

Human-Inspired Walking via Unified PD and Impedance Control

Wen-Loong Ma¹, Hui-Hua Zhao¹, Shishir Kolathaya¹ and Aaron D. Ames²

Abstract—This paper describes a torque control scheme unifying feedback PD control and feed-forward impedance control to realize human-inspired walking on a novel planar footed bipedal robot: AMBER2. It starts with high fidelity modeling of the robot including nonlinear dynamics, motor model, and impact dynamics. Human data is then used by an optimization algorithm to produce a human-like gait that can be implemented on the robot. To realize the bipedal walking, first a PD controller is utilized to track the optimized trajectory. Next, impedance control parameters are estimated from the experimental data. Finally, the unified PD, impedance torque control law is experimentally realized on the bipedal robot AMBER2. Through the evidence of sustainable and unsupported walking on AMBER2 showing high consistency with the simulated gait, the feasibility of AMBER2 walking scheme will be verified.

I. INTRODUCTION

Due to the complexity present in achieving bipedal robotic walking, its study is often split into two extremes: theoretical results aimed at developing torque controllers (e.g., controlled symmetries [18], geometric reduction [6], [17], inverted pendulum [14], [10]) that are provably correct, and simulation/experimental results guided by heuristics (e.g., ZMP methods [21], [22], passivity based control [16], [11], reinforcement learning [12] and the central pattern generators [15]) that provide better real world behavior than complex nonlinear controllers can achieve. Both of these extremes are important in the study of robotic walking, yet to achieve truly human-like walking it is necessary to bridge the gap between these two methodologies. While gains have been made toward this goal, most notably through the application of hybrid zero dynamics to achieve robotic walking and running [8], [19], [23], novel methods are still needed to unify theoretical results with experimental realization.

We begin by introducing a high fidelity model of AMBER2 in Sect. II, which includes all of the most relevant aspects of the robot: nonlinear dynamics, models of the motors and boom, and impact dynamics. The end result is a hybrid system model of the bipedal robot, with motor torques as input. Based on this model, the human-inspired optimization problem, subject to certain physical constraints (Sect. III) that provably guarantee robotic walking, is developed (this methodology has been successfully applied to

*This research is supported by NSF awards CPS-1239085, CNS-1136104 and CNS-0953823.

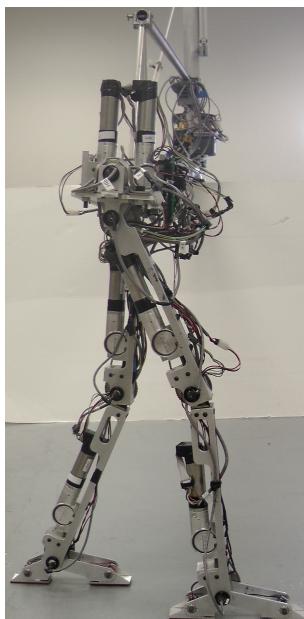
¹WL.Ma, HH.Zhao, and S.Kolathaya are with the department of Mechanical Engineering, Texas A & M University, 3128 TAMU, College Station, USA wenlongma, huihuazhao, shishirny@tamu.edu

²Dr. A.Ames is with department of Mechanical Engineering and Electrical Engineering, Texas A & M University, 3128 TAMU, College Station, USA aames@tamu.edu

other bipedal robots including NAO [5] and AMBER [24]). The end results are the parameters for canonical walking functions [3] that produce human-like trajectories that are amendable to implementation on the physical robot.

The torque controller for the physical robot is formed by two elements: a feedback controller, which is a standard PD based torque controller tracking trajectories reconstructed from the partial hybrid zero dynamics (PHZD) reconstruction obtained via human-inspired optimization; and a feed-forward controller, which is an impedance controller obtained from fitting impedance parameters recorded from the torque profiles from an experimental walking gait. This formal unification of applying PD and impedance control is what differentiates this approach from others. Feed-forward control is a widely used strategy in the field of locomotion controller design. While it improves the performance and reduces the hysteresis of the system [7], it relies heavily on knowledge of the system and thus is sensitive to modeling error. However, in section IV, this novel impedance control scheme can avoid the shortages while keeping the benefits.

The main contribution of this paper is the implementation of unified PD, impedance human-inspired control approach on the physical robot AMBER2 and the experimental results achieved via this implementation. With the detailed introduction of the experiment design in section V including the



(a) Sideview of AMBER2.

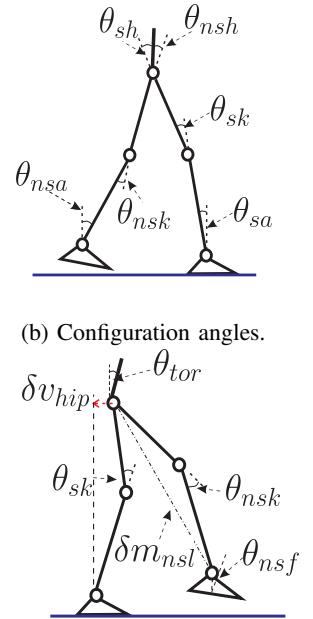


Fig. 1: The bipedal robot AMBER2 (a), robot joint angles (b), robot outputs (c).

