end{bmatrix},$$

where the desired COM position  $\mathbf{p}_{\text{com2st}}^{x,y}$  is generated by the high-level DDPC planner, and the other desired components are taken as Bézier polynomials with the coefficient matrix of  $\alpha$  and the step length of  $\lambda^{x,y}$ . More specifically, the coefficients of pelvis orientation  $\phi_{\text{pelv}}$  and swing foot orientation  $\phi_{\text{sw}}$ , and  $z$ -height of CoM and swing foot trajectory are fixed and from the aforementioned reference trajectory. The swing foot  $x, y$  trajectories are determined by Bézier polynomials connecting the swing foot position at the beginning of the domain (i.e., post-impact state) and the desire foot targets, i.e.,  $\mathbf{p}_{\text{sw}}^{x,y}(\tau) = (1 - \beta(\tau)) \mathbf{p}_{\text{sw}}(q^+) + \beta(\tau) \lambda^{x,y}$ , where  $\beta$  is a phase-based weighting function.

### C. Low-Level Feedback Controller

Our low-level controller, implemented in C++, receives trajectories for the CoM position target from the DDPC-based trajectory planner. Given that the controller operates at 1 kHz, much faster than the planner's replan frequency and the discrete-time interval, CoM target positions and velocities for each control tick is obtained via linear interpolation. Subsequently, we employ a Newton-Raphson numerical inverse kinematics (IK) algorithm to calculate the desired joint targets, which are then tracked using a passivity-based control method [30]. The kinematics and dynamics evaluation is performed with Pinocchio [31]. To account for uncertainty in state estimation and impact time, we switch to the next stance domain only when the swing foot ground reaction force exceeds some pre-defined threshold instead of time-based switching.

## V. EXPERIMENTAL VALIDATIONS

In this section, we present the numerical simulation and hardware experiment results for our proposed framework. The experiment video could be found in [32].

### A. Simulation Results

We validate the effectiveness of our framework with numerical simulations in MuJoCo [33]. To account for uncertainty in mass distribution estimation for human users,

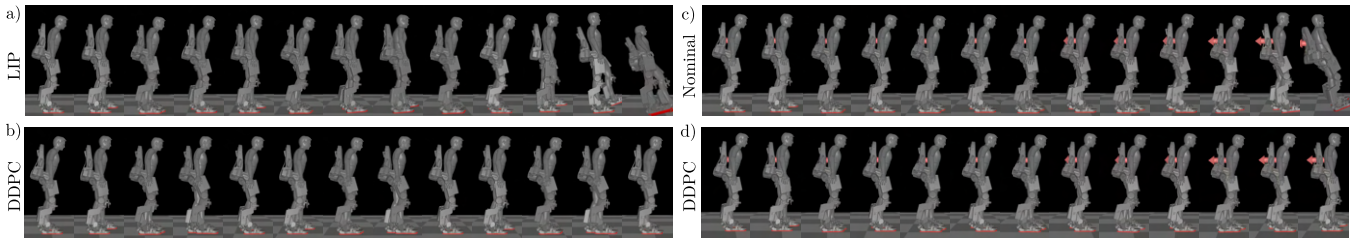


Fig. 6: Gait tiles for simulation. Simulations with a) LIP-based MPC and b) DDPC controllers for walking at the speed of 0.16 (m/s), c) Nominal trajectory and d) DDPC under time-varying perturbations.

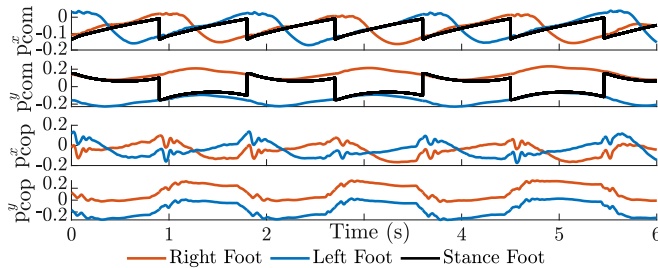


Fig. 7: Desired trajectories from the DDPC-based trajectory planner (black solid line) and actual evolving CoM/CoP states from the hardware experiment.

we generated 50 randomized models with identical user total mass by varying the CoM offset and inertia properties, resulting in a  $p_{\text{com}}$  ranging from  $[-0.122, -0.106]$  m in nominal standing configuration. The low level controller is using the information from the same nominal model constructed with parameters from [34].

