Human-Inspired Multi-Contact Locomotion with AMBER2

Hui-Hua Zhao
Mechanical Engineering
Texas A & M University
College Station, USA
huihuazhao@tamu.edu

Wen-Loong Ma
Mechanical Engineering
Texas A & M University
College Station, USA
wenlongma@tamu.edu

Michael B. Zeagler
Mechanical Engineering
Texas A & M University
College Station, USA
wallowinginfun@tamu.edu

Aaron D. Ames
Mechanical Engineering
Texas A & M University
College Station, USA
aames@tamu.edu

ABSTRACT

This paper presents a methodology for translating a key feature encoded in human locomotion–multi-contact behavior–to a physical 2D bipedal robot, AMBER2, by leveraging novel controller design, optimization methods, and software structures for the translation to hardware. This paper begins with the analysis of human locomotion data and uses it to motivate the construction of a hybrid system model representing a multi-contact robotic walking gait. By again looking to human data for inspiration, human-inspired controllers are developed and used in the formulation of an optimization problem that yields stable human-like multi-domain walking in simulation. These formal results are translated to hardware implementation via a novel dynamic trajectory generation strategy. Finally, the specific software structures utilized to translate these trajectories to hardware are presented. The end result is experimentally realized stable robotic walking with remarkably human-like multi-contact foot behaviors.

1. INTRODUCTION

Cyber-physical systems involve tight coupling between computation and the physical world; as such, humans provide a prime example of such a system. Nowhere is this more prevalent than in the simple act of locomoting. During the course of a step, humans undergo changes in phase [2, 6], i.e., a change in contact points with the environment, including a heel-lift and toe strike; for an instance, the heel lift at the single support domain yields higher foot clearance and lets the human utilize the rotation momentum efficiently to save energy. In the process of progressing through these phases, the physics of the system change along with the degrees of actuation. While dealing with these dynamic changes is seemingly effortless for a human, imbuing robotic systems with these same capabilities is fraught with complexity in terms of computation and control. The goal of this paper is to take the first steps towards addressing these problems with the objective of human-like multi-contact bipedal robotic walking.

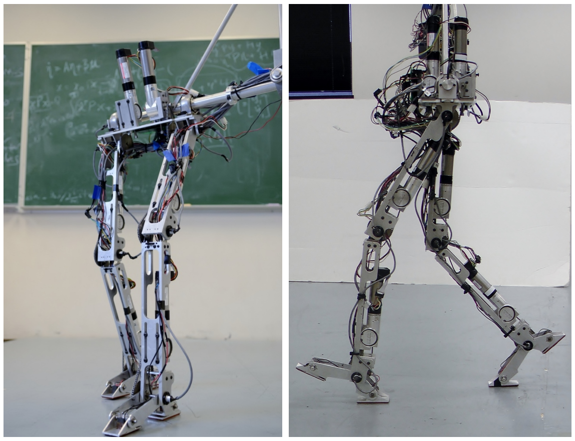


Figure 1: The bipedal robot AMBER2 (left) is constructed with the specific goal of multi-contact locomotion as indicated by the design of the feet (right).

Robotic walking has been studied from a variety of viewpoints, most of which are aimed at the ultimate goal of achieving human-like locomotion capabilities on bipedal robots. Most approaches try to reduce the complexity of the problem through simplifying assumptions revolving around the ZMP point [13]; yet, by nature of the fact that these methods require the center of pressure to lie within the foot, they require flat foot locomotion. As a result, their fundamental assumptions tend to preclude the human-like behavior of multi-contact locomotion. On the other end of the spectrum, formal methods for achieving bipedal locomotion have been presented in the case of underactuated walking robots through the use of hybrid zero dynamics [11], some of which have been adopted to multi-contact walking in simulation [7]. Finally, recent work from the coauthors has looked toward human-locomotion for inspiration for the synthesis of walking controllers, both in the case of under [15] and full actuation [3, 16]. Noticeably lacking from existing methods from any of these perspectives is a formal way to generate multi-contact locomotion in a manner that is both formally correct, along with being physically realizable.

This objectives of this paper are twofold: (a) present a formal means by which multi-contact robotic walking can be achieved, and (b) realize this approach experimentally on AMBER2 (see Fig. 1) through novel computational means.

For objective (a), we begin by noting that the multi-phase behavior of human locomotion can be represented as a multi-domain hybrid system (Sec. 2). In this fashion, we create a hybrid system for a bipedal robot in Sec. 3 that has phases of full, under and over actuation with transitions occurring due to heel-strike, toe-strike and heel lift. Further motivated by human walking, we then introduce human-inspired controllers for the continuous dynamics in each of these discrete domains (Sec. 4). To account for the hybrid nature of the system, a novel multi-domain optimization is proposed in Sec. 5 that ensures invariance of the zero dynamics surfaces that are a natural byproduct of human-inspired control. The end result is the generation controllers that yield multi-contact locomotion as verified in simulation.

To achieve objective (b), it is necessary to address the cyber-physical aspects of a robotic system, and specifically AMBER2, through algorithms that will allow for the implementation of the formal controllers. A novel method for translating gaits to physical hardware is presented in Sec. 6; importantly, these methods utilize the essential formal elements of the controllers and optimization which generated these gaits. This is made explicit through the essential code structures utilized for implementation (Sec. 7). The end result is the realization, on the bipedal robot AMBER2, of human-inspired multi-contact bipedal locomotion. These results are discussed in Sec. 8. Importantly, the resulting waling on AMBER2 consists of all of the key behaviors present in human walking: heel strike, toe strike and heel lift. Comparison between the experimental and simulation results shows good agreement between the experimental results and the theory from which they were derived.

2. MULTI-DOMAIN LOCOMOTION

This section reviews the multi-domain feature embedded in human locomotion. Based on the experimental data of human kinematics, three domains are extracted to characterize one step cycle of human walking.

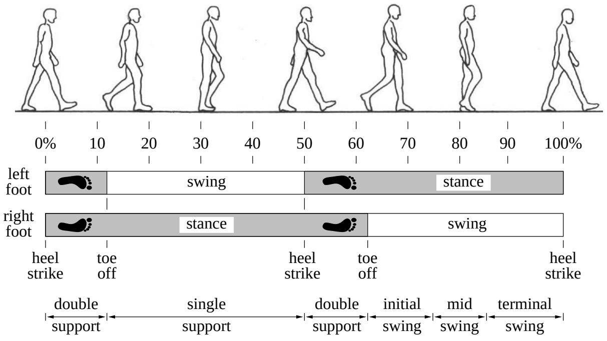


Figure 2: Diagram of a typical gait cycle.

Multi-Domain Structure of Human Walking. Understanding the walking pattern of a normal leg is of obvious importance when attempting to reproduce it robotically. Normally, a bipedal gait consists of two phases: stance phase, when the foot is on the ground, and swing phase, when the foot is in the air [4]. Specifically, sub-phases are usually disintegrated from each phase to describe human locomotion more explicitly. For example, at the beginning of the stance phase, for a portion of time both feet are in contact with the ground, which we define as double support domain. For the rest of the stance phase, only one foot is on ground, which we term as single support domain as shown in Fig. 2 (see [2]). Though different approaches break one step into different phases (for example, in impedance prosthetic control [6], the swing phase is divided into two sub-phases based on the knee angle), this work will focus only on the stance phase (which is the most critical factor of bipedal walking) by looking at the contact points between the feet and ground. The domain breakdown scheme of experimental human data is discussed in the following section.

Locomotion Domain Breakdown. With the goal of achieving human-like walking, we turn to the most prevalent source —the human body—for inspiration. The human locomotion data considered in this paper was obtained through a high speed motion capture system, the setup details of which can be found in [16]. Using the domain breakdown method discussed in [16], one step (only stance phase is considered) is divided into four sub-phases as shown in Fig. 3.

However, one can note that there is one sub-phase that only takes a small portion of one step (which is 3.1746% for the example subject considered in this work). Omission of this phase is possible without sacrificing the ability to capture the essentials of multi-domain human locomotion. Therefore, this work will focus on the other three domains of a single step as shown in Fig. 3. In particular, we define the domains explicitly as:

- *Over-actuated Domain*, *oa*: In this domain, only the stance heel and swing toe are on the ground;
- *Fully-actuated Domain*, fa: In this domain, both the toe and heel of stance foot are on the ground;

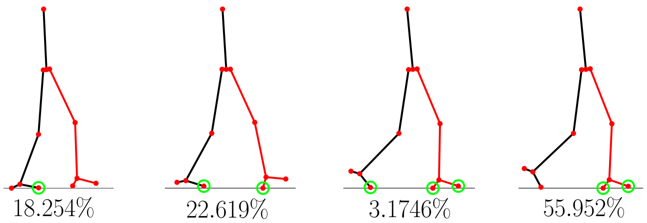


Figure 3: Domain breakdown of one step of one subject. The numbers below each tile indicate the percentage of time spent in that domain. The blue circle indicates that particular point of the feet (toe or heel) is in contact with ground. The red lines indicate the "non-stance" leg and the black lines represent the "stance" leg.