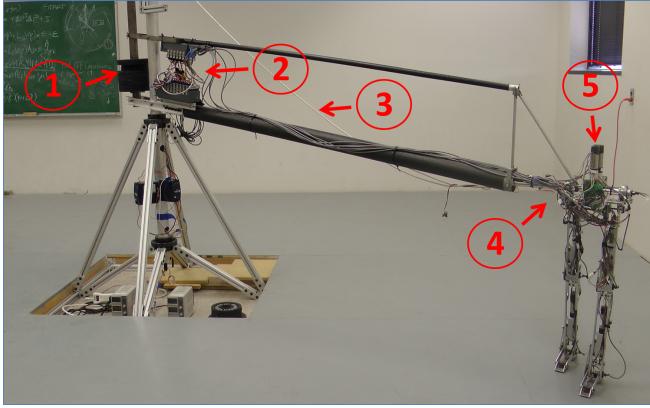


Fig. 2: AMBER2 with the boom and electronics. The boom restricts motion to the sagittal plane. As shown in the figure: (1) Counterweight used to balance the boom around the pivot, (2) Controller module where the walking algorithm is running, (3) The boom, (4) Boom support structure which keeps the torso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

pseudocode, state machine and block diagram of the low-level controller, the framework of the AMBER2 walking strategy is verified in both theory and practice. To highlight the benefits of the unified PD, impedance controller, we compare the results to that of AMBER2 walking only with a PD controller. Finally, the unification of simulated and experimental walking shows the consistency between formal theoretical methods and experimental implementation.

II. AMBER2 MODEL

AMBER2 is a 2D bipedal robot with seven links (two calves, two thighs, two feet and a torso, see Fig. 1a) as a second generation robot and expansion of its predecessor, the non-footed bipedal robot, AMBER (see [24]). Each joint is actuated by a BLDC motor. In addition, the motion is restricted to the sagittal plane via a boom shown in Fig. 2, which is configured as a parallel four-link mechanism so that the boom support structure is always horizontal. The boom is fixed rigidly to a rotating mechanism, which allows the biped to walk in a circle with minimal friction. In addition, counterweights are provided to cancel out the weight on the robot due to the boom. The motor H-bridges are located close to the pivot of the boom along with the other sensing and controller modules supplied by National Instruments. The modules are remotely connected to the stationary power supply with the help of slip rings located below the pivot.

Continuous Dynamics: Let $(\theta_{sa}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa}) \in Q \subset \mathbb{R}^6$ be the angles of the stance ankle (ankle of the stance leg), stance knee, stance hip, non-stance hip, non-stance knee and non-stance ankle respectively (see Fig. 1b) and L_c, L_t be the lengths of the calf and thigh respectively (values are given in Table. I). Given the configuration $\theta = (\theta_{sa}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa})^T \in Q$, and calculating the mass and inertia properties of each link of the robot through a

Solidworks model yields the Lagrangian:

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

The AMBER2 model also includes the motors and the boom. The way the inertias of the two elements are included in the model is slightly different. This approach was first considered in [24], and will be revisited here. I_r is the rotational inertia of the rotor and I_g is the rotational inertia of the gearbox. Due to the large gear ratio, I_g is small and ignored in the calculation. Similarly, the distance between the axis of rotation of the rotor and the corresponding joint is small. In addition, the mass of the rotor is small, resulting in the fact that the inertia of the motor w.r.t the joint axis can be approximated to be the inertia w.r.t the rotor axis. Since the biped end of the boom can move up-down and forward-backward, the boom exhibits yaw and roll about the pivot. This would correspond to the x and z component of the velocities of the torso. If I_{boom} is the inertia of the boom taken from the pivot, then 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. Thus the new combined mass inertia matrix, D_{com} , used in the Lagrangian will be:

$$D_{com}(\theta) = D(\theta) + diag(0, I_{m,sk}, I_{m,sh}, I_{m,nsh}, I_{m,nsk}, I_{m,nsa}) + J(\theta)^T M_{boom} J(\theta), \quad (2)$$

where $I_{m,sk}, I_{m,sh}, I_{m,nsh}, I_{m,nsk}$ correspond to the scaled motor inertia of respective links and $J(\theta)$ is the body Jacobin of the center of mass of the torso. Using the modeling techniques presented, we can realize the Euler-Lagrange equations in the following manner:

$$D_{com}(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) = B(\theta)u,$$

where u is a vector of torque inputs. Converting the equations of motion to a first order ODE yields the control system (f, g) with the form of $\dot{x} = f(x) + g(x)u$, where $x = (\theta, \dot{\theta})$. Impact dynamics are also included in modeling (see [23] for further details).

III. HUMAN-INSPIRED TRAJECTORY CONSTRUCTION

This section reviews *human-inspired optimization* so as to properly frame the formal results that are utilized to experimentally achieve robotic walking. Specifically, we review the

Model Parameters				
Parameter	Mass g	Length m	Inertia x-axis $\times 10^3 \text{ g m}^2$	Inertia z-axis $\times 10^3 \text{ g m}^2$
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 I: The mass parameters for each link of the robot.

formal results from [4] (also see [3], [5] for related results in the case of full actuation) with a view toward torque control.

