Human-inspired Walking via Unified PD and Impedance Control

Wen-Loong Ma[1], Hui-Hua Zhao[1], Shishir Kolathaya[1] and Aaron D. Ames[2]

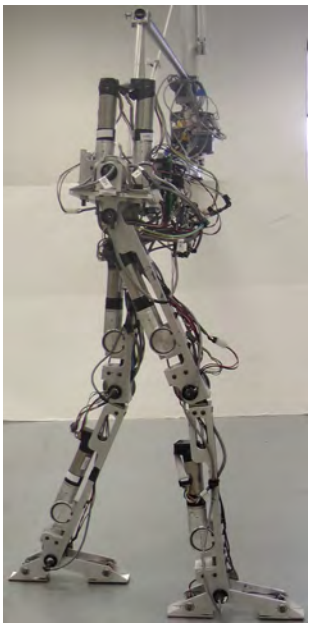
***Abstract*— This paper describes a torque control scheme unifying feedback PD control and feed-forward impedance control to realize human-inspired walking on a novel planar footed bipedal robot: AMBER2. It starts with high fidelity modeling of the robot including nonlinear dynamics, motor model, and impact dynamics. Human data is used by an optimization algorithm to develop a human-like gait for the robot, which is represented in the form of canonical walking functions. A PD controller is employed to track the optimized trajectory initially. Next, impedance control parameters are estimated from experimental data. Finally, the unified PD, impedance torque control law is experimentally realized on the bipedal robot AMBER2. Simulation results also verify that this control results in robotic walking on the formal model. The end result is sustainable unsupported walking that shows high consistency with the simulated gait.**

I. INTRODUCTION

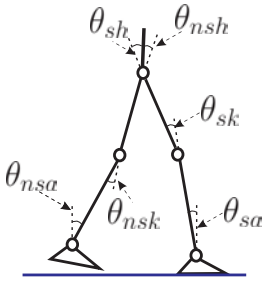
Due to the complexity present in achieving bipedal robotic walking, its study is often split between two extremes: theoretical results aimed at developing torque controllers (e.g., controlled symmetries [18], geometric reduction [6], [17], inverted pendulum [15], [11]) that are provably correct, and simulation/experimental results guided by heuristics (e.g., ZMP methods [21], [22], passivity based control [9], [12], reinforcement learning [14] and the central pattern generators [16]) that, often, 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 robotic 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.

Aiming to make the first step toward bridging the gap between theoretical and experimental results, this paper introduces a novel control framework for a planar bipedal robot with feet—AMBER2. We begin by introducing a high fidelity model of AMBER2 in Sect. II. In order to ensure agreement between the simulated behavior of this model and the behavior observed experimentally; this model 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 for the bipedal robot, for which the torque command of the motors is the input. Utilizing this particular model, the human-inspired optimization problem, subjected to certain physical constraints (Sect. III) that provably guarantee robotic walking, is developed (which has been successfully applied to other bipedal robots including NAO [5] and AMBER [24] as well), which yields parameters for canonical walking functions [3] that produce human-like trajectories.

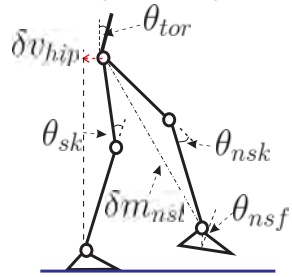
The torque controller for the physical robot is formed by two elements: a feedback controller, which is a standard PD based torque controller using trajectories from the partial hybrid zero dynamics (PHZD) reconstruction obtained via human-inspired optimization reconstruction, and a feed-forward controller, which is an impedance controller obtained from fitting impedance parameters to 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 can improve the performance and reduce 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, the utilization of a novel impedance control



(a) Sideview of AMBER2.



(b) Configuration angles.



(c) Outputs.

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

*This research is supported by NASA grant NNX11AN06H, NSF grants CNS-0953823 and CNS-1136104, and NHARP award 00512-0184-2009.