- *Under-actuated Domain*, *ua*: In this domain, only the stance toe is on the ground.

3. HYBRID MODEL OF AMBER2

This section presents the mathematical model of a footed bipedal robot with multi-domain locomotion. Considering the changes of foot contact points over a gait cycle (lifting and striking of the heel and toe), a hybrid system model is developed with both continuous dynamics and discrete dynamics. The physical planar robot AMBER2 is introduced at the end with details.

3.1 Hybrid System Model

Formally, the multi-domain locomotion bipedal system can be modeled as a hybrid control system [4], [12] as follows:

$$\mathscr{HC} = (\Gamma, D, U, S, \Delta, FG), \tag{1}$$

where

- $\Gamma = (V, E)$ is a *directed cycle*, with vertices $V = \{\text{oa}, \text{fa}, \text{ua}\}$; and edges $E = \{e_1 = \{\text{oa} \to \text{fa}\}, e_2 = \{\text{fa} \to \text{ua}\}, e_2 = \{\text{ua} \to \text{oa}\}\}$,
- $D = \{D_{\text{oa}}, D_{\text{fa}}, D_{\text{ua}}\}$ is a set of *domains of admissibility*,
- $U = \{U_{\text{oa}}, U_{\text{fa}}, U_{\text{ua}}\}$ is the set of *admissible controls*,
- $S = \{S_{\text{oa}\to\text{fa}}, S_{\text{fa}\to\text{ua}}, S_{\text{ua}\to\text{oa}}\}$ is a set of *guards*,
- $\Delta = \{\Delta_{\text{oa}\to\text{fa}}, \Delta_{\text{fa}\to\text{ua}}, \Delta_{\text{ua}\to\text{oa}}\}$ is a set of *reset maps*,
- $FG = \{(f_v, g_v)\}_{v\in V}$ with (f_v, g_v) a *control system* on D_v, i.e., $\dot{x} = f_v(x) + g_v(x)u_v$ for $x \in D_v$ and $u_v \in U$.

The remainder of this section will devote to developing the specific elements of this hybrid system in the context of the multi-domain walking gait of interest.

3.2 Robot Dynamics

Due to the changes of contact points on the foot throughout the course of the gait, generalized coordinates are naturally used to characterize the robot. Specifically, the configuration space $Q = \mathbb{R}^2 \times SO(2) \times Q_b$ is represented in coordinates as $\theta = \{p_x, p_y, \varphi_0, \theta_b\}$, where the extended coordinates

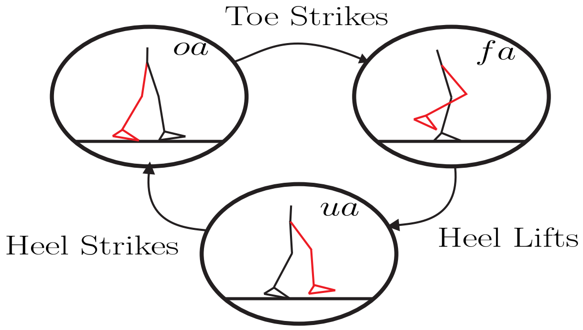


Figure 4: The directed graph of 3 domain walking.

$\{p_x, p_z, \varphi_0\}$ represent the position and rotation angle of the body fixed frame R_b with respect to a fixed inertial frame R_0; and $\theta_b = \{\theta_{sa}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa}\}$ denotes the body coordinates of the robot as shown in Fig. 5.

Continuous Dynamics. With the domain specification and the generalized coordinates $x = (\theta, \dot{\theta})$ in hand, we can now construct the control system FG for each domain D_v with $v \in V$ for hybrid system $\mathscr{HC}$.

The dynamics on each domain will be obtained from general "unpinned" dynamics through the use of holonomic constraints. Calculating the mass and inertia properties (as shown in Table 1) of each link through a CAD model allows for the construction of the Lagrangian:

$$L(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T D(\theta)\dot{\theta} - V(\theta). \tag{2}$$

Holonomic constraints are then added to enforce contact conditions (the detailed construction can be referred to [8]). The end result is a constrained dynamical system as follows,

$$M(\theta)\ddot{\theta} + H(\theta, \dot{\theta}) = B(\theta)u + J_v(\theta)^T F_v(\theta, \dot{\theta}), \tag{3}$$

where $M(\theta) \in \mathbb{R}^{9\times 9}$ is the inertial matrix, and $H(\theta, \dot{\theta}) \in \mathbb{R}^{9\times 1}$ contains the terms resulting from the Coriolis effect $C(\theta, \dot{\theta})\dot{\theta}$ and the gravity vector $G(\theta)$. $B(\theta)$ denotes the torque distribution matrix. $F_v(\theta, \dot{\theta})$, which are the reaction forces due to the holonomic constraints, are defined for each domain based on the contact conditions of the heel and toe. F_v can be explicitly derived from the states x and the controller u by differentiating the holonomic constraints twice. The details are omitted here and can be referred to [10].

Discrete Dynamics. With a given vertex $v \in V$, the continuous domain, D_v, describes the admissible configuration of the system restricted by the guard S_v. In particular, we consider two types of constraints: unilateral, denoted as h_v, and holonomic, denoted as η_v. The unilateral constraints determine the set of admissible configurations, i.e., the domains; while holonomic constraints are used to dictate specific contact points with the ground. With this setup, each domain

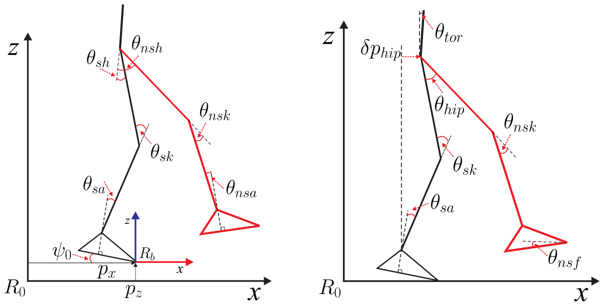


Figure 5: Robot configuration (left) and outputs (right).

can be defined as:

$$D_v = \left\{(\theta, \dot{\theta}) \in TQ : h_v(\theta) \geq 0,\ \eta_v = 0\right\}. \tag{4}$$

with the constraints given by:

- For $v = oa$, $h_{oa}(\theta)$ consists of the vertical height of stance toe, while $\eta_{oa}(\theta)$ consists the vertical height of stance heel and swing toe;
- For $v = fa$, $h_{fa}(\theta, \dot{\theta})$ is the vertical reaction force at the stance heel, while $\eta_{fa}(\theta)$ consists the vertical height of stance heel and toe;
- For $v = ua$, $h_{ua}(\theta)$ is the non-stance heel vertical height, while $\eta_{ua}(\theta)$ is the vertical height of stance toe.

Guards. The guards are defined when $h_v = 0$ with the additional assumption that h_v is decreasing; therefore, resulting the guard defined as:

$$S_e = \left\{(\theta, \dot{\theta}) \in TQ : h_v(\theta) = 0 \text{ and } \dot{h}_v(\theta) < 0\right\}, \tag{5}$$

Reset Maps. The evolution of domains combines with contact point leaves or impacts the ground. Reasonable reset that maps from pre-switch state x_e^- to post-switch x_e^+ state is required for stable walking.

When a specific contact point lifts from the ground, there are no velocity changes. Therefore, the reset map can be defined as identity. In particular, $\Delta_{e_2} = I_{18}$ falls into this case, i.e., $x_{e_2}^+ = x_{e_2}^-$. When a specific point impacts the ground, we assume no rebounding or slipping occurs[1], thus resulting the unchanged configuration ($\theta_e^+ = \theta_e^-$). However, the velocities must be updated due to the impact. Reset maps Δ_{e_1} and Δ_{e_3} fall into this category. The perfectly plastic impact model is considered to derive the reset map with impacts, which is given as the following,

$$x_e^+ = \Delta_e(\theta, \dot{\theta}) x_e^-, \quad with\ \Delta_e(\theta, \dot{\theta}) = \begin{bmatrix} \Delta_{\theta,e} \\ \Delta_{\theta,\dot{\theta},e} \end{bmatrix}, \tag{6}$$

where $e \in \{e_1, e_3\}$ [2] with $\Delta_{\theta,\dot{\theta},e}$ determining the velocity changes due to impact. The detailed discussion on its computation can be found in [8].

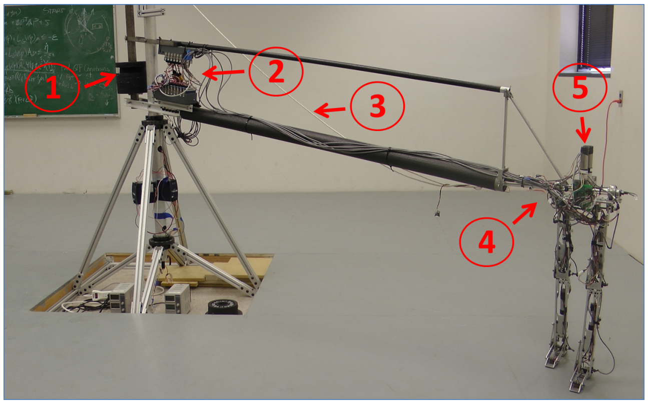


Figure 6: AMBER2 with the boom and electronics. The boom restricts motion to the sagittal plane. As shown in the figure: **(1)** Counterweight used to balance the boom around the pivot, **(2)** Controller module where the walking algorithm is running, **(3)** The boom, **(4)** Boom support structure which keeps the torso horizontal, **(5)** The bipedal robot AMBER2.