**Tracking Performance:** We compare tracking performance over these randomly generated models. For each model, simulation data with different desired step lengths ranging from 0.1 to 0.15 m were collected to construct the  $\mathcal{G}$  matrix. An example trajectory generated by the DDPC planner and the corresponding actual CoM and CoP states are shown in Fig. 4. Moreover, DDPC is able to achieve stable walking in various speed as described in Fig. 5a. Additionally, we implemented an MPC planner based on the LIP dynamics for comparative analysis. This MPC shares a similar problem structure to that described in (7), but it substitutes the Hankel matrix trajectory constraints with those of the LIP dynamics. Considering the kinematic limits and feasibility concerns, we opted for a 0.8 m CoM z-position in the LIP model (as opposed to 0.85 m) and set  $\delta_t = 0.01$  s to discretize the LIP dynamics. The time horizon for the MPC is also chosen as  $N_{\text{LIP}} = 300$ .

The DDPC planner’s tracking performance was evaluated against that of the MPC and a nominal trajectory for desired walking speeds between 0.14 and 0.19 m/s, with a constant step duration of 1 s. As shown in Fig. 5b, the DDPC planner reliably achieves close tracking of the desired step length. As the desired speed increases by increasing the desired step length, the validity of LIP dynamics starts to degrade. In particular, the LIP MPC’s performance begins to decline at speeds above 0.16 m/s, and the nominal trajectory cannot be appropriately tracked beyond 0.17 m/s, as described in Fig. 5b. Furthermore, the DDPC planner maintains the system’s stability for a longer duration than the other controllers as

demonstrated in Fig. 5c. Here, we define a robot maintaining CoM above a certain threshold throughout the maximum simulation time as a success. It is a failure if QP or IK becomes infeasible or CoM falls below a specified threshold. An example of failure case is shown in Fig. 6a for MPC. Since the DDPC-generated trajectory is based on the feasible trajectory used to construct  $\mathcal{G}$ , it tends to run into kinematics issues less frequently than physics-based reduced-order model like LIP, where no full-order model system information is exposed to the planner.

**Adaptation to Perturbation:** We further evaluate the DDPC’s robustness against time-varying external disturbances (see Fig. 6c and 6d). Specifically, we continuously apply an external perturbation force to the torso’s negative  $x$  direction. This direction is chosen because the device is more sensitive towards perturbation applied along this axis with its fairly negative nominal CoM position. The initial perturbation force was set at 5 N with a discrete increase of 3 N every 3 s. To adapt to this time-varying perturbation, the Hankel matrix is updated online. The choice of the Hankel matrix size requires a balance between satisfying persistently exciting requirements and using obsolete data. The simulation’s first 5 s were used for data collection, which was a smaller trajectory length  $T = 250$ , and  $\mathcal{G}$  is reconstructed every 1.5 s. The least-square approximation step is completed within 70 – 100 ms. We ran the same setup for the aforementioned 50 randomly generated models and compared it with the nominal trajectory at 0.13 m/s desired speed. The DDPC has a higher success rate, defined by the percentage of the models being upright, compared to the nominal trajectory (see Fig. 5d).

## B. Hardware Results

The same C++ low-level controller used in simulation, with additional code to interface with Wandercraft API, is run directly on the Atalante onboard computer (i5-4300U CPU @1.90GHz with 8GB RAM at 1kHz). The DDPC planner is run on an external PC (i7-8700K CPU @3.70GHz) communicating with the onboard computer via a UDP network. To account for package delay, a segment of the desired trajectory is sent, and the low-level controller finds the closest discrete target point given the current phase variable. As a result, the low-level controller receives planned trajectory with the upcoming time stamps and could handle the delay caused by planning computation time and UDP communication. An example planned trajectory of CoM and CoP under this setup is shown in Fig. 7.