Human-Inspired Outputs. To achieve human-like walking, we begin by seeking “outputs” of human locomotion [3]. Six outputs are considered for the 6-DOF robot: $\delta p_{\text{hip}}(\theta)$, the linearized position of the hip; θ_{sk} , the stance knee angle; θ_{nsk} , the non-stance knee angle; m_{nsl} , the linearized slope of the non-stance leg; $\theta_{tor}(\theta)$, the torso angle from vertical, and θ_{nsf} , the angle of the non-stance foot w.r.t the horizontal, as denoted in Fig. 1b. Analysis of the chosen outputs calculated from human data indicates that $\delta p_{\text{hip}}^d(t, v)$ is a linear function of time $\delta p_{\text{hip}}^d(t, v) = v_{\text{hip}}t$, and the other outputs can be characterized by the solution of a linear mass-spring-damper system, termed the *canonical walking function* (CWF):

$$y_H(t, \alpha) = e^{-\alpha_1 t} (\alpha_2 \cos(\alpha_3 t) + \alpha_4 \sin(\alpha_3 t)) + \alpha_5, \quad (3)$$

where a detailed explanation can be found in [3]. Based upon the evolution of the linearized hip position, we parameterize the time as:

$$\tau(\theta) = (\delta p_{\text{hip}}(\theta) - \delta p_{\text{hip}}(\theta^+))/v_{\text{hip}}, \quad (4)$$

which removes the dependence of time in (3) and renders an autonomous system [23]. Note that, θ^+ represents the robot configuration of the beginning of a step. The human-inspired outputs can then be defined as:

$$y_\alpha(\theta, \dot{\theta}) = \begin{bmatrix} y_1(\theta, \dot{\theta}) \\ y_2(\theta) \end{bmatrix} = \begin{bmatrix} y_{a,1}(\theta, \dot{\theta}) - v_{\text{hip}} \\ y_{a,2}(\theta) - y_{d,2}(\tau(\theta), \alpha) \end{bmatrix}, \quad (5)$$

where $y_1(\theta, \dot{\theta})$ is the relative degree one output, which is the difference between the actual forward hip velocity $y_{a,1}(\theta, \dot{\theta})$ and the desired hip velocity v_{hip} , and $y_2(\theta)$ are the relative degree two human-inspired outputs which are the difference between the actual relative degree two outputs $y_{a,2}(\theta)$ and desired relative degree two outputs $y_{d,2}(t, \theta)$, defined as:

$$y_{d,2}(t, \alpha) = \begin{bmatrix} y_H(t, \alpha_{sk}) \\ y_H(t, \alpha_{nsk}) \\ y_H(t, \alpha_{nsl}) \\ y_H(t, \alpha_{tor}) \\ y_H(t, \alpha_{nsf}) \end{bmatrix}, \quad y_{a,2}(\theta) = \begin{bmatrix} \theta_{sk} \\ \theta_{nsk} \\ \delta m_{nsl}(\theta) \\ \theta_{tor}(\theta) \\ \theta_{nsf}(\theta) \end{bmatrix}, \quad (6)$$

where $\alpha = (v_{\text{hip}}, \alpha_{sk}, \alpha_{nsk}, \alpha_{nsl}, \alpha_{tor}, \alpha_{nsf}) \in \mathbb{R}^{26}$ is the vector of the grouped parameters. Note that $y_{a,2}(\theta)$ is linear in joint angles θ , and can be written as $y_{a,2}(\theta) = H\theta$, which will be used later in this paper.

Partial Hybrid Zero Dynamics. Of particular interest in robotic walking are the relative degree 2 outputs, $y_2(\theta) = y_{a,2} - y_{d,2}$. The surface for which the actual and desired outputs agree for all time is given by the *partial zero dynamics surface*:

$$\mathbf{PZ}_\alpha = \{(\theta, \dot{\theta}) \in TQ : y_2(\theta) = \mathbf{0}, L_f y_2(\theta, \dot{\theta}) = \mathbf{0}\}. \quad (7)$$

Importantly, a feedback linearization controller can easily render this surface stable and invariant for the continuous dynamics. However, this may not be true for a hybrid system with impacts. The goal of *partial hybrid zero dynamics* (PHZD) is to find parameters α that ensure this surface

remains invariant through impact: $\Delta(S \cap \mathbf{PZ}_\alpha) \subset \mathbf{PZ}_\alpha$. This constraint motivates an optimization problem that guarantees this condition.

Human-Inspired Optimization. Aiming at finding the walking parameters α , which deliver provably stable robotic walking, an optimization problem subject to PHZD and other physically realizable constraints is given by:

$$\alpha^* = \underset{\alpha \in \mathbb{R}^{26}}{\operatorname{argmin}} \operatorname{Cost}_{\text{HD}}(\alpha) \quad (\text{HIO})$$

$$\text{s.t } \text{PHZD} \quad (\text{C1})$$

$$\text{Physical Constraint} \quad (\text{C2})$$

where, the cost function (HIO) is the least squares fit between the human experimental data and the CWF representation [5]. Note that, in addition to the PHZD constraint which guarantees exponentially stable orbits in hybrid systems [3], we also consider several physical constraints such that the optimized result can be used for the specific physical robot. In particular, the following two physical constraints are considered:

- 1) **Torque Constraints.** The maximum torque input limited by the capacity of the hardware should be respected in the optimized walking gait.
- 2) **Foot Scuffing Conditions.** The swing foot height clearance and stride length during the swing phase must be sufficient enough to avoid scuffing amidst sensor noise, tracking error, uneven ground and even imperfection in the mechanical design.

This optimization problem ends with a feedback linearization control law that results in provably stable robotic walking for the hybrid system model of AMBER2 (see [3] for a proof which easily extends to the case of AMBER2) and the α parameters representing a walking gait that best fit human-walking data while enforcing the desired constraints, which will be used on the physical robot AMBER2.