3.3 AMBER2 Model

AMBER2 is a 2D footed bipedal robot with seven links (two calves, two thighs, two feet and a torso, see Fig. 1). Six brushless DC (BLDC) motors actuate six joints. As a planar robot, the motion of AMBER2 has been restricted to the sagittal plane via a boom shown in Fig. 6, which are configured as parallel four-bar linkage such that no support in the sagittal plan is provided by the boom. The boom is fixed rigidly to a low friction rotating mechanism, allowing the biped to walk in a circular fashion. In addition, counterweights are provided to negate the weight of the boom on the robot.

The mathematical model of AMBER2 also contains the motors and the boom. The inertias of these two elements are included in the model differently. Details of this approach were explained in [15]. Since the end of the boom can move vertically and horizontally, it exhibits yaw and roll about the pivot. This would correspond to the x component and z component of the velocities of the torso. The CoM of the boom

Model Parameters				
Parameter	Mass	Length	Inertia x-axis	Inertia z-axis
	g	m	$\times 10^3\ g\,mm^2$	$\times 10^3\ g\,mm^2$
Stance foot	204.42	0.07445	139.698	406.384
Stance calf	1119.43	0.34313	9343.395	22211.105
Stance knee	1172.57	0.29845	9004.044	22404.696
Torso	2154.79	0.10401	20342.192	64678.601
Non-stance knee	1172.57	0.29845	9004.044	22404.696
Non-stance calf	1119.43	0.34313	9343.395	22211.105
Non-stance foot	204.42	0.07445	139.698	406.384

Table 1: The mass and length parameters of the robot.

[1] These are common assumptions in bipedal robot literature [14].

[2] Note that as a result of considering "stance" and "non-stance" legs, the labelling on the legs must be swapped at the end of a step (otherwise, $\Delta_{\theta,e} = I$); in this paper, this switching occurs at heel strike, i.e., $e_3 = \text{ua} \rightarrow \text{oa}$. This is common practice in robotic walking to reduce the number of discrete domains.

can be approximated to be at the center of the pivot considering the counterweight mass. With I_{boom} being the inertia of the boom, its mass matrix, $M_{boom} \in \mathbb{R}^{6\times 6}$, is:

$$M_{boom} = \begin{bmatrix} \frac{I_{boom}}{L_{boom}^2} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix},$$

where L_{boom} is the distance between CoM of the torso and the pivot.

The new combined mass inertia matrix, M_{com}, used in the lagrangian will be updated as:

$$M_{com}(\theta) = M(\theta) + diag(0, I_{m,sk}, I_{m,sh}, I_{m,nsh}, I_{m,nsk}, I_{m,nsa}) \\ + J(\theta)^T M_{boom} J(\theta), \quad (7)$$

where $I_{m,sk}, I_{m,sh}, I_{m,nsh}, I_{m,nsk}, I_{m,nsa}$ correspond to the motor inertia of respective links and $J(\theta)$ is the body Jacobin of the center of mass of the torso.

4. HUMAN-INSPIRED CONTROLLER

This section begins with the formal definition of human outputs combination. A simple function is then introduced to characterize these specifically selected outputs for tracking purposes. The human-inspired outputs of the robot is defined based on the human outputs and the representation function. The control laws that leverage the robot to display human-like multi-domain locomotion is constructed at the end.

Human Outputs Definition. We formally define the human outputs combination as follows:

DEFINITION 1. *A human output combination for $v \in V$ is a tuple $Y_v^H = (Q_b, y_{1,v}^H, y_{2,v}^H)$ consisting of a configuration space Q_b, a velocity-modulating output $y_{1,v}^H : Q_b \to \mathbb{R}$ and position-modulating outputs $y_{2,v}^H : Q_b \to \mathbb{R}^{n_v - 1}$ with n_v the degrees of freedom. Let O_v be an index set for $y_{2,v}^H$ whereby $y_{2,v}^H(\theta) = [y_{2,v}^H(\theta)_o]_{o\in O_v}$.*

A set of human outputs are *independent* if

$$rank(\begin{bmatrix} y_{1,v}^H(\theta) \\ y_{2,v}^H(\theta) \end{bmatrix}) = n_v, \quad (8)$$

on Q_b; and linear if

$$y_{1,v}^H(\theta) = c_v \theta, \quad (9)$$

$$y_{2,v}^H(\theta) = H_v \theta, \quad (10)$$

for $c_v \in \mathbb{R}^{1\times n_v}$ and $H_v \in \mathbb{R}^{(n_v - 1)\times n_v}$.

By investigating experimental human locomotion data, seven linear and independent output candidates are selected for the robot in this paper: $\delta p_{\text{hip}}(\theta)$, the linearized forward position of the hip measured from the stance ankle joint; θ_{sa}, the stance ankle angle; θ_{sk}, the stance knee angle; θ_{nsk}, the non-stance knee angle; θ_{hip}, the hip angle between two thighs; $\theta_{tor}(\theta)$, the torso angle from vertical, and $\theta_{nsf}(\theta)$, the angle of the non-stance foot w.r.t the horizontal, the details of which are denoted in Fig. 5.

To construct these human outputs in a formal way, the linearized hip velocity is considered as the velocity-modulating output for all three domains and is characterized by:

$$c_v = \begin{bmatrix} 0 & 0 & -Lc - Lt & -Lc - Lt & -Lt & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (11)$$

where $v \in V$, L_c and L_t are the length of calf and thigh, respectively. The remaining six position-modulating outputs can be written in the matrix form as:

$$H_{ua} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}. \quad (12)$$

The particular reason we write it as H_{ua} is that this output combination is also the outputs chosen for the under-actuated domain. The position-modulating outputs for the over- and fully- actuated domains are chosen to be sub-matrices of H_{ua} based upon the available degrees of actuation in these respective domains. In particular, $H_{oa} = (H_{ua})_{1,2,5,6}$ and $H_{fa} = (H_{ua})_{2-6}$, where here we use the notation $(H_{ua})_i$ to denote the i^{th} row of H_{ua}.

Human-Inspired Outputs. With the goal to replicate human locomotion on a robot, specific functions have to be defined to characterize those human outputs for tracking purposes. Analysis of the chosen outputs data shows that, the linearized hip position can be approximated by a linear function of time $\delta p_{\text{hip}}^d(t) = v_{hip} t$ for all three domains, and the position-modulating outputs can be characterized by a simple function, which we term the *extended canonical walking function* (ECWF):

$$y_{ecwf}(t) = e^{-\alpha_4 t}(\alpha_1 \cos(\alpha_2 t) + \alpha_3 \sin(\alpha_2 t)) + ... \\ \alpha_5 \cos(\alpha_6 t) + \kappa(\alpha)\sin(\alpha_6 t) + \alpha_7, \quad (13)$$

where $\kappa(\alpha) = (2\alpha_4\alpha_5\alpha_6/((\alpha_2)^2 + (\alpha_4)^2 + (\alpha_6)^2)$. It is found that this simple function can seemingly fit human locomotion data universally with high correlations. A detailed explanation can be found in [16].

Based on the linear fashion of the linearized hip position, we can parameterize the time as:

$$\tau(\theta) = (\delta p_{hip}(\theta) - \delta p_{hip}(\theta^+))/v_{\text{hip}}, \quad (14)$$

which removes the dependence of time in (13) and renders an autonomous ECWF that only depends on the states [14]. Specifically, θ^+ represents the robot configuration at the beginning of one step which can be defined arbitrarily. In particular, the over-actuated double support domain is chosen as the beginning domain of one step.