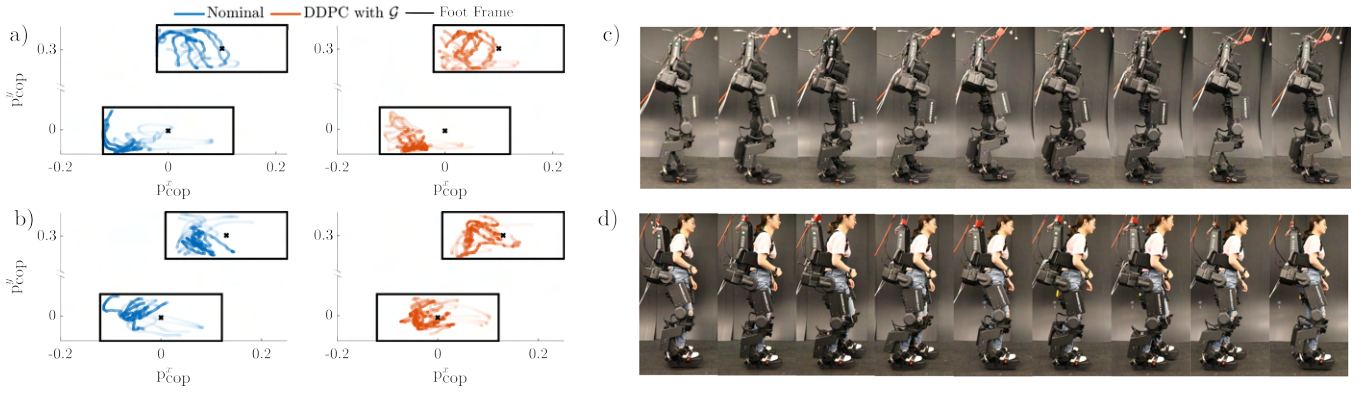


Fig. 8: The system evolution with the data-driven layered framework is shown in orange, and the system evolution with nominal trajectory is shown in blue. a) Experiment result for exoskeleton carrying 20 kg of payload. CoP position in the foot frame for 5 left stance foot steps and 5 right stance foot steps. b) CoP position for exoskeleton with user inside. c) Gait tiles with the DDPG planner for the 20 kg payload experiment. d) Gait tiles with the DDPG planner for the experiment with user.

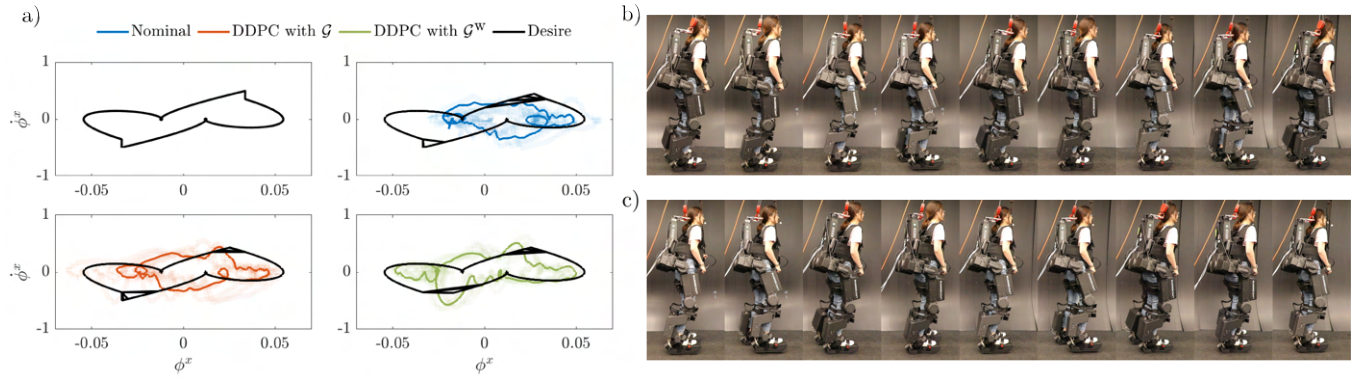


Fig. 9: a) Phase portrait over pelvis roll  $\phi_x$  and  $\dot{\phi}_x$  for hardware experiment with user wearing additional weight for nominal, DDPG with  $\mathcal{G}$ , DDPG with  $\mathcal{G}^w$  b) Gait tiles for experiment with DDPG controller with  $\mathcal{G}$  with additional weight c) Gait tiles for experiment with DDPG controller with  $\mathcal{G}^w$  with additional weight

**Different User Settings:** We conducted two sets of experiments to test the framework with different payloads and kinematics as this platform is designed to be used by different users. The data collection part for the hardware experiment is similar to that of the simulation setup for tracking performance comparison with gaits at different step lengths. The first set of experiments is conducted with the exoskeleton carrying 20 kg of payload with the link lengths of a subject of 1.74 m. DDPG planner is able to regulate the CoP position to a more centered position as depicted in Fig. 8a and 8b. A different set of hyperparameters with  $T = 800$ ,  $T_{ini} = 20$ ,  $N = 100$ , and  $\delta_t = 0.015$  is used. The second experiment is conducted with a user of height 1.63 m and 52 kg with the same hyperparameter to construct the Hankel Matrix as in tracking performance case. The gait tiles for the experiment with different payloads are shown in Fig. 8c and 8d.

**Uncertain User Mass:** To investigate the effect of the planner’s capability to handle the uncertain mass of the user or in case of the user wanting to carry additional payload, we also conducted a set of experiments with the user carrying an additional 14 kg of weight (see Fig. 9). We test both DDPGs with  $\mathcal{G}$  constructed from data without carrying the additional weight and with  $\mathcal{G}^w$  which is constructed with

trials where the user is carrying additional weight. From the phase portrait described in Fig. 9a, we could see that the DDPG with  $\mathcal{G}^w$  resembles a desired limit cycle compared to the DDPG with  $\mathcal{G}$  and the nominal controller. The gait tiles for the experiments with  $\mathcal{G}$  and  $\mathcal{G}^w$  are shown in Fig. 9b and 9c, respectively. We also evaluate the cumulative tracking error over  $t_{total} = 10$  s for output other than the CoM position and yaw related, denoted by  $\mathbf{y}_{part}$ , as the CoM positions are different across controllers and the yaw related output is not well approximated due to IMU drifting via evaluating  $e = \int_0^{t_{total}} \|\mathbf{y}_{part}^{act}(t) - \mathbf{y}_{part}^{des}(t)\|^2 dt$ . This value for the DDPG with  $\mathcal{G}$ ,  $\mathcal{G}^w$ , and nominal is 0.4627, 0.4319, and 0.5181, respectively.