IV. CONTROLLER DESIGN

Having constructed the human-inspired trajectory from optimization, the objective of this section is to design the appropriate controller to achieve the desired walking. However, in order to realize robotic walking, state based PHZD reconstruction methodology needs to be introduced first.

A. PHZD Reconstruction

The idea is to find the desired joint angles and angular velocities of the robot in every iteration through an inverse projection from the PHZD surface. Given the PHZD surface, the coordinates can be defined as:

$$\xi_1 = \delta p_{\text{hip}}(\theta) := c\theta \quad (8)$$

$$\xi_2 = y_1^a(\theta, \dot{\theta}) := \delta \dot{p}_{\text{hip}}^R(\theta) := c\dot{\theta}$$

where c is obtained from (4). Since ξ_1 is the linearized position of the hip, which is used to parameterize time as (4),

we can write the desired outputs $y_{d,2}(\tau(\theta), \alpha) = y_{d,2}(\xi_1, \alpha)$. We can also write the actual outputs as:

$$\begin{aligned}\eta_1 &= y_{2,a} = H\theta \\ \eta_2 &= L_f y_{2,a}(\theta, \dot{\theta}) = H\dot{\theta}\end{aligned}\quad (9)$$

Then we can use PHZD to obtain an approximation of the solution to the full-order system. On the partial zero dynamics surface, the actual outputs are equal to the desired outputs. Therefore we have the following relationship between the desired joint angles and velocities and the desired outputs:

$$\begin{aligned}\theta_d(\tau) &= \Psi(\xi_1, \alpha) = \begin{bmatrix} c \\ H \end{bmatrix}^{-1} \begin{pmatrix} \xi_1 \\ y_{d,2}(\xi_1, \alpha) \end{pmatrix} \\ \dot{\theta}_d(\tau) &= \Phi(\xi_1, \xi_2, \alpha) = \begin{bmatrix} c \\ H \end{bmatrix}^{-1} \begin{pmatrix} v_{hip} \\ \frac{\partial y_{d,2}(\xi_1, \alpha)}{\partial \xi_1} \xi_2 \end{pmatrix}\end{aligned}\quad (10)$$

Since the system is fully actuated, feedback linearization results in linear dynamics when the PHZD conditions are satisfied. It follows that the relative degree 1 output evolves according to $\dot{y}_1 = -\varepsilon y_1$. Therefore, because of the definition of the partial zero dynamics, the partial hybrid zero dynamics evolve according to the linear ODE:

$$\begin{aligned}\dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= -\varepsilon(\xi_2 - v_{hip})\end{aligned}\quad (11)$$

This system can be solved in closed form, and by knowing ξ_1 , ξ_2 , the desired angles and velocities are obtained from (10). In other words, since θ_d , $\dot{\theta}_d$ are derived from the outputs $y_1(\theta, \dot{\theta})$ and $y_2(\tau, \alpha)$, tracking these joint angles on the robot is equivalent to tracking the desired human-inspired outputs.

B. Feedback PD control

To achieve a walking gait on hardware, a PD controller is utilized to track joint trajectories obtained from PHZD reconstruction:

$$\tau_{PD}^f = K_p(\theta_a - \theta_d) + K_d(\dot{\theta}_a - \dot{\theta}_d)\quad (12)$$

where K_p and K_d are proportional and derivative constant matrices that depend specifically on corresponding motors.

C. Feed-forward Impedance Control.

In this section, we will demonstrate that one of the most popular approaches in the prosthesis control field—impedance control—can be utilized as a feed-forward term in a robotic walking controller. We then discuss the algorithm for impedance parameter estimation.

Impedance Control. Based on the pioneering work of impedance control by Hogan [9], the torque at each joint during a single step can be represented in a piecewise fashion by a series of passive impedance functions [20]:

$$\tau = k(\theta - \theta^e) + b\dot{\theta}. \quad (13)$$

Inspired by the previous work [2], we first analyze AMBER2 walking data (which is achieved by using PD control), which implies that one gait cycle can be divided into four phases based on the knee joints, which are denoted as $p = 1, 2, 3, 4$.

Specifically, each phase begins at time t_0^p and ends at t_f^p . The phase separation principle is similar as that in [2] but with values specific to the gait of AMBER2. The impedance torque for specific joint i during a phase $p \in \{1, 2, 3, 4\}$, can be represented by the following equation:

$$\tau_{i,p}^f = k_{i,p}(\theta_i(t) - \theta_{i,p}^e) + b_{i,p}\dot{\theta}_i(t), \quad (14)$$

where $\theta_i(t)$ and $\dot{\theta}_i(t)$ denote angle and angular velocity of the joint i . Impedance parameters $k_{i,p}$, $\theta_{i,p}^e$ and $b_{i,p}$ represent the constant stiffness, equilibrium angle and damping respectively, which are constant during a specific phase p .