With the autonomous ECWF in hand, we formally define the human-inspired outputs as:

$$y_{\alpha_v,v}(\theta,\dot{\theta}) = \begin{bmatrix} y_{1,v}(\theta,\dot{\theta}) \\ y_{2,v}(\theta,\alpha_v) \end{bmatrix} = \begin{bmatrix} y_{1,v}^H(\theta,\dot{\theta}) - v_{hip} \\ y_{2,v}^H(\theta) - y_{2,v}^d(\tau(\theta),\alpha_v) \end{bmatrix}, \quad (15)$$

where $y_{1,v}(\theta,\dot{\theta})$ is the velocity-modulating human-inspired output that is the difference between the actual hip velocity $y^H_{1,v}(\theta,\dot{\theta})$ and the desired hip velocity v_{hip}. $y_{2,v}(\theta,\alpha)$ are the position-modulating human-inspired outputs, which are the difference between the actual outputs $y^H_{2,v}(\theta)$ and the desired outputs $y^d_{2,v}(\theta,\alpha)$. Particularly, we consider the under-actuated domain as an example; in this case, $y^H_{2,ua}$ and $y^d_{2,ua}$ can be configured as:

$$y^H_{2,ua} = H_{ua}\theta, \quad y^d_{2,ua}(\tau(\theta),\alpha_{ua}) = \begin{bmatrix} y_{ecwf}(\tau(\theta),\alpha_{sa}) \\ y_{ecwf}(\tau(\theta),\alpha_{sk}) \\ y_{ecwf}(\tau(\theta),\alpha_{nsk}) \\ y_{ecwf}(\tau(\theta),\alpha_{hip}) \\ y_{ecwf}(\tau(\theta),\alpha_{tor}) \\ y_{ecwf}(\tau(\theta),\alpha_{nsf}) \end{bmatrix}. \quad (16)$$

We will group all the output parameters consisting both the velocity-modulating output and the position-modulating outputs to form the parameter set $\alpha = \{v_{hip},\alpha_{sa},\alpha_{sk},\alpha_{nsk},\alpha_{hip},\alpha_{tor},\alpha_{nsf}\} \in \mathbb{R}^{43}$. In particular, by defining $\alpha_{v_{hip}} = \{v_{hip},0,0,0,0,0,0\} \in \mathbb{R}^7$, the vector components of α can be stacked to a matrix form. Specifically, we have $\alpha \in \mathbb{R}^{7\times 7}$. Note that, only one ECWF has been used to characterize a specific output during a whole step including all three domains. Due to the fact that the degrees of freedom of each domain is different, the parameter set matrix for specific domain will be the sub-row matrices of α. In particular, $\alpha_{oa} = \alpha([1-3,6,7],:)$, $\alpha_{fa} = \alpha([1,3-7],:)$ and $\alpha_{ua} = \alpha([2-7],:)$.

Control Law Construction. The goal of the controller is to drive the outputs of robot to the outputs of human as represented by the ECWF in each domain. Considering the fact that bipedal robots display highly nonlinear dynamics, we naturally choose the Input/Output Linearization controller to drive $y_{\alpha_v,v} \to 0$ in an exponential convergence fashion. In particular, in the domain of full- and over- actuation, we define the *human-inspired controller* as:

$$u^{\alpha_v,\varepsilon}_v(\theta,\dot{\theta}) = -A^{-1}_v(\theta,\dot{\theta})\left(\begin{bmatrix} 0 \\ L^2_{f_v}y_{2,v}(\theta,\dot{\theta}) \end{bmatrix} + \begin{bmatrix} L_{f_v}y_{1,v}(\theta,\dot{\theta}) \\ 2\varepsilon L_{f_v}y_{2,v}(\theta,\dot{\theta}) \end{bmatrix} + \begin{bmatrix} \varepsilon y_{1,v}(\theta,\dot{\theta}) \\ \varepsilon^2 y_{2,v}(\theta,\dot{\theta}) \end{bmatrix}\right), \quad (17)$$

with $v \in \{\mathrm{fa},\mathrm{oa}\}$ and L the Lie derivative. Note that, $\varepsilon > 0$ is a user defined control gain that determines the convergence rate of $y_{\alpha_v,v} \to 0$. The decoupling matrix $A_v(\theta,\dot{\theta})$ is given:

$$A_v(\theta,\dot{\theta}) = \begin{bmatrix} L_{g_v}y_{1,v}(\theta,\dot{\theta}) \\ L_{g_v}L_{f_v}y_{2,v}(\theta,\dot{\theta}) \end{bmatrix}, \quad (18)$$

which is nonsingular because of the specific selection of outputs combination. For the under-actuated domain ua, the controller is defined as,

$$u^{\alpha_{ua},\varepsilon}_{\mathrm{ua}}(\theta,\dot{\theta}) = -A^{-1}_{\mathrm{ua}}(\theta,\dot{\theta})\left(L^2_{f_{\mathrm{ua}}}y_{2,\mathrm{ua}}(\theta,\dot{\theta}) + 2\varepsilon L_{f_{\mathrm{ua}}}y_{2,\mathrm{ua}}(\theta,\dot{\theta}) + \varepsilon^2 y_{2,\mathrm{ua}}(\theta,\dot{\theta})\right), \quad (19)$$

with $A_{ua} = L_{g_{\mathrm{ua}}}L_{f_{\mathrm{ua}}}y_{2,\mathrm{ua}}(\theta,\dot{\theta})$.

5. MULTI-DOMAIN OPTIMIZATION

This section will focus on developing the human-like multi-domain optimization that yields the controller parameters α that will result in human-inspired multi-domain walking on AMBER2. Both mathematical constraints (for stable walking) and physical constraints (for implementation on a physical robot) are discussed in detail. Simulation results using the obtained parameters with human-inspired control are presented at the end to show the stability of the walking.

Full and Partial Zero Dynamics. The human-inspired controller can drive the human-inspired outputs $y_{\alpha_v,v}(\theta,\dot{\theta}) \to 0$ exponentially, which renders the *full zero dynamics surface* for continuous dynamics:

$$\mathbf{FZ}_{\alpha_v} = \{(\theta,\dot{\theta}) \in TQ : y_{\alpha_v,v}(\theta,\dot{\theta}) = \mathbf{0},\ L_{f_v}y_{2,v}(\theta,\alpha_v) = \mathbf{0}\}. \quad (20)$$

The invariance of this surface does not extend to the case of discrete dynamics with impacts, i.e., it does not necessarily extend to the case of hybrid systems. With a view towards the importance of the position-modulating outputs $y_{2,v}(\theta,\alpha_v)$, we define the surface for which these outputs agree for all time as *partial zero dynamics surface*[3]:

$$\mathbf{PZ}_{\alpha_v} = \{(\theta,\dot{\theta}) \in TQ : y_{2,v}(\theta,\alpha_v) = \mathbf{0},\ L_{f_v}y_{2,v}(\theta,\alpha_v) = \mathbf{0}\}. \quad (21)$$

With the relaxation of the velocity-modulating output, the $\mathbf{PZ}_{\alpha_v}$ surface can be specifically designed such that it is invariant even for a hybrid system with impact. That is to say, the goal of partial *hybrid* zero dynamics (PHZD) is to find parameters α ensuring that this surface remains invariant through the major impact of the multi-domain walking as: $\Delta_{fa\to oa}(S_{e_2} \cap \mathbf{FZ}_{\alpha_{fa}}) \subset \mathbf{PZ}_{\alpha_{oa}}$. Note that, the impact equation $\Delta_{fa\to oa}$ includes both the impact due to the heel and toe, along with numerical integration through the under actuated domain. Therefore, all of the impacts of the system can be represented through the single impact equation $\Delta_{\mathrm{fa}\to oa}$.

PHZD Reconstruction. The desired joint angles and angular velocities of the robot are found through inverse projection from the PHZD surface. With the assumption that the system evolves on the PHZD surface, a low dimensional representation of the system can be obtained by defining the zero dynamics coordinates of domain $v \in \{oa, fa\}$:

$$\xi_{1,v} = \delta p^H_{hip}(\theta) := c_v\theta,$$
$$\xi_{2,v} = y^H_{1,v}(\theta,\dot{\theta}) := \delta\dot{p}^H_{hip}(\theta) := c_v\dot{\theta}. \quad (22)$$

Since ξ_1 is the linearized position of the hip, which is used to parameterize time as (14), we can write the desired outputs $y^H_{2,v}(\theta) = y^d_{2,v}(\xi_{1,v},\alpha_v)$. Therefore, we have the following relationship between the desired joint state and the desired

[3] Note that, domain $v = ua$ only has position-modulating outputs, therefore, the $\mathbf{PZ}_{\alpha_v}$ surface for under-actuated domain is actually *full zero dynamic surface*, i.e., $\mathbf{FZ}_{\alpha_{ua}} := \mathbf{PZ}_{\alpha_{ua}}$. More importantly, the $\mathbf{PZ}_{\alpha_v}$ will converge to the $\mathbf{FZ}_{\alpha_v}$ if the system is fully-actuated or over-actuated.

outputs of the robot as:

$$\theta_v^d = \Psi(\xi_{1,v}, \alpha_v) = \begin{bmatrix} c_v \\ H_v \end{bmatrix}^{-1} \begin{pmatrix} \xi_{1,v} \\ y_{2,v}^d(\xi_{1,v}, \alpha_v) \end{pmatrix},$$

$$\dot{\theta}_v^d = \Phi(\xi_{1,v}, \xi_{2,v}, \alpha_v) = \begin{bmatrix} c_v \\ H_v \end{bmatrix}^{-1} \begin{pmatrix} v_{hip} \\ \frac{\partial y_{2,v}^d(\xi_{1,v}, \alpha_v)}{\partial \xi_{1,v}} \xi_{2,v} \end{pmatrix}. \quad (23)$$

Multi-Domain Optimization. With the goal of finding the controller parameters, α, which deliver stable multi-domain robotic walking, a human-inspired optimization problem subject to PHZD is given by the following:

$$\alpha^* = \underset{\alpha \in \mathbb{R}^{43}}{\text{argmin}} \ \text{Cost}_{\text{HD}}(\alpha) \quad \text{(HIO)}$$

$$\text{s.t} \quad \Delta_{fa \to oa}(S_{e_2} \cap \mathbf{PZ}_{\alpha_{fa}}) \subset \mathbf{PZ}_{\alpha_{oa}} \quad \text{(PHZD)}$$

where,

$$\Delta_{fa \to oa}(\theta, \dot{\theta}) = \Delta_{e_3}(\varphi^{ua}_{T_{ua}(\theta,\dot{\theta})}(\theta, \dot{\theta})) \quad (24)$$

with φ^{ua} the solution to the vector field (f_{ua}, g_{ua}) with initial condition $(\theta, \dot{\theta}) \in S_{e_2} \cap \mathbf{FZ}_{\alpha_{fa}}$; and $T_{ua}(\theta, \dot{\theta})$ is the *time to impact* function that determines the first time when the solution intersects the guard (see [9]). Note that, the cost function (HIO) is the least squares fit between the human experimental data and the ECWF representations [16]. In order to solve this nonlinear optimization, the following sections will be devoted to reformulating the constraints in an explicit manner such that the optimization can be numerically solved.

