FLAT-FOOT DYNAMIC WALKING VIA HUMAN-INSPIRED CONTROLLER DESIGN

A Thesis

by

WENLONG MA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Aaron D. Ames
Committee Members,	Bryan P. Rasmussen
	John Hurtado
Head of Department,	Andreas Polycarpou

May 2014

Major Subject: Mechanical Engineering

ABSTRACT

This thesis 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 walking gait that can be implemented on the robot, which is represented in the form of canonical walking functions. 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 of a successful walking with AMBER2. Finally, the unified PD, impedance torque control law is experimentally realized on the bipedal robot AMBER2. Through the evidence of sustainable and unsupported walking achieved on AMBER2 showing high consistency with the simulated walking gait, the feasibility of AMBER2 walking scheme will be verified.

To my grandfather,

Zhan-Gen Zhang,

for leaving me the best strength to realize my dream.

May you peace in another world.

ACKNOWLEDGEMENTS

I would like to include my greatest gratitude to my advisor —Dr. Aaron Ames, for giving me this fantastic opportunity to realize my dream and inspiring me with his wisdom, passion and patience. I also appreciate my treasure friends and teammates Hui-Hua, Shishir, Michael, Ryan and John for being part of team AMBER2 and sharing abundant and valuable ideas and insights with me that finally leads this project to a great success. Thank all the other lab-mates, Ayongga, Eric, Jordan, Matthew, Shao-Chen, James, Shawanee, and Murali for keeping AMBER lab such a wonderful place to pursue science and knowledge. I would also extend my sincere gratitude to my advisory committee Dr. Hurtado and Dr. Rasmussen for their keen inputs in making this work a valuable literature. Particularly, I would very much thank Dr. Rasmussen, who initially led me the way to the world of control theory in fall2011 when I was a beginner and extremely confused by this unknown world. Without his encouragement and patience I cannot image how could I ever achieve this far. My thesis work is supported by NSF grants CPS-1239085, CNS-1136104 and CNS-0953823.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION*