[1]WL.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

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

scheme avoids these 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 and the experimental results achieved with this implementation. With the detailed introduction of the experiment design in SectV, the framework of the AMBER2 walking strategy was verified in both theory and practice. To highlight the advantage of the unified PD, impedance controller, we compared these results to those of AMBER2 walking only with a PD controller. The method for achieving this walking is outlined through the presented pseudo-code, block diagram of the low-level controller and a state machine for the logic used. The end result is sustained and unsupported bipedal robotic walking on AMBER2. These experimental results are compared against the simulated walking and provide a bridge between the formal 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). AMBER2 is a second generation bipedal robot and an expansion upon its predecessor, the non-footed (point foot) bipedal robot, AMBER (see [24]). Each joint is actuated by brushless DC (BLDC) motors. In addition, with motion being restricted to the sagittal plane via a boom shown in Fig. 2, which are configured as parallel four-link mechanism, the boom support structure (4) in Fig. 2 is always horizontal. The boom is fixed rigidly to a rotating mechanism (see Fig. 2), which allows the biped to walk in a circle with minimal friction. In addition, counterweights are provided (see Fig. 2) 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. The joint angles of the robot are measured by PWM absolute MR encoders and single-ended incremental quadrature encoders and sent into the FPGA in the controller.

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 (knee of the stance leg), stance hip, non-stance (of the swing leg) hip, non-stance knee and non-stance ankle respectively (see Fig. 1b). These variables form the configuration space of the robot, and are shown in Fig. 1b. Let L_c, L_t be the lengths of the calf and thigh respectively (values are given in Table. I).

Continuous Dynamics: 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 allows for the construction of the Lagrangian:

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

The AMBER2 model also contains the motors and the boom. The way the inertias of these 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 is 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 inertia of the motor w.r.t the joint axis can be approximated to be the inertia w.r.t the rotor axis.

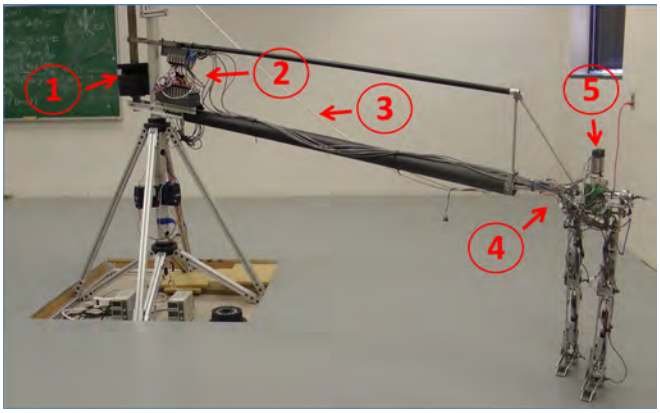


Fig. 2: AMBER2 with the boom and electronics. The boom restricts motion to the sagittal plane. As shown in the figure: **(1)** Counterweight used to balance the boom around the pivot, **(2)** Controller module where the walking algorithm is running, **(3)** The boom, **(4)** Boom support structure which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

Since the biped end of the boom can move up-down and forward-backward, 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 can be approximated to be at the center of the pivot. If I_{boom} is the inertia of the boom, then its mass matrix, $M_{boom} \in \mathbb{R}^{6\times6}$, is:

$$M_{boom} = \begin{bmatrix} \frac{I_{boom}}{L_{boom}^2} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix},$$

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

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

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

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

where $I_{m,sk}, I_{m,sh}, I_{m,nsh}, I_{m,nsk}$ correspond to the 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), which is in the form $\dot{x} = f(x) + g(x)u$, where $x = (\theta,\dot{\theta})$ Impact dynamics is 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 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. With the goal of achieving human-like walking, we begin with seeking "outputs" of the human locomotion data [3]. Six outputs are considered for the 6-DOF robot considered in this paper: $\delta p_{\text{hip}}(\theta)$, 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, which are also denoted in Fig. 1b.

Analysis of the chosen outputs data indicates that, the linearized hip position is a linear function of time $\delta p^d_{\text{hip}}(t,v) = v_{hip}t$, and the other outputs can be characterized by the solution of a linear mass-spring-damper system, which we term the *canonical walking function* (CWF):

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

where the detail explanations can be referred to [3]. Based on the linear fashion of the linearized hip position, we parameterized the time as:

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

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 the step.