Fully-actuated to Under-actuated Constraints. In order to reframe PHZD in a way that can be numerically approached, we use the PHZD reconstruction strategy to construct a point $(\upsilon, \dot{\upsilon}) \in \mathbf{FZ}_{fa} \cap S_{e_3}$, where, by definition, we know that $\xi_{2,fa} = v_{hip}$. In order to get the hip position $\xi_{1,fa}$, we add an additional parameter by defining $\xi_{1,fa} = \alpha^{fa}_{p_{hip}}$. Therefore, we expand our set of parameters with defining: $\beta_{fa} = \{\alpha^{fa}_{p_{hip}}, \alpha_{fa}\}$. By doing so, we can explicitly solve the point $(\upsilon(\beta_{fa}), \dot{\upsilon}(\beta_{fa}))$ as $\upsilon(\beta_{fa}) = \Psi(\alpha^{fa}_{p_{hip}}, \alpha_{fa})$ and $\dot{\upsilon}(\beta_{fa}) = \Phi(\alpha^{fa}_{p_{hip}}, v_{hip}, \alpha_{fa}))$.

With this construction, we can specifically impose the constraint of domain fa that the reaction force on the heel has to cross zero,

$$h_{\text{fa}}(\upsilon(\beta_{fa}), \dot{\upsilon}(\beta_{fa}) = 0 \quad \text{(PC1)}$$

Note that, the fully-actuated domain fa will switch to the under-actuated domain ua smoothly without requirement of any further constraints except the guard condition. This is benefit of the unique ECWF used through all three domains. Particularly, by adding the addition parameter $\alpha^{fa}_{p_{hip}}$, the time of the switch S_{e_2} can be optimized.

Under-actuated to Over-actuated Constraints. With the point $(\upsilon(\beta_{fa}), \dot{\upsilon}(\beta_{fa})) \in \mathbf{FZ}_{fa} \cap S_{e_2}$ constructed above, we know this point is also the initial point of domain ua due to the fact $\Delta_{\theta,e_2} = I$. With φ^{ua} denoting the solution of to the vector (f_{ua}, g_{ua}), we can define the following point:

$$(\varphi(\beta_{fa}), \dot{\varphi}(\beta_{fa})) = \varphi^{ua}_{T_{ua}(\upsilon(\beta_{fa}), \dot{\upsilon}(\beta_{fa}))}(\upsilon(\beta_{fa}), \dot{\upsilon}(\beta_{fa})). \quad (25)$$

Clearly, $(\varphi(\beta_{fa}), \dot{\varphi}(\beta_{fa})) \in S_{e_3}$. In order to satisfy the PHZD constraints, the post impact state of $(\varphi(\beta_{fa}), \dot{\varphi}(\beta_{fa}))$ has to be on the surface of $\mathbf{PZ}_{oa}$, which implies the following constraints:

$$y_{2,oa}(\Delta_{\theta,e_3}\varphi(\beta_{fa})) = 0 \quad \text{(PC2)}$$

$$dy_{2,oa}(\Delta_{\theta,e_3}\varphi(\beta_{fa}))\Delta_{\theta,\dot{\theta},e_3}\dot{\varphi}(\beta_{fa}) = 0 \quad \text{(PC3)}$$

$$\frac{\partial h_{ua}(\varphi(\beta_{fa}))}{\partial \varphi(\beta_{fa})}\dot{\varphi}(\beta_{fa}) < 0 \quad \text{(PC4)}$$

where constraint (PC4) implies that the impact is transverse to the guard [3].

Over-actuated to Fully-actuated Constraints. Analogous to the PHZD reconstruction at the end of domain fa, we seek to construct a point $(\upsilon, \dot{\upsilon}) \in \mathbf{FZ}_{oa} \cap S_{e_1}$ with an additional parameter $\alpha^{oa}_{p_{hip}}$ denoting the hip position at the end of domain oa. Note that, with the assumption that the controller gain ε is large enough to drive the dynamics to $\mathbf{FZ}_{oa}$ with sufficient speed (before the end of domain oa), we have $\xi_{2,oa} = v_{hip}$. Therefore, by defining the extended parameter set to be $\beta_{oa} = \{\alpha^{oa}_{p_{hip}}, \alpha_{oa}\}$, we can solve for this point as $\upsilon(\beta_{oa}) = \Psi(\alpha^{oa}_{p_{hip}}, \alpha_{oa})$ and $\dot{\upsilon}(\beta_{oa}) = \Phi(\alpha^{oa}_{p_{hip}}, v_{hip}, \alpha_{oa}))$.

Finally, we can explicitly compute the point at the beginning of the domain fa using the reset map Δ_{e_1} with $\Delta_{\theta,e_1} = I$ and $\Delta_{\theta,\dot{\theta},e_1}$ as discussed in (6). Thence, the PHZD condition implies the following constraints:

$$y_{\{hip,tor\},fa}(\upsilon(\beta_{fa})) = 0 \quad \text{(PC5)}$$

$$|dy_{2,fa}(\upsilon(\beta_{oa}))\Delta_{\theta,\dot{\theta},e_1}\dot{\upsilon}(\beta_{oa})| < \sigma \quad \text{(PC6)}$$

$$\frac{\partial h_{oa}(\upsilon(\beta_{oa}))}{\partial \upsilon(\beta_{oa})}\dot{\upsilon}(\beta_{oa}) < 0 \quad \text{(PC7)}$$

where (PC7) implies that the impact is transverse to the guard and σ is a small positive user-defined value (which is chosen to be 0.1 in our application). Note that, since only one ECWF has been utilized to characterize the outputs of a whole gait cycle, the PHZD surface can not be fully guaranteed throughout the whole step which contains three domains and two impacts. Therefore, the PHZD constraints for the switch between over-actuated domain oa and fully-actuated domain fa have to be relaxed by only constraining the positions of the outputs. In particular, the shared position modulating outputs between domain oa and fa will be continuous by construction due to the identity map. Constraints (PC5) enforce that the outputs $y_{hip,\text{fa}}$ and $y_{tor,\text{fa}}$ that are not tracked during the domain oa should be on the surface of $\mathbf{PZ}_{\text{fa}}$. Constraints (PC6) make sure that the velocity changes due to the minor impact of the toe strike are sufficiently smaller than a specific value. As a result, the system will not be off the PHZD surface too much and will converge back to the surface quickly.

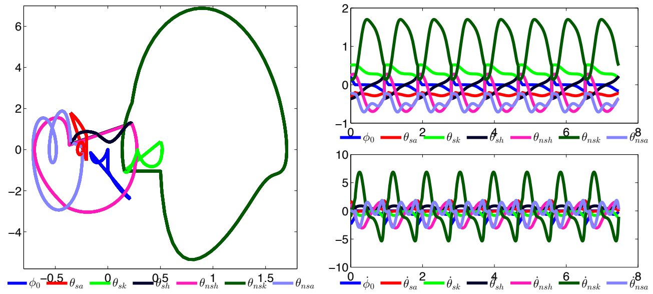


Figure 7: On the left is the phase plot for 8 steps of the designed 3 domain walking gait. The joint angles and angular velocities are shown in the plots in the top and bottom right.