Impedance Parameter Estimation. In the previous work [2], the authors showed that the impedance parameters for a lower-limb prosthesis can be learned by the observation of unimpaired human walkers. These results have been validated in simulation and experimentally with a transfemoral prosthetic device. To identify the control parameters for each sub-phase, we again estimate the impedance parameters by observing the experimental data for AMBER2 walking with only PD control. We first define the impedance parameter set as $\beta_{i,p} = \{k_{i,p}, b_{i,p}, q_{i,p}^e\}$ for a specific joint i and sub-phase p . With the logged walking data $\{\theta_{i,p}^a, \dot{\theta}_{i,p}^a\}$ and torque data $\tau_{i,p}^a$ obtained from the experiment only using PD control, we can form the least square errors minimization problem as:

$$\beta_{i,p}^* = \underset{\beta_{i,p}}{\operatorname{argmin}} \int_{t_0^p}^{t_f^p} (\tau_{i,p}^f - \tau_{i,p}^a)^2 dt, \quad (15)$$

where $\tau_{i,p}^a$ is the actual experimental torque input on the joint i at sub-phase p . By solving this minimization problem for all the joints at different phases, the impedance parameters for the feed-forward impedance controller can be identified.

D. Control Law Construction

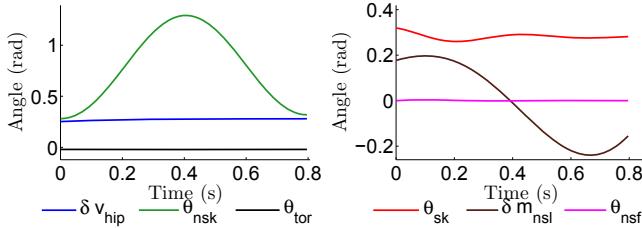
Finally, a unified PD-Impedance control approach is presented, where the PD based feedback controller is used to track the walking gait obtained formally through the human-inspired optimization; and impedance control forms the feed-forward controller which compensates for the nonlinear dynamics of the robot. This approach yields stable robotic walking in both simulation and physical experiments. The control law is defined as:

$$\tau^f = \tau_{PD}^f(\theta_a, \theta_d, \dot{\theta}_a, \dot{\theta}_d, K_p, K_d) + \tau_{i,p}^f(\theta_a, \dot{\theta}_a, k, b, q_e) \quad (16)$$

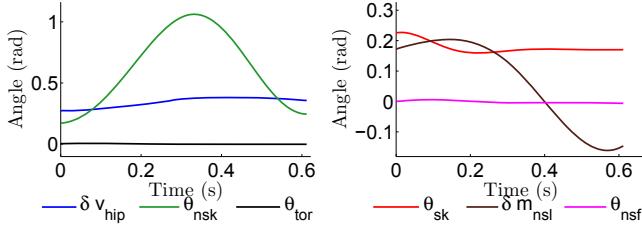
Specifically, although there are 36 gains used by impedance control and 12 gains for PD control, satisfactory tracking was achieved without any further gain tuning. In addition, due to the simplicity of the impedance controller, unlike other feed-forward methodologies such as nonlinear polynomials, it did not cause significant distortions in tracking.

V. EXPERIMENTAL REALIZATION

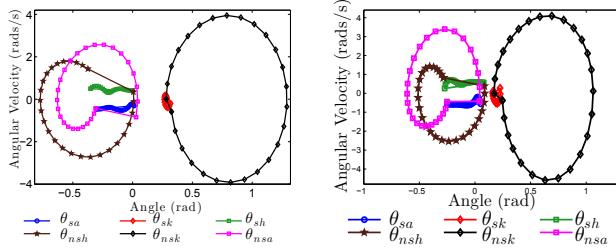
The controller implementation for AMBER2 has two levels: the high level controller which is realized in Real-Time (RT), and the low level controller realized by a Field-Programmable Gate Array (FPGA). The objective of this section is to introduce the control structure of AMBER2.



(a) Outputs of the robot for the linearizing controller.



(b) Outputs of the robot for the PD controller.



(c) Limit cycles associated with the walking gaits for the feedback linearizing controller (left) and the PD controller (right).

Fig. 3: Simulation results for the feedback linearizing and PD voltage controllers.

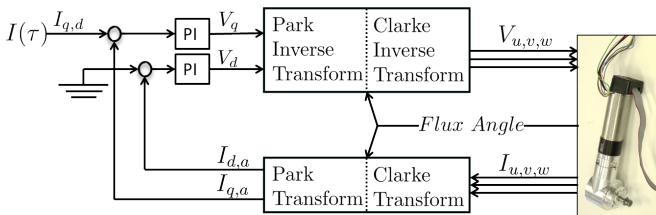


Fig. 4: Field-oriented control block diagram

High Level Controller. The Real Time control has the following major functionality incorporated: 1) Interface with the FPGA, read joint angles and angular velocities, send torque commands to the low level controller; 2) Compute the time parameter τ using (4); 3) Compute torque commands by applying the PD with impedance control law to the corresponding motors. Note that for AMBER2, the sampling rate is 143Hz. The high level controller is coded into shared libraries to interface with C++ so that the execution can be more efficient. The NI9144 EtherCAT Slave chassis is connected to the cRIO by EtherCAT to extend the capacity. For this configuration, each chassis is in charge of one leg. The pseudocode running in RT is shown in Algorithm 1.