With the autonomous CWF in hand, we define the human-inspired outputs:

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

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 $y_{d,2}(\theta)$, defined as:

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

where $\alpha = (v_{hip}, \alpha_{sk}, \alpha_{nsk}, \alpha_{nsl}, \alpha_{tor}, \alpha_{nsf}) \in \mathbb{R}^{26}$ 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 the 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 these outputs agree for all time is given by the *partial zero dynamics surface*

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

Importantly, a feedback linearization controller can easily render this surface stable and invariant for continuous system, 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 that this surface remains invariant through impact: $\Delta(S \cap \mathbf{PZ}_\alpha) \subset \mathbf{PZ}_\alpha$. This constraint motives the introduction of an optimization problem that guarantees this condition.

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

$$\begin{aligned} \alpha^* = \underset{\alpha\in\mathbb{R}^{26}}{\text{argmin}}\ & \text{Cost}_{\text{HD}}(\alpha) && \text{(HIO)} \\ \text{s.t}\quad & \textit{PHZD} && \text{(C1)} \\ & \text{Physical Constrains} && \text{(C2)} \end{aligned}$$

where, the cost function (HIO) is the least squares fit between the human experimental data and the CWF representations [5]. Note that, despite the PHZD constraints which guarantees exponentially stable orbits in hybrid systems [3], we also need to 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 Constrains.** 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.
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. Therefore, foot scuffing conditions have to be imposed to insure sustainable walking.

Above all, by solving this optimization problem, we can obtain α parameters that best fit human-walking data while

enforcing the desired constraints. The end result of this optimization is that the feedback linearization control law results in provable stable robotic walking for the hybrid system model of AMBER2 (see [3] for a proof which easily extends to the case of AMBER2). More importantly, the optimized parameters α will be shown to give human-like robotic walking on the physical robot AMBER2.

IV. CONTRLLER DESIGN

Having constructed the human-inspired trajectory from optimization, the objective of this section is to design the appropriate controller, which delivers provable walking with the robot. However, in order to realize robotic walking, State based partial hybrid zero dynamics (PHZD) reconstruction methodology needs to be introduced first, and then we will present the human-inspired controller.

A. PHZD Reconstruction

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

$$\begin{aligned} \xi_1 &= \delta p^R_{hip}(\theta) := c\theta \\ \xi_2 &= y^a_1(\theta,\dot\theta) := \delta \dot p^R_{hip}(\theta) := c\dot\theta \end{aligned} \tag{8}$$

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:

$$\begin{aligned} \eta_1 &= y_{2,a} = H\theta \\ \eta_2 &= L_{f^R} y_{2,a}(\theta,\dot\theta) = H\dot\theta \end{aligned} \tag{9}$$

Then we can use PHZD dynamics 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 joints angles and velocities and the desired outputs of the robot:

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

As a result of the fact that we have fully actuated and completely linearized dynamics, 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:

$$\begin{aligned} \dot\xi_1 &= \xi_2 \\ \dot\xi_2 &= -\varepsilon(\xi_2 - v_{hip}) \end{aligned} \tag{11}$$

Having known ξ_1,ξ_2, the desired angles and velocities are obtained from (10). In other words, since $\theta_d, \dot\theta_d$ are derived from the outputs $y_1(\theta,\dot\theta)$ and $y_2(\tau,\alpha)$, tracking these joint angles and velocities in robot is equivalent with tracking the outputs of the robot. Therefore, the restriction of the dynamics to the partial zero dynamics surface still maintained.

B. Feedback PD control

Based upon the theoretic methods discussed so far, the PD controller is employed to tracking joint trajectories obtained from PHZD reconstruction:

$$\tau^f_{PD} = K_p(\theta_a - \theta_d) + K_d(\dot\theta_a - \dot\theta_d) \tag{12}$$

where K_p and K_d are proportional and derivative constant matrices respectively. Note here, K_p and K_d matrices depend specifically on corresponding motors.

C. Feed-forward Impedance Control.

Impedance control is one of most popular approaches in the prosthesis control field. Attracted by these advantages, we demonstrate that impedance control can be also applied as a feed forward term for bipedal robotic control. In this section, we will introduce the impedance control first and then discuss the algorithm for impedance parameters estimation.