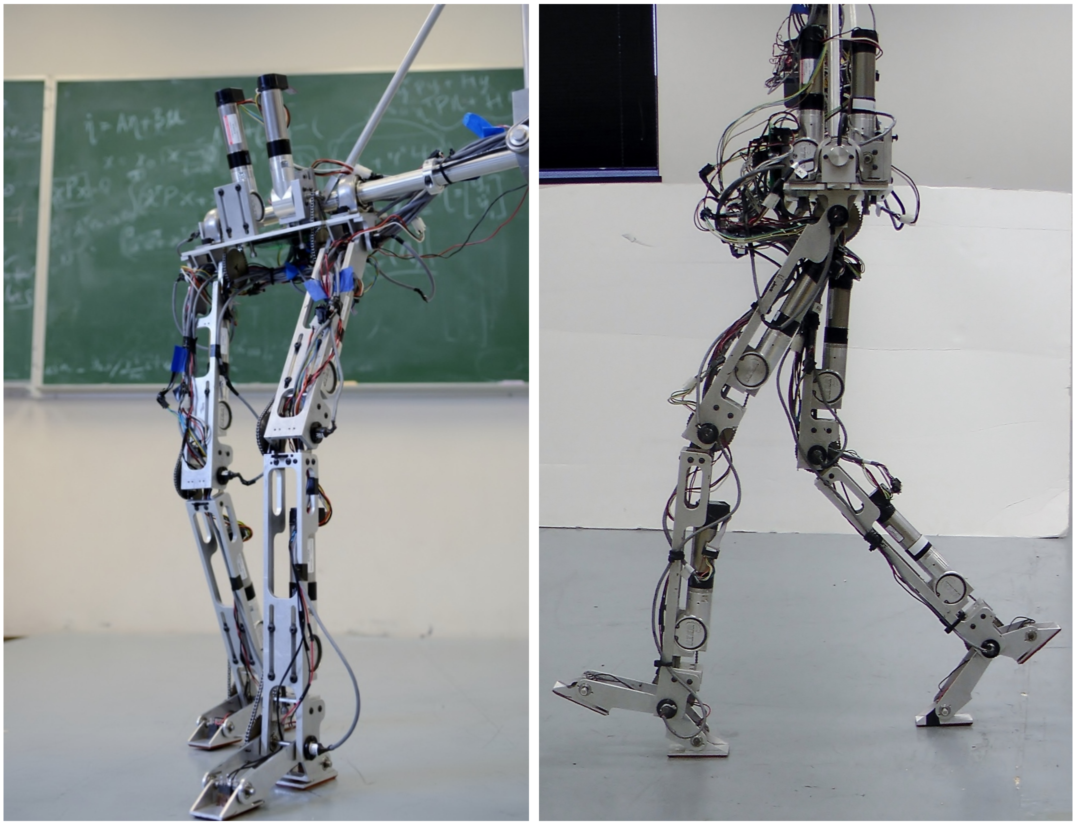


Figure 1.1: The bipedal robot AMBER2

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 [20], geometric reduction [6, 19], inverted pendulum [17, 13]) that are provably correct, and simulation/experimental results guided

*Portions of this thesis have been reprinted with permission from "Human-inspired Walking via Unified PD and Impedance Control" by W. Ma, H. Zhao, S. Kolathaya and A. D. Ames, 2014. IEEE International Conference on Robotics and Automation (ICRA 2014). Copyright 2014 by IEEE.

by heuristics (e.g., ZMP methods [23, 24], passivity based control [10, 14], reinforcement learning [15] and the central pattern generators [18]) 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 [9, 21, 25], novel methods are still needed to unify theoretical results with experimental realization.

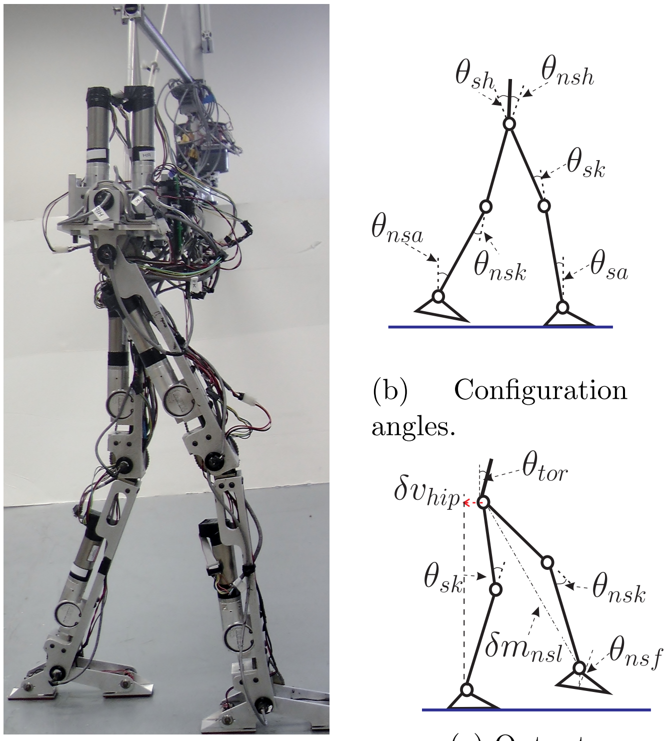
With the goal of making the first step toward bridging the gap between theoretical simulation and experimental implementation, this thesis focuses on flat-feet robotic walking with a planar bipedal robot with feet—AMBER2 (Fig. 1.1) using the human-inspired optimization and controller design. To introduce the control framework, this thesis starts with introducing a high fidelity model of AMBER2 in Sect. 2. 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 the 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, subject to certain physical constraints (Sect. 3) that provably guarantee robotic walking, is developed (this method has been successfully applied to other bipedal robots including NAO [5] and AMBER [26] as well), the end results are the parameters for canonical walking functions [3] that produce human-like trajectories that are amendable to implementation on 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 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 4, the utilization of a novel impedance control scheme can avoid the shortages while keeping the benefits.

The main contribution of this research is the experimental implementation of the 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 Sect5, the framework of the AMBER2 walking strategy was verified in both simulation and experiment. 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.