Physical Constraints. Note that, despite the PHZD constraints which render stable periodic orbits in hybrid systems [3], we also need to consider several physical constraints such that the results of the optimization are in a form that can be implemented on physical robots. In particular, the following two physical constraints are considered:

Torque Constraints. Torques acting on the joints are limited by the capacity of the motors and the modules. Therefore, the optimized gait has to respect the hardware torque bounds, which is stated as:

$$\max_{0\le\tau(\theta(\alpha))} ||u(\theta(\alpha),\dot{\theta}(\alpha),\varepsilon)|| \le MAX_{torque}. \tag{PYSC1}$$

Foot Scuffing Conditions. The swing height clearance of toe and heel, and stride length during the swing phase must be sufficient to avoid scuffing amidst sensor noise, tracking error, uneven ground and even imperfections in the mechanical design. Therefore, foot scuffing conditions must be imposed to insure sustainable walking. Explicitly, we define:

$$\begin{aligned}
&\max_{0\le\tau(\theta(\alpha))} (h_{nst}(\theta(\alpha)) - h_{quad}(\theta(\alpha),hmax)) > 0,\\
&\max_{0\le\tau(\theta(\alpha))} (h_{nsh}(\theta(\alpha)) - h_{quad}(\theta(\alpha),hmax)) > 0,\\
&\max_{0\le\tau(\theta(\alpha))} l_{nsf}(\theta(\alpha)) - MIN_{steplength} > 0,
\end{aligned} \tag{PYSC2-4}$$

where h_{quad} is a quadratic polynomial above which the height of non-stance toe (h_{nst}) and heel (h_{nsh}) must remain during the course of a step. The stride length l_{nsf} is constrained to be greater than a minimum specified stride length, $MIN_{steplength}$.

Main Results. Utilizing all of the formal constructions above, together with the constraints needed for practical implementation, the final optimization problem can be stated as:

$$\begin{aligned}
\beta^* = \operatorname*{argmin}_{\beta\in\mathbb{R}^{45}} \; & \mathrm{Cost}_{\mathrm{HD}}(\beta) && \text{(HIO)}\\
\text{s.t} \quad & PHZD \;\; Constraints && \text{(PC1-7)}\\
& Physical \;\; Constraints && \text{(PYSC1-4)}
\end{aligned}$$

where $\beta = \{\alpha^{oa}_{p_{hip}}, \alpha^{fa}_{p_{hip}}, \alpha\}$ is the final expanded parameter set. By solving this optimization problem using the MATLAB built-in function $fmincon$, we can obtain β^* parameters that best fit human-walking data while enforcing the desired constraints to achieve multi-domain robotic walking. The end result of this optimization is that by only using one set of parameters β^*, stable multi-domain robotic walking (with the human-inspired controller) of AMBER2 has been achieved in simulation. The phase portrait and the joint trajectories for multi-steps can be seen in Fig 7. The Poincarѐ map has been utilized to numerically prove the stability of this gait with the maximum eigenvalue smaller than a unit ($-0.3422e^{-8}$). More importantly, the optimized parameters β^* will be shown to give human-like multi-domain walking on the physical robot of AMBER2.

6. TRAJECTORY RECONSTRUCTION

With the human-inspired outputs obtained from the optimization problem, this section will introduce the PHZD reconstruction methodology for the reconstruction of the desired state based joint trajectory for the physical robot.

Zero Dynamics. With the zero dynamics coordinates defined in Sec. 5, we can explicitly construct the ODE of the zero dynamics. Particularly, we have the fact that the velocity-modulating output evolves according to $\dot{y}_{1,v} = -\varepsilon y_{1,v}$ with $v \in \{oa, fa\}$. Therefore, with the definition of PHZD, the zero dynamics evolve according to the linear ODE:

$$\begin{aligned}
\dot{\xi}_{1,v} &= \xi_{2,v},\\
\dot{\xi}_{2,v} &= -\varepsilon(\xi_{2,v} - v_{hip}).
\end{aligned} \tag{26}$$

Having known $\xi_{1,v}$, $\xi_{2,v}$, the desired angles and velocities are obtained from (23). In other words, since $(\theta_d, \dot{\theta}_d)$ are derived from the outputs v_{hip} and $y^d_{2,v}(\tau(\theta(\alpha_v)), \alpha_v)$, tracking these joint angles and velocities is equivalent with tracking the robot's outputs. Therefore, the restriction of the dynamics to the PHZD surface is maintained.

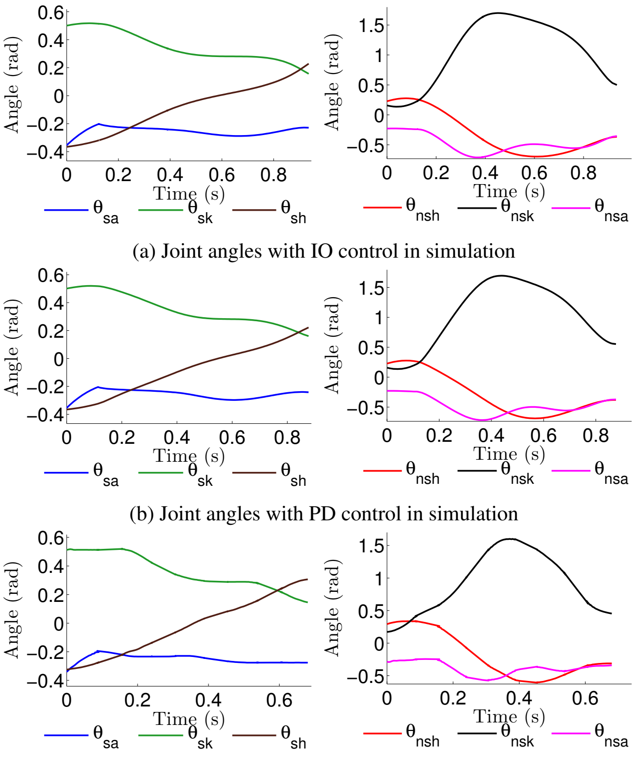
Trajectory Reconstruction of AMBER2. As discussed above, in order to obtain the desired trajectories $(\theta_d, \dot{\theta}_d)$ for AMBER2, both $\xi_{1,v}$ and $\xi_{2,v}$ for each domain must be computed based on the current state. However, as the velocity term $\xi_{2,v}$ is associated with multiple encoders, the actual $\xi_{2,v}$ will accumulate the signal errors of all its contributing encoders. The end result will be inaccurate velocity data. To bypass this shortcoming, we solve the ODE shown in (26) explicitly as the following:

$$\xi_1(t) = v^*_{hip}t + \frac{(1-\exp(-\varepsilon t))}{\varepsilon}(v^0_{hip} - v^*_{hip}) + \delta p^0_{hip}, \tag{27}$$

$$\xi_2(t) = v^*_{hip} + \exp(-\varepsilon t)(v^0_{hip} - v^*_{hip}), \tag{28}$$

where δp^0_{hip} and v^0_{hip} are the initial hip position and hip velocity at the beginning of the step; v^*_{hip} is the optimized desired hip velocity.

Instead of using time t, we replace it with the parameterized time $\tau(\theta)$ to achieve state based tracking. The more de-



(a) Joint angles with IO control in simulation

(b) Joint angles with PD control in simulation

(c) Joint angles with PD control in experiment

Figure 8: Comparison of actual joint angles between simulation results and experimental results logged during AMBER2 walking.

tailed implementation on the physical robot will be discussed later with pseudo code. Note that, even though the velocity modulating output is not tracked in the under-actuated domain, the use of this method is reasonable to achieve an approximation of the desired trajectory considering the short duration of the domain [4].

For the other two domains oa and fa, the desired trajectory are computed with minor modifications.

Over-actuated Domain. In the over-actuated domain, AMBER2 has 5 DOF but is actuated with 6 independent motors. In simulation, the redundant actuation is constrained by the holonomic constraints. This technique, however, is not applicable for a physical robot using pure PD control. Therefore, instead of using all of the angles computed from (23) directly, the geometric constraint is applied to update the redundant joint angle. By tracking the updated desired trajectory, both the stance heel and swing toe will remain on the ground, therefore satisfying the holonomic constraints.

Fully-acutated Domain. In the fully-actuated domain, all the extended coordinates should be $\mathbf{0}$ with the stance foot being flat on the ground throughout the domain. More importantly, the output θ_{sa} is not tracked. Therefore, the linear form of the outputs can be reformulated as:

$$c_{fa} = \begin{bmatrix} -Lc-Lt & -Lt & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$H_{fa} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}.$$

Specifically, with this modification, the ψ_0 term has been removed from the output matrix and only the body coordinates θ_b are considered. Importantly, removal of the output θ_{sa} allows us to eliminate the control's redundancy.

With the desired joint trajectory reconstructed above, the following section will focus on the controller implementation on the physical robot AMBER2.

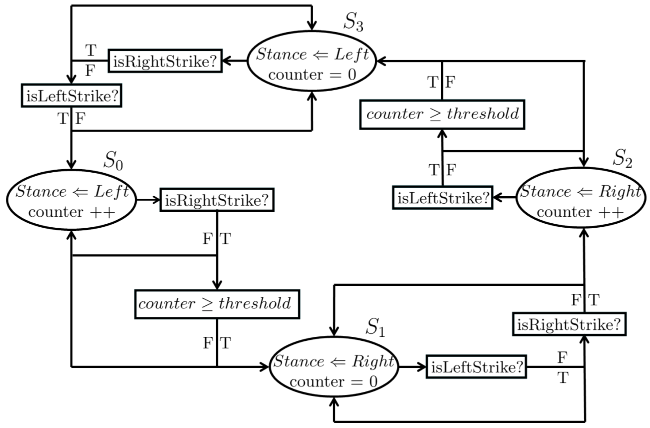


Figure 9: State machine showing the foot contact and the logic used to determine the stance leg.