Impedance Control. Based on the pioneering work of impedance control by Hogan [10], the torque at each joint during a single step can be represented in a piecewise fashion by a series of passive impedance functions [20], which have the following form:

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

Inspired by the previous work [2], analysis of AMBER2 walking data (which is achieved by using PD control) shows 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^p_0 and ends at t^p_f. 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:

$$\tau^f_{i,p} = k_{i,p}(\theta_i(t) - \theta^e_{i,p}) + b_{i,p}\dot\theta_i(t), \tag{14}$$

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

Impedance Parameter Estimation. With the phase transitions defined above, the remaining problem is to identify the control parameters for each sub-phase. In the previous work [2], the authors showed that the impedance parameters for a lower-limb prosthesis can be learned by the observation from the unimpaired human walkers. The results have been validated both in the simulation and in the experiment with a transfemoral prosthetic device. To extend these results, we utilize a similar method to estimate the impedance parameters by observing the data of good walking steps of AMBER2 achieved by just using the PD controller.

We first define the impedance parameter set as $\beta_{i,p} = \{k_{i,p}, b_{i,p}, q^e_{i,p}\}$ for specific joint i and sub-phase p. With

the recorded walking data $\{\theta^a_{i,p}, \dot{\theta}^a_{i,p}\}$ and torque data $\tau^a_{i,p}$ obtained by utilizing the PD controller on AMBER2 in experiment, we can form the least square errors minimization problem as following:

$$\beta^*_{i,p} = \underset{\beta_{i,p}}{\operatorname{argmin}} \int_{t_0^p}^{t_f^p} (\tau^f_{i,p} - \tau^a_{i,p})^2 dt, \tag{15}$$

where $\tau^f_{i,p}$ is defined as (14) and $\tau^a_{i,p}$ is the actual experimental input torque on the joint i at sub-phase p. By solving this minimization problem for all the joints at different phases, we can obtain the estimated impedance parameters for the feed forward impedance controller.

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 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:

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

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.

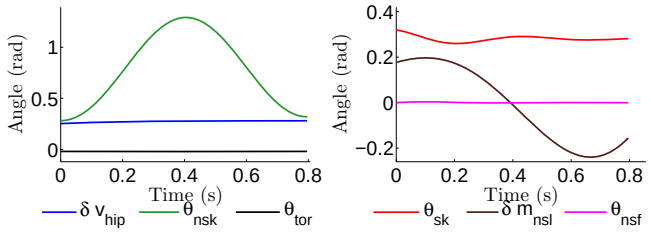
V. EXPERIMENTAL REALIZATION

The controller for AMBER2 has two levels: high level controller which is realized by Real-Time (RT) control, and low level controller realized by Field-Programmable Gate Array (FPGA). The objective of this section is to introduce the control structure of AMBER2. However, experimental setup is needed before robotic walking can be achieved: the calibration of absolute encoders and auto-phasing BLDC motors, which will determine the configuration of hall sensors, index angles and step angle increment of incremental encoders (see [13] for details).

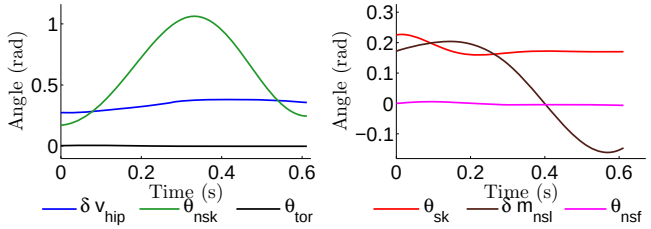
High Level Controller. The Real Time control has the following major functionality incorporated:

1) Interface with FPGA, read joint angles and angular velocities, send torque command to low level controller, enable/disable motors.
2) Compute the time parameter τ using (4).
3) Compute torque command by applying PD with impedance control law to corresponding motors, the desired torque input T^f_{in} is fed into the FPGA.