Low Level Controller. The low level controller is coded into FPGA, which has the following major functionality: 1) Measure angular velocities by the incremental quadrature

Algorithm 1 Real Time Module

```

Input: AMBER2 Parameters: Calf Length( $L_c$ ), Thigh Length( $L_t$ );
Input: Optimization Parameters:  $\delta p_{\text{hip}}^R(\theta^+), v_{\text{hip}}, \alpha$ ;
Input: Calibration Results:  $\theta_{abs}$ 
Input: PD Controller Gain:  $K_p, K_d$ 
Input: Impedance Parameters:  $K_i, \theta_{i,e}, b_i, \tau_{flag}$ 
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: L/R stance; Encoder Status; Drive Status;
Output: Enable/Disable Motor Drives;
Output: Desired Torque for FOC;
1: Enable Motor Drives;
2: repeat
3:   Wait till all motor drives are Enabled
4: until ( Drive-Status == Enable )
5: while (  $\neg$  Stop-RT ) do
6:   Reform  $\theta, \dot{\theta}$  from Left/Right( $\theta_{LR}$ ) to Stance/nonStance( $\theta_{ShS}$ );
7:   Calculate actual time parameter  $\tau_a$  ;
8:   Desired  $\tau_d = \tau_a + \delta T$ ;
9:   Calculate(  $\xi_1, \xi_2$  );
10:  Calculate(  $y_d, \dot{y}_d$  );
11:  Calculate(  $\theta_d, \dot{\theta}_d$  );
12:  Apply PD Controller:
     $\tau_{PD}^f = K_p(\theta_d - \theta_a) + K_d(\dot{\theta}_d - \dot{\theta}_a)$ ;
13:  Based on  $\tau_a$  and  $\tau_{flag}$ , determine Impedance Phase;
14:  Apply Impedance Controller:
     $\tau_{imp}^f = \text{Impedance}(\theta, \dot{\theta}, K, \theta_e, b_i)$ ;
15:  Control Law Constructed:
     $\tau^f = \tau_{PD}^f + \tau_{imp}^f$ ;
16:  Change  $\tau^f$  from Stance/nonStance to Left/Right;
17:  Sending Torque Command to FPGA;
18:  Log Data into Remote Desktop;
19: end while
20: Disable Motor Drives;
21: Report Errors and Stop the Real Time VI;

```

encoders attached to every rotor; 2) Measure joint angles by using absolute encoders for home position and integrating velocity data for angle increment; 3) Determine the stance foot using the two switches located on each foot (one in front and the other one in back). The foot logic is shown by the state machine (Fig. 5); 4) Execute hardware protection logic when any joint goes beyond its workspace; 5) To realize torque control on the motor level, field-oriented control (FOC) is employed to control the 6 BLDC motors. As shown in the block diagram in Fig. 4, the torque is translated to current first. Then the flux angle is computed from the hall sensor and incremental encoder data, whose initial configuration was determined by auto-phasing (see [13] for details). Finally, by applying a PI controller on the quadrature and direct current, the motors are actuated accordingly. The pseudocode is shown in Algorithm 2.

VI. RESULTS AND CONCLUSIONS

The proposed controller was first verified in simulation. Comparing with the simulated results of using the human-inspired controller as seen in Fig. 3a, we can see that the unified PD-impedance controller has achieved similar performance as seen in Fig. 3b. The phase portraits of using both methods show that stable walking in simulation has been achieved with both controllers.

When the suggested control method was then applied to the physical robot, it was shown that AMBER2 was able to achieve sustainable walking (see [1] for the video). As it is

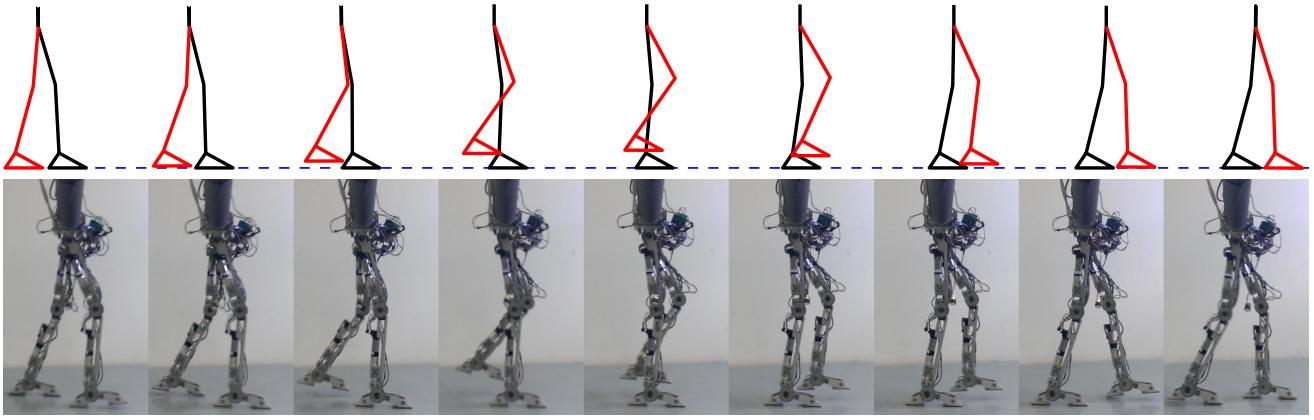


Fig. 8: Comparison of walking tiles of simulated and experimental walking with the unified PD, impedance control.