7. EXPERIMENTAL REALIZATION

AMBER2's control infrastructure consists of two levels: a high level controller which is realized by Real-Time (RT) control, and a low level controller realized by Field-Progra--mmable Gate Array (FPGA). The objective of this section is to introduce the control scheme of AMBER2 as it is realized experimentally and in simulation.

High Level Controller. The Real Time control is in charge of the following major functionalities incorporated as:

1. Interface with FPGA, including read joint angles and angular velocities, send torque command to low level controller, enable/disable motors.
2. Compute the time parameter τ, determine current system domain.
3. PHZD reconstructon and geometric reconstructon to find desired joint angles.
4. Compute torque command by applying PD control law to corresponding motors, which is fed into the FPGA.

[4] The reason is that the ankle joint motor can not provide enough torque (hardware limitations) in the under-actuated domain to rotate the weight of the robot around the pivot point at the toe.

Algorithm 1 Real Time Module

```
Input: AMBER2 Parameters: Calf Length($L_c$), Thigh Length($L_t$);
Input: Optimizated trajectory parameters: $\delta p^R_{\text{hip}}(\theta^+), v_{\text{hip}}, \alpha$;
Input: PD Controller Gain: $K_p$, $K_d$
Input: $\theta_{La}, \theta_{Lk}, \theta_{Lh}, \theta_{Rh}, \theta_{Rk}, \theta_{Ra}, \dot{\theta}_{La}, \dot{\theta}_{Lk}, \dot{\theta}_{Lh}, \dot{\theta}_{Rh}, \dot{\theta}_{Rk}, \dot{\theta}_{Ra}$ ;
Input: Feet States; Encoder Status; Drive Status;
Output: Enable/Disable Motor Drives;
Output: Desired Torque for FOC;
Enable Motor Drives;
repeat
    Wait till all motor drives are Enabled
until ( Drive-Status == Enable )
while ( ¬ Stop-RT ) do
    Reform $\theta, \dot{\theta}$ from Left/Right($\theta_{LR}$) to Stance/nonStance($\theta_{SnS}$);
    Calculate time threshold for different domians $\tau_1$, $\tau_2$, $\tau_3$;
    Calculate actual time parameter $\tau_a$ ;
    if $\textit{real time} \leq \tau_1$ then
        $\textit{domain} \Leftarrow 1$;
        Desired $\tau_d = \textit{real time}$;
    else
        if $\tau_a > \tau_2$ then
            $\textit{domain} \Leftarrow 2$;
        else
            $\textit{domain} \Leftarrow 3$;
        end if
        Desired $\tau_d = \tau_a + \delta T$;
    end if
    Calculate( $\xi_1$, $\xi_2$);
    Calculate( $y_d$, $\dot{y}_d$ ) based on $\tau_d$;
    if $\textit{domain} == 1$ then
        Calculate( $\theta_d$, $\dot{\theta}_d$ ) via PHZD reconstruction and Geometric Reconstruction;
    else if $\textit{domain} == 2$ then
        Calculate( $\theta_d$, $\dot{\theta}_d$ ) via PHZD reconstruction;
    else
        Calculate( $\theta_d$, $\dot{\theta}_d$ ) via PHZD reconstruction;
    end if
    Apply PD Control law:
        $\tau^f_{PD} = K_p(\theta_a - \theta_d) + K_d(\dot{\theta}_a - \dot{\theta}_d)$;
    Reform $\tau^f_{PD}$ from Stance/nonStance to Left/Right;
    Sending Torque Command to FPGA;
    Log Data into Remote Desktop;
end while
Disable Motor Drives;
Report Errors and Stop the Real Time VI;
```

Algorithm 2 FPGA Module

```
Input: PWM Pulses from Absolute Encoders ;
Input: Hall Sensor Signal, Incremental Encoder Signal;
Input: Status of Foot Contact Switches;
Input: Auto-phasing results: Hall Angle, Index Angle;
Input: Hardware Setup: Sample Rate, Current Limitation, FOC Gains;
Input: Enable/Disable Motor Drives;
Input: Three Phase Current From BLDC motors;
Input: Torque Command from RT;
Output: Three Phase PWM Signals to Motor Drives;
Output: $\theta_{abs}, \dot{\theta}_{incremental}$;
Output: L/R Stance Foot; Encoder Status; Drive Status;
loop
    Absolute Encoder Reading logic(10MHz);    // Refer to data sheet of absolute encoder, US digital MAE3 kit
    if ( Signal low for 2 periods of encoder pulse) then
        Encoder Not Working ← 1;
    else
        Encoder Not Working ← 0;
    end if
    Incremental Quadrature Encoder Reading Logic(40MHz);
end loop
loop
    Compute Desired Current from Torque Command from RT;
    if (Joint Angle exceeds Workspace and Torque Command not trying to stop it) then
        Reset Desired Current to 0;
    end if
    Compute Three Phase Voltage through Field-oriented Control Logic; (shown in Fig. 10)    // Operation Frequency: 40MHz
    PWM signal Generation logic;
end loop
loop
    Guard and Stance Leg Detection Logic using foot contact switches (shown in Fig. 9);
    if ( Left Leg stance ) then
        L/R stance ← 0;
    else if ( Right Leg stance ) then
        L/R stance ← 1;
    end if
end loop
```

Note that for AMBER2, the sample rate and command rate are both 200Hz. The high level controller is coded into shared libraries to interface in C++ to improve the efficiency of execution. The NI9144 EhterCAT Slave chassis is connected to the cRIO to increase the capacity. For this configuration, each chassis is in charge of one leg. The pseudo-code running in RT is shown in Algorithm 1.

Low Level Controller. The low level controller is coded to the FPGA with on board clock 40MHz, which serves the following major functionalities:

1. Measure angular velocity by the single-ended incremental quadrature encoders attached to every rotor. Measure joint angle by integrating velocity data. In particular, the incremental encoders operate at 40MHz.
2. Detect stance foot by the heel and toe contact switches. Foot logic is shown in the state machine Fig. 9.
3. Torque control. Field-oriented control (FOC) is employed at the motor level. As shown in the control block diagram in Fig. 10, the torque is translated to current command first. Then the flux angle is computed from the hall sensor and incremental encoder data, which are initialized by auto-phasing. Finally, the motors are actuated by applying a PI controller on the quadrature and direct current.

Pseudo-code running in FPGA is shown in Algorithm 2.

8. RESULTS AND CONCLUSIONS

Before implementing the controller on the physical robot, the proposed controller was first verified in simulation. Comparing with the simulated results generated using the human-inspired controller as seen in Fig. 8a, we can see that the PD controller with the reconstruction strategy has achieved similar performance as seen in Fig. 8b.

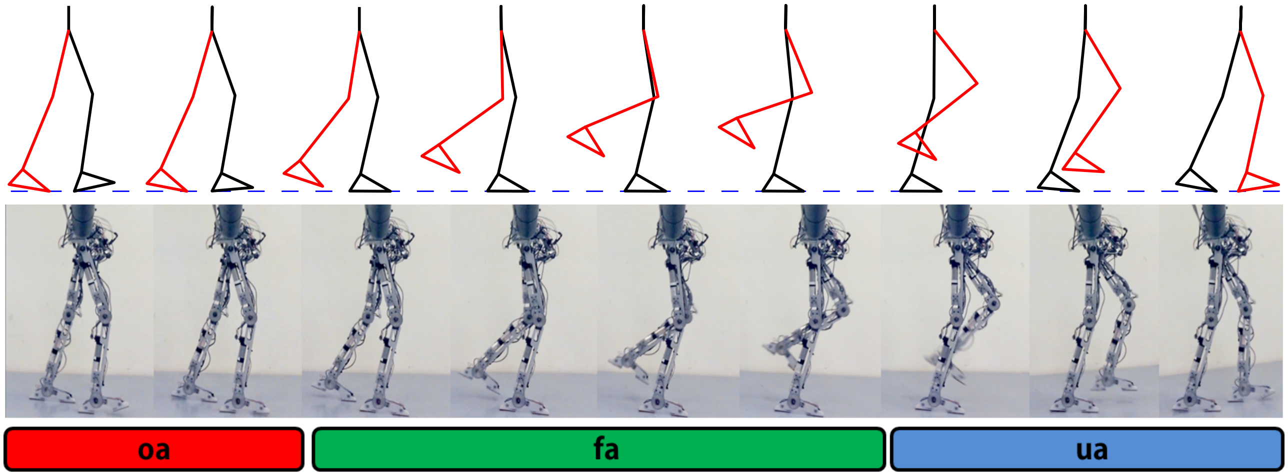


Figure 13: Comparison of walking tiles of simulated and experimental walking with PD control.