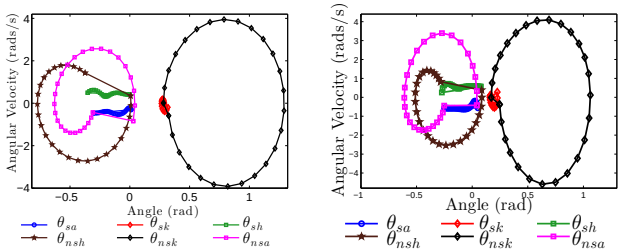
Note that for AMBER2, the sample rate and command rate are both 143Hz. The high level controller is coded into shared libraries to interface with C++ so that the execution can be more efficient. The NI9144 EhterCAT Slave chassis is connected to the cRIO by EtherCAT to extend the capacity.



(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.

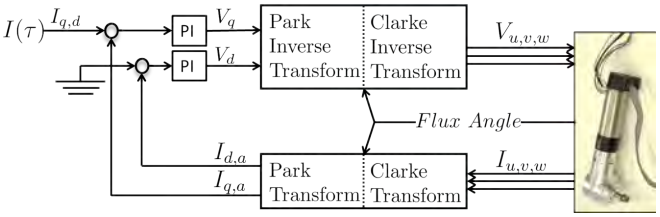


Fig. 4: Field-oriented control block diagram

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 in Field-programmable Gate Array (FPGA) with on board clock 40MHz, which is in charge of the following major functionality:

1) Measure Angular velocity by the single-ended incremental quadrature encoders attached to every rotor.
2) Measure joint angle by using PWM absolute encoders for home position and integrating velocity data for angle increment. In particular, both absolute encoders and incremental encoders operate at 40MHz.
3) Detect stance foot by the two switches placed on each foot (one in front and the other one in back). Foot logic is shown in the state machine Fig. 5.
4) Execute hardware protection logic when any joint is

Algorithm 1 Real Time Module

Input: AMBER2 Parameters: Calf Length(L_c), Thigh Length(L_t);
Input: Optimization Parameters: $\delta p^R_{\text{hip}}(\theta^+), v_{\text{hip}}, \alpha$;
Input: Calibration Results: θ_{abs}
Input: PD Controller Gain: K_p, K_d
Input: Impedance Parameters: K_i, $\theta_{i,e}$, b_i, τ_{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;

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(θ_{LR}) to Stance/nonStance(θ_{SnS});
    Calculate actual time parameter τ_a ;
    Desired $\tau_d = \tau_a + \delta T$;
    Calculate(ξ_1, ξ_2);
    Calculate(y_d, $\dot{y}_d$);
    Calculate(θ_d, $\dot{\theta}_d$);
    Apply PD Controller:
        $\tau^f_{PD} = K_p(\theta_a - \theta_d) + K_d(\dot{\theta}_a - \dot{\theta}_d)$;
    Based on τ_a and τ_{flag}, determine *Impedance Phase*;
    Apply Impedance Controller:
        $\tau^f_{imp} = Impedance(\theta,\ \dot{\theta},\ K,\ \theta_e,\ b_i)$;
    Control Law Constructed:
        $\tau^f = \tau^f_{PD} + \tau^f_{imp}$;
    Change τ^f 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;

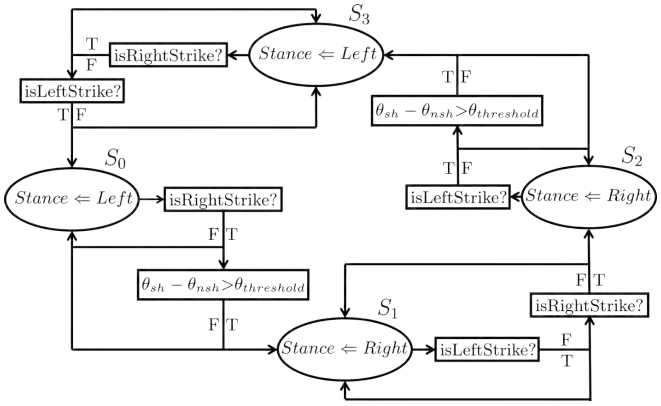


Fig. 5: State machine showing the foot contact and the logic used to determine the stance leg.

trying to go beyond its working space, which is done by resetting torque command to zero.

5) Torque control. To realize torque control on the motor level, field-oriented control (FOC) is employed to control the 6 BLDC motors. As shown in the control 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, which are initialized by auto-phasing. Finally, by applying a PI controller on the quadrature and direct current, the motors are actuated accordingly.