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, Torque 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;
1: loop
2:   Absolute Encoder Reading logic;
3:   if ( Signal low for 2 periods of encoder pulse) then
4:     Encoder Not Working  $\leftarrow 1$ ;
5:   else
6:     Encoder Not Working  $\leftarrow 0$ ;
7:   end if
8:   Incremental Quadrature Encoder Reading Logic;
9: end loop
10: loop
11:   Compute Desired Current from Torque Command from RT;
12:   if (Joint Angle exceeds Workspace and Torque Command not trying
      to stop it) then
13:     Reset Desired Current to 0;
14:   end if
15:   Compute Three Phase Voltage through Field-oriented Control Logic;
      (shown in Fig. 4)
16:   PWM signal Generation logic;
17: end loop
18: loop
19:   Guard and Stance Leg Detection Logic using foot contact switches
      (shown in Fig. 5);
20:   if ( Left Leg stance ) then
21:     L/R stance  $\leftarrow 0$ ;
22:   else if ( Right Leg stance ) then
23:     L/R stance  $\leftarrow 1$ ;
24:   end if
25: end loop

```

shown in the walking tiles in Fig. 8 and the experimental data of comparison between actual and desired values of different joints in Fig. 7, the walking achieved experimentally agrees with the simulated walking, with a maximum tracking error of 0.12 rad. Experimental results of walking only with PD controller are also included Fig. 6. The maximum walking distance only with PD controller is approximately 30 meters, whereas AMBER2 can walk more than 100 meters with

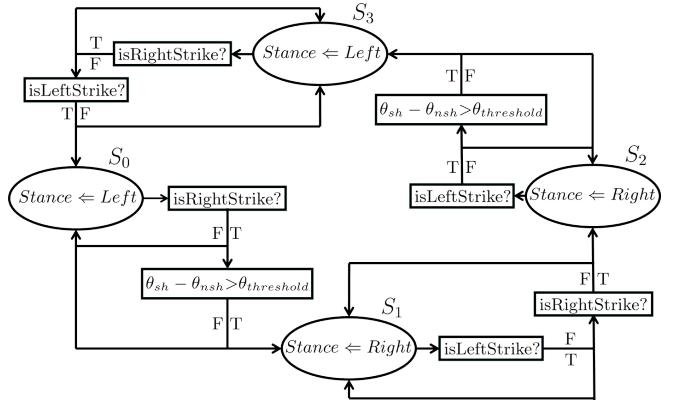
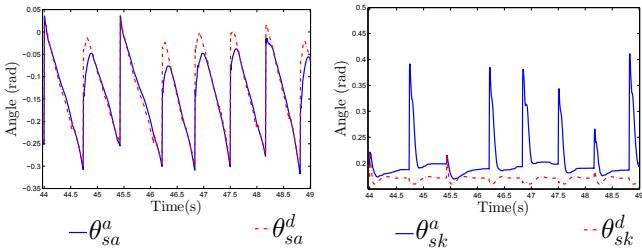


Fig. 5: State machine showing the foot contact and the logic used to determine the stance leg.

the unified controller without any indication of falling. That being said, the unified controller not only realized better tracking performance but also more robustness. It is very important to note that the system is developed with minimum sensing requirements, utilizing only foot contact switches and absolute and incremental encoders. The inherent spring-damper responses embedded in the CWF and the design method adopted for the robot facilitated the ease of applying such simple control laws to realize walking. These factors also combined to yield low torque consumption throughout the step. During continuous walking, the maximum torque input for the ankle, knee and hip motors were 5Nm, 5Nm and 10Nm respectively. In conclusion, the synchronization between the simulated and experimental walking and the small tracking error prove that the optimization algorithm and the unified control approach is correct and efficient. In other words, AMBER2 has initialized an important step toward bridging the gap between formal theoretical methods and real world implementation.

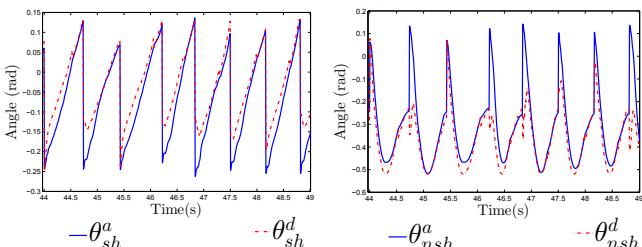
REFERENCES

- [1] Sustained robotic walking of AMBER2.
<http://youtu.be/d6oM5sLI9vA>.
- [2] N. Aghasadeghi, H. Zhao, L. J. Hargrove, A. D. Ames, E. J. Perreault, and T. Bretl. Learning impedance controller parameters for lower-limb



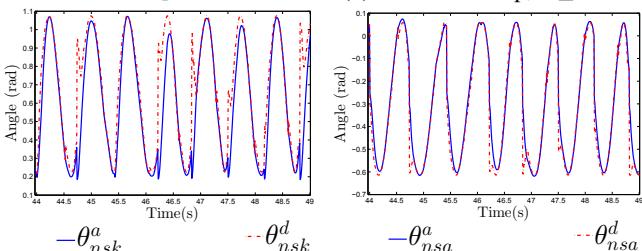
(a) Stance ankle, $e \leq 0.1\text{rad}$

(b) Stance knee, $e \leq 0.2\text{rad}$



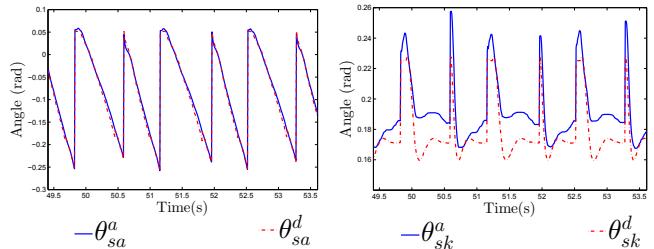
(c) Stance hip, $e \leq 0.2\text{rad}$

(d) Non-stance hip, $e \leq 0.6\text{rad}$



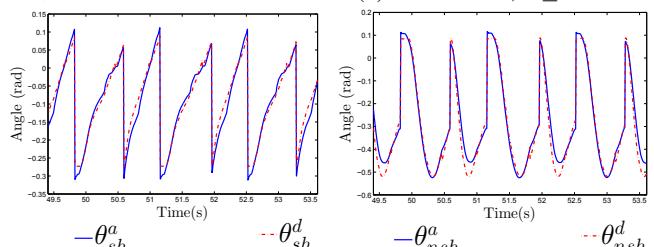
(e) Non-stance knee, $e \leq 0.8\text{rad}$ (f) Non-stance ankle, $e \leq 0.3\text{rad}$

Fig. 6: Actual vs. desired joint angles logged during AMBER2 walking with PD controller, with e the tracking error.



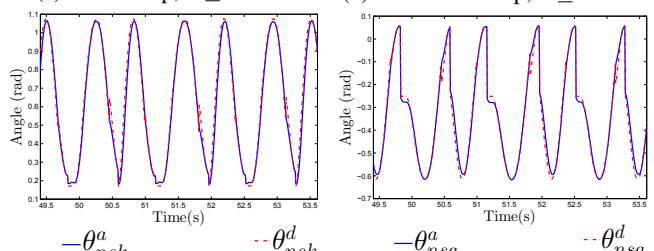