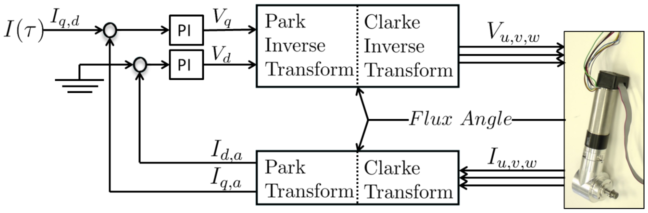
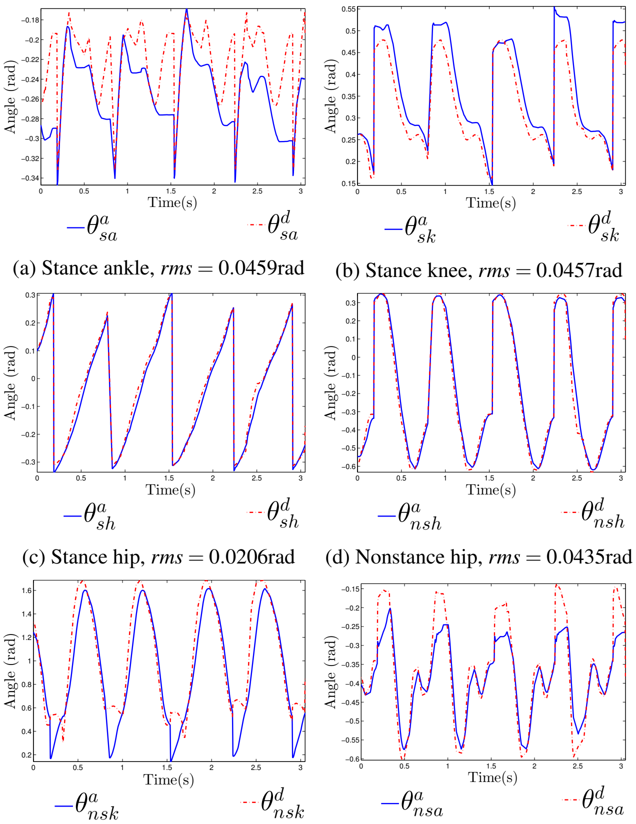


Figure 10: Field-oriented control block diagram

Experimental Results. By applying a PD controller to track the reconstructed joint trajectories, AMBER2 has achieved sustainable human-like multi-domain walking. From the attached video [1], the multi-domain walking of AMBER2 displays all the key features of human-like locomotion: toe strike, heel lift and heel strike. Particularly, AMBER2 has continuously walked for 45min with an approximated 1100m traveling distance. The test ended due to mechanical failure of a chain. The comparison between experimental gait tiles and simulated gait tiles is shown in Fig. 13, and the actual joint angles of one step are shown in Fig. 8c to compare with the simulated results. These two comparisons show good agreement between theory and practice. Also of note is that the system is developed with minimum sensing requirements by only using foot contact switches and incremental encoders. The inherent advantages (simpler form and better behavior outside of the nominal operation window, see [16] for more details) imbibed in the ECWF, as well as the robot's design methodology, facilitated the ease of applying such simple control laws to realize walking, which also result in low torque consumption throughout the step. The actual joint angles of multiple steps along with the reconstructed desired joint trajectories are shown in Fig. 11. The corresponding input torques are shown in Fig. 12.

Discussion. Readers may notice that the walking speed of the physical robot (approximately 0.4m/s) is faster than the speed of simulated walking (0.28m/s). It is also noted that the experimental under-actuated domain is shorter than that in simulation. These phenomena are caused by the ankle

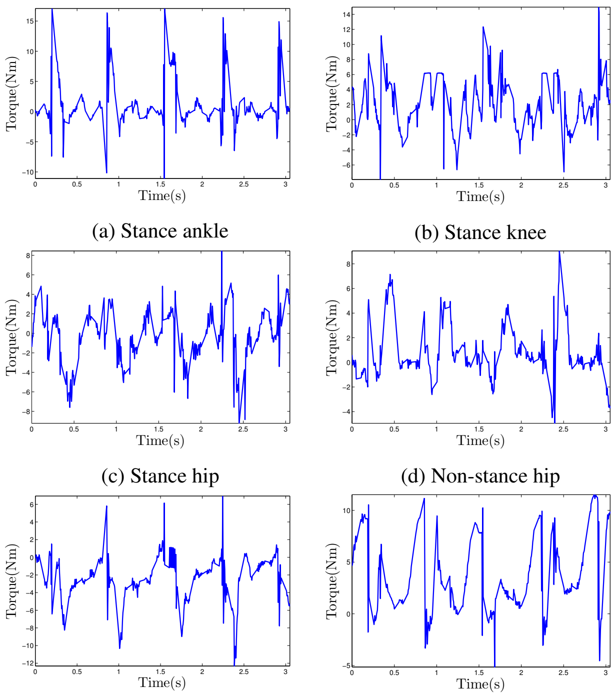


(a) Stance ankle, *rms* = 0.0459rad (b) Stance knee, *rms* = 0.0457rad

(c) Stance hip, *rms* = 0.0206rad (d) Nonstance hip, *rms* = 0.0435rad

(e) Nonstance knee, *rms* = 0.2033rad (f) Nonstance ankle, *rms*=0.0567rad

Figure 11: Actual vs. desired joint angles logged during AMBER2 walking with the unified control law, with *rms* the root mean square of tracking error.



(a) Stance ankle

(b) Stance knee

(c) Stance hip

(d) Non-stance hip

(e) Non-stance knee

(f) Non-stance ankle

Figure 12: Joint torque inputs logged during AMBER2 walking with PD control.

joint's requirement for higher torque input, which is needed to both support and roll the entire mass of the robot forward about the toe joint. Similar results can also be found in the studies of human gait analysis [5]. This level of performance, however, is beyond the hardware's ability. This further affects the speed of the robot, as the walking speed in under-actuated domain is slower than that in fully-actuated domain.

Despite the difference in walking speed, the robot displays qualitatively human-like walking with distinct multi-contact behaviors in a dynamic fashion. In comparison with the simulated walking with synchronized speed, the experimental walking gesture (i.e., walking pattern) matches up perfectly with the simulated walking, which can be seen in the video [16]. In conclusion, by utilizing the PHZD based human output reconstruction strategy with the optimized parameters β^*, AMBER2 has achieved human-like multi-domain locomotion, thus, fulfilling an important step of bridging the gap between theory and real world implementation.

9. REFERENCES

[1] Sustained robotic multi-domain walking of AMBER2. `http://youtu.be/VvkIdCKlL54`.

[2] M. Ackermann. Dynamics and energetics of walking with prostheses. 2007.

[3] A. D. Ames. First steps toward automatically generating bipedal robotic walking from human data. In *Robotic Motion and Control 2011*, volume 422 of *LNICS*, pages 89–116. Springer, 2012.

[4] A. D. Ames, R. Vasudevan, and R. Bajcsy. Human-data based cost of bipedal robotic walking. In *Hybrid Systems: Computation and Control*, Chicago, IL, 2011.

[5] S. Au, P. Dilworth, and H. Herr. An ankle-foot emulation system for the study of human walking biomechanics. pages 2939–45, 2006.

[6] J. A. Blaya and H. Herr. Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 2004.

[7] C. Chevallereau, D. Djoudi, and J. W. Grizzle. Stable bipedal walking with foot rotation through direct regulation of the zero moment point. *Robotics, IEEE Transactions on*, 24(2):390–401, 2008.

[8] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet. 3D bipedal robotic walking: models, feedback control, and open problems. In *IFAC Symposium on Nonlinear Control Systems*, Bologna, 2010.

[9] Z. Huihua, S. Nadubettu Yadukumar, and A. D. Ames. Bipedal robotic running with partial hybrid zero dynamics and human-inspired optimization. In *IROS, 2012 IEEE/RSJ International Conference on*, pages 1821–27. IEEE, 2012.

[10] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, 1994.

[11] C. Shih, J. W. Grizzle, and C. Chevallereau. Asymptotically stable walking of a simple underactuated 3D bipedal robot. In *33rd Annual Conf. of the IEEE Industrial Electronics Society (IECON)*, pages 2766–71, Taipei, 2007.

[12] R. W. Sinnet, M. J. Powell, R. P. Shah, and A. D. Ames. A human-inspired hybrid control approach to bipedal robotic walking. In *18th IFAC World Congress*, pages 6904–11, 2011.

[13] M. Vukobratović and B. Borovac. Zero-moment point – thirty-five years of its life. *Intl. J. of Humanoid Robotics*, 1(1):157–173, Mar.

[14] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, 2007.

[15] S. N. Yadukumar, M. Pasupuleti, and A. D. Ames. From formal methods to algorithmic implementation of human inspired control on bipedal robots. In *Tenth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2012.

[16] H. Zhao, M. Powell, and A. D. Ames. Human-inspired motion primitives and transitions for bipedal robotic locomotion in diverse terrain. *Optimal Control Applications and Methods*, 2013.