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

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;

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. 4) // Operation Frequency: 40MHz
    PWM signal Generation logic;
end loop
loop
    Guard and Stance Leg Detection Logic using foot contact switches (shown in Fig. 5);
    if (Left Leg stance) **then**
        L/R stance ← 0;
    else if (Right Leg stance) **then**
        L/R stance ← 1;
    end if
end loop

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 methodology was then applied to the physical robot, it is shown that AMBER 2 was able to achieve sustainable walking (see [1] for the video). The gait tiles, Fig. 8, show good agreement between theory and simulation, and the comparison between actual and desired values of different joints are shown in Fig. 7 and Fig. 10. The walking achieved experimentally agrees with the walking predicted in simulation, with a maximum tracking error of 0.12 rad. Experimental results of waking only with PD controller are also included Fig. 6. Maximum walking distance only with PD controller is 30 meters on record, whereas AMBER2 can walk more than 100 meters with the unified controller without any indication of falling. That being said, the unified controller not only brought better tracking performance but also robustness. It is very

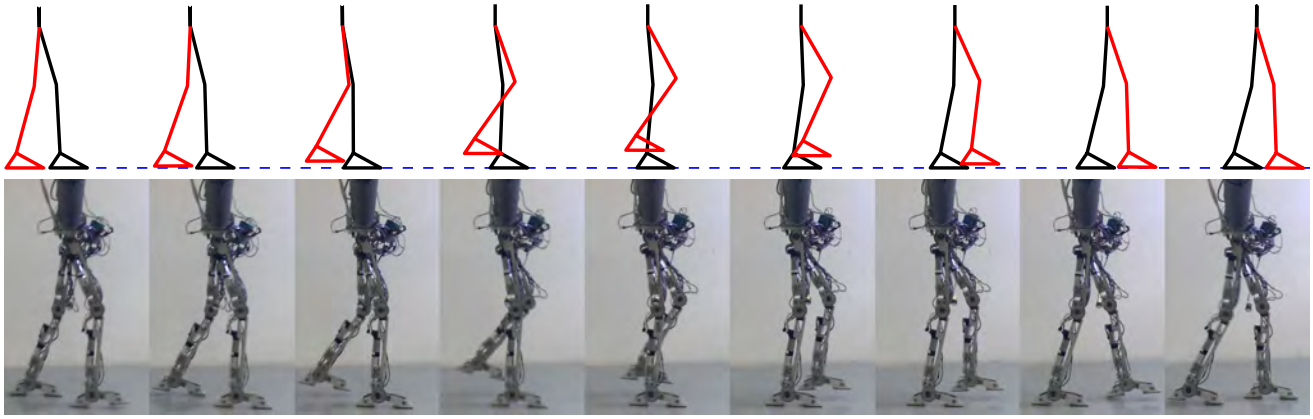
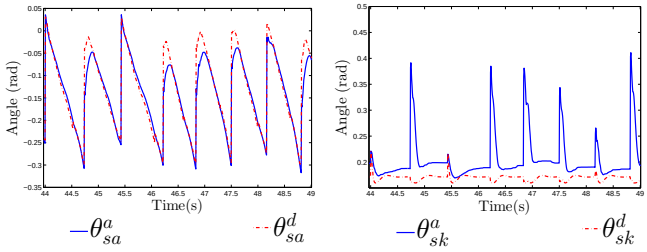
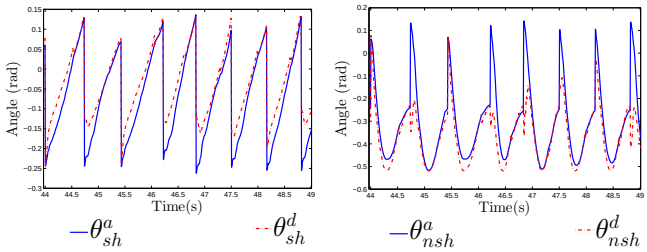


Fig. 8: Comparison of walking tiles of simulated and experimental walking with the unified PD, impedance control.