(a) Stance ankle, $e \leq 0.02\text{rad}$

(b) Stance knee, $e \leq 0.04\text{rad}$



(c) Stance hip, $e \leq 0.04\text{rad}$

(d) Non-stance hip, $e \leq 0.09\text{rad}$



(e) Non-stance knee, $e \leq 0.12\text{rad}$ (f) Non-stance ankle, $e \leq 0.09\text{rad}$

Fig. 7: Actual vs. desired joint angles logged during AMBER2 walking with the unified control law, with e the tracking error.

- prostheses. In *Intelligent Robots and Systems (IROS)*, pages 4268–4274, Nov 2013.
- [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. First steps toward underactuated human-inspired bipedal robotic walking. In *2012 IEEE Intl. Conf. on Robotics and Automation*, pages 1011–1017, St. Paul, Minnesota, May 2012.
 - [5] A. D. Ames, E. A. Cousineau, and M. J. Powell. Dynamically stable robotic walking with NAO via human-inspired hybrid zero dynamics. In *15th ACM Intl. Conf. on Hybrid Systems: Computation and Control*, pages 135–44, Beijing, April 2012. ACM.
 - [6] A. M. Bloch, D. Chang, N. Leonard, and J. E. Marsden. Controlled lagrangians and the stabilization of mechanical systems: Potential shaping. *IEEE Transactions on Automatic Control*, 46:1556–71, 2001.
 - [7] B. Joseph C. Brosilow. *Techniques of Model-Based Control*, chapter 9, Feedforward Control. Prentice Hall, 2002.
 - [8] C. Chevallereau, E.R. Westervelt, and J.W. Grizzle. Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. *International Journal of Robotics Research*, 24:431–64, 2005.
 - [9] N. Hogan. Impedance control: An approach to manipulation. pages 304–313, 1984.
 - [10] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *Siam Review*, 48(2):207–304, 2006.
 - [11] F. Iida and R. Tedrake. Minimalistic control of biped walking in rough terrain. *Autonomous Robots*, 28:355–365.
 - [12] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics & Automation*, New Orleans, LA, May, 2004.
 - [13] NI. Getting started with the NI 9502 brushless servo drive module.
 - [14] I. Poulakakis and J. W. Grizzle. The Spring Loaded Inverted Pendulum

as the Hybrid Zero Dynamics of an Asymmetric Hopper. *Transaction on Automatic Control*, 54(8):1779–1793, 2009.

- [15] S. Rutishauser, A. Sprowitz, L. Righetti, and A. J. Ijspeert. Passive compliant quadruped robot using central pattern generators for locomotion control. *IEEE RAS & EMBS International Conference*, pages 710–715, October.
- [16] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, 2005.
- [17] R.W. Sinet and A.D. Ames. Extending two-dimensional human-inspired bipedal robotic walking to three dimensions through geometric reduction. In *American Control Conference (ACC)*, 2012, pages 4831–4836, 2012.
- [18] M. W. Spong and F. Bullo. Controlled symmetries and passive walking. *IEEE Transactions on Automatic Control*, 50:1025–31, 2005.
- [19] K. Sreenath, H.W. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for achieving stable, efficient and fast bipedal walking on mabel. *Int. J. Robot. Res.*, September 2010.
- [20] F. Sup, A. Bohara, and M. Goldfarb. Design and Control of a Powered Transfemoral Prosthesis. *The International journal of robotics research*, 27(2):263–273, February 2008.
- [21] M. Vukobratović and B. Borovac. Zero-moment point – thirty-five years of its life. *Intl. J. of Humanoid Robotics*, 1(1):157–173, March.
- [22] M. Vukobratović, B. Borovac, D. Surla, and D. Stokic. *Biped Locomotion*. Springer-Verlag, Berlin, March 1990.
- [23] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, 2007.
- [24] S. Nadubettu 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)*, Cambridge, MA, 2012.