## VI. CONCLUSIONS

This paper successfully demonstrated the DDPG framework’s application on lower-body exoskeleton, both in simulations and on hardware. Through detailed simulation analyses, the DDPG framework proved its effectiveness in stabilizing the system, surpassing traditional physics-based template models such as LIP, particularly at increased desired speeds. The framework’s robustness was also validated on hardware, showcasing its ability to accommodate model discrepancies beyond the initial model used for data collection. Further-

more, we introduced a time-varying perturbation in the simulation while updating the transition matrix online. Although further research is necessary to refine the online update process of the Hankel matrices systematically, these results underscore the framework's capacity to adapt to changing environments. However, it is crucial to note that the DDPC planner's performance is intrinsically linked to the trajectory used to construct the Hankel matrices. It can only enhance and build upon the capabilities of the nominal controller used for data collection. Moreover, considering the inherent limitations in the actuation of the CoM horizontal position, future studies will explore how to extend the proposed framework to incorporate foot placement and step timing planning to enhance stability and performance. Additionally, further investigation into incorporating more information on user movement would be beneficial.

### ACKNOWLEDGMENTS

The authors would like to thank the entire Wandercraft team for their continued guidance and technical support with Atalante.

### REFERENCES

- [1] 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.
- [2] 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.
- [3] M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. W. Burdick, and A. D. Ames, "Preference-based learning for exoskeleton gait optimization," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 2351–2357.
- [4] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.
- [5] 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.
- [6] J. P. Reher, A. Hereid, S. Kolathaya, C. M. Hubicki, and A. D. Ames, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot durus," in *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 400–415.
- [7] 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.
- [8] K. Li, M. Tucker, R. Gehlhar, Y. Yue, and A. D. Ames, "Natural multicontact walking for robotic assistive devices via musculoskeletal models and hybrid zero dynamics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4283–4290, 2022.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1620–1626.
- [10] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5134–5140.
- [11] J. Engelsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *Ieee transactions on robotics*, vol. 31, no. 2, pp. 355–368, 2015.
- [12] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2811–2817.
- [13] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.
- [14] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural lander: Stable drone landing control using learned dynamics," in *2019 international conference on robotics and automation (icra)*. IEEE, 2019, pp. 9784–9790.
- [15] M. Dai, X. Xiong, J. Lee, and A. D. Ames, "Data-driven adaptation for robust bipedal locomotion with step-to-step dynamics," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8574–8581.
- [16] X. Xiong, Y. Chen, and A. D. Ames, "Robust disturbance rejection for robotic bipedal walking: System-level-synthesis with step-to-step dynamics approximation," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 697–704.
- [17] I. Markovskiy, J. C. Willems, S. Van Huffel, and B. De Moor, *Exact and approximate modeling of linear systems: A behavioral approach*. SIAM, 2006.
- [18] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 307–312.
- [19] R. T. Fawcett, K. Afsari, A. D. Ames, and K. Akbari Hamed, "Toward a data-driven template model for quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7636–7643, 2022.
- [20] R. T. Fawcett, L. Amanzadeh, J. Kim, A. D. Ames, and K. Akbari Hamed, "Distributed data-driven predictive control for multi-agent collaborative legged locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9924–9930.
- [21] R. T. Fawcett, A. D. Ames, and K. Akbari Hamed, "Distributed planning of collaborative locomotion: A physics-based and data-driven approach," *IEEE Access*, vol. 11, pp. 128 369–128 382, 2023.
- [22] X. Xiong and A. Ames, "3-d underactuated bipedal walking via h-hip based gait synthesis and stepping stabilization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2405–2425, 2022.
- [23] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.
- [24] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer handbook of robotics*. Springer, 2016, pp. 1203–1234.
- [25] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for biped walking," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 31–37.
- [26] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.
- [27] J. C. Willems, P. Rapisarda, I. Markovskiy, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.
- [28] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *IEEE Conference on Decision and Control*, 2019, pp. 8130–8135.
- [29] 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)*. IEEE, 2017, pp. 719–726.
- [30] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 12, p. 121006, 2022.
- [31] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.
- [32] Supplementary video: <https://www.youtube.com/watch?v=rgs36YXRb4I>.
- [33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [34] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & sons, 2009.