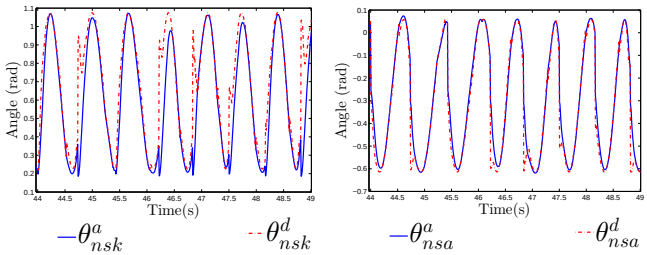
(a) Stance ankle, $e \leq 0.1$rad

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



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

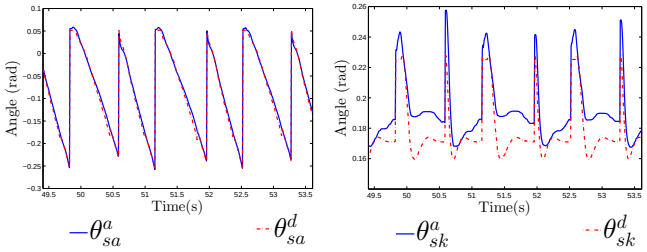
(d) Non-stance hip, $e \leq 0.6$rad



(e) Non-stance knee, $e \leq 0.8$rad

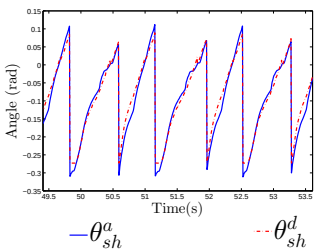
(f) Non-stance ankle, $e \leq 0.3$rad

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

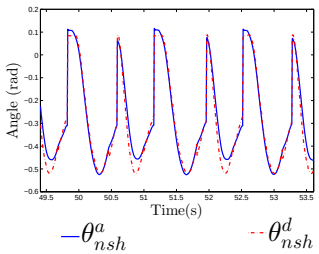


(a) Stance ankle, $e \leq 0.02$rad

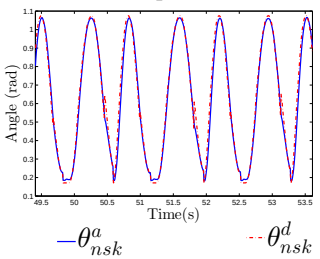
(b) Stance knee, $e \leq 0.04$rad



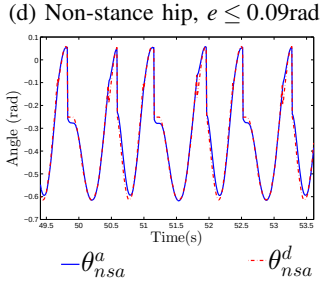
(c) Stance hip, $e \leq 0.04$rad



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



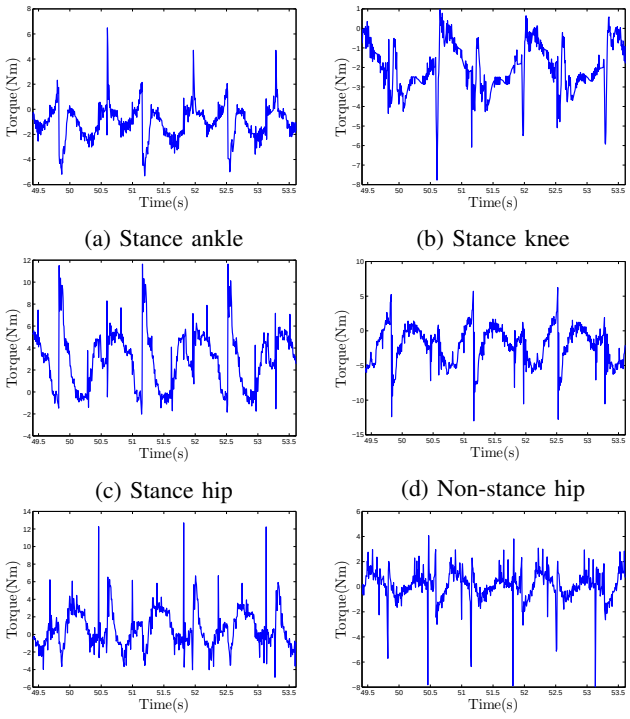
(e) Non-stance knee, $e \leq 0.12$rad



(f) Non-stance ankle, $e \leq 0.09$rad

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

important to note that the system is developed with minimum sensing requirements, foot contact switches, absolute and incremental encoders. The inherent spring-damper responses imbibed in the CWF and the methodology of design adopted for the robot facilitated the ease of applying such simple control laws to realize walking, which also results in low torque consumption throughout the step. During continuous walking, maximum torque input for ankle, knee and hip motors are 5Nm, 5Nm, 10Nm accordingly (Fig. 9). In conclusion, the synchronization between simulated walking and implementation as shown in the video and the small tracking error shows that the optimization algorithm and the unified control approach suggested is correct and efficient. In other words, AMBER2 has fulfilled an important step bridging the gap between theory and real world implementation.

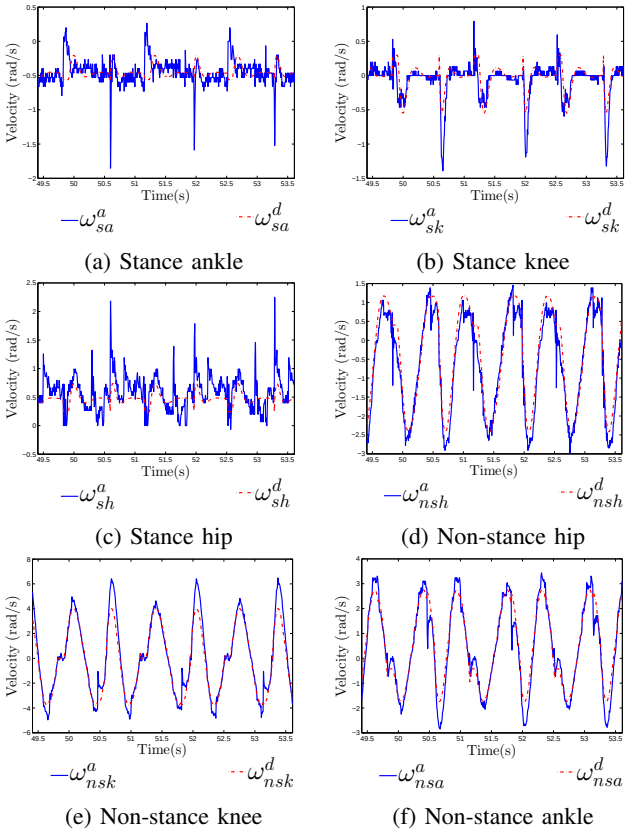


(a) Stance ankle (b) Stance knee

(c) Stance hip (d) Non-stance hip

(e) Non-stance knee (f) Non-stance ankle

Fig. 9: Joint torque inputs logged during AMBER2 walking with PD, impedance control.



(a) Stance ankle (b) Stance knee

(c) Stance hip (d) Non-stance hip

(e) Non-stance knee (f) Non-stance ankle

Fig. 10: Actual vs. desired joint angular velocities logged during AMBER2 walking with PD, impedance control.

REFERENCES

[1] Sustained robotic walking of AMBER2. http://youtu.be/d6oM5sLI9vA.

[2] Navid Aghasadeghi, Huihua Zhao, Levi J Hargrove, Aaron D Ames, Eric J Perreault, and Timothy Bretl. Learning impedance controller parameters for lower-limb prostheses.

[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] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, 2005.

[10] Neville Hogan. Impedance control: An approach to manipulation. pages 304–313, 1984.

[11] Philip Holmes, Robert J Full, Dan Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *Siam Review*, 48(2):207–304, 2006.

[12] F. Iida and R. Tedrake. Minimalistic control of biped walking in rough terrain. *Autonomous Robots*, 28:355–365.

[13] National Instrument. Getting started with the ni 9502 brushless servo drive module and ni akm motors.

[14] 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.

[15] 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.

[16] S. Rutishause, 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.

[17] R.W. Sinnet 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] Mark W. Spong. and Francesco 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] Frank Sup, Amit Bohara, and Michael 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.