2. AMBER2 FLAT-FOOT WALKING MODEL

AMBER2 is a 2D bipedal robot with seven links (two calves, two thighs, two feet and a torso, see Fig. 2.1a). AMBER2 is a second generation bipedal robot and an expansion upon its predecessor, the non-footed (point foot) bipedal robot, AMBER (see [26]). Each joint is actuated by a brushless DC (BLDC) motor. In addition, with motion being restricted to the sagittal plane via a boom shown in Fig. 2.2, which are configured as parallel four-link mechanism, the boom support structure (4) in Fig. 2.2 is always horizontal. The boom is fixed rigidly to a rotating mechanism (see Fig. 2.2), which allows the biped to walk in a circle with minimal friction. In



(a) Sideview of AMBER2.

(b) Configuration angles.

(c) Outputs.

Figure 2.1: (a) The bipedal robot AMBER2, (b) Robot joint angles, (c) Robot outputs.

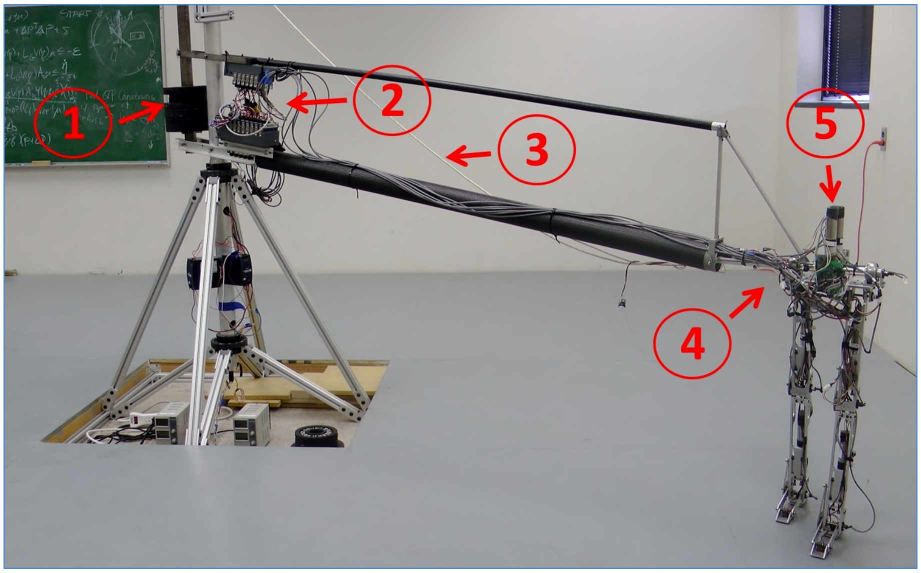


Figure 2.2: AMBER2 with the boom and electronics. The boom restricts motion to the sagittal plane. As shown in the figure: (1) Counterweight used to balance the boom around the pivot, (2) Controller module where the walking algorithm is running, (3) The boom, (4) Boom support structure which keeps the torso horizontal, (5) The bipedal robot AMBER2.

addition, counterweights are provided (see Fig. 2.2) to cancel out the weight on the robot due to the boom weight. 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. 2.1b). These variables form the configuration space of the robot, and are shown in Fig. 2.1b. L_c, L_t are the lengths of the calf and thigh respectively (values are given in Table. 2.1).

2.1 Continuous Dynamics

Given the configuration $\theta = (\theta_{sa}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa}\ )^T \in Q$, and computing the mass and inertia properties of each link of the robot through a SolidWorks model allows for the construction of the Lagrangian:

$$L(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T D(\theta)\dot{\theta} - V(\theta) \tag{2.1}$$

Explicitly, this is done symbolically through the method of exponential twists [16] in Wolfram Mathematica.

The AMBER2 model also contains the motors and boom. The way the inertia of these two elements are included in the model is slightly different. This approach was first considered in [26], and will be revisited here. Let I_r be the rotational inertia of the rotor and I_g be the rotational inertia of the gearbox. Due to the large gear ratio, I_g is small enough to be 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

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

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 (along z-axis) and forward-backward (along x-axis), the boom exhibits yaw and roll about the pivot. This would correspond to the x component and z component of the velocities of the torso. The center of mass 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, $M_{boom} \in \mathbb{R}^{6\times 6}$, 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. Then the new combined mass inertia matrix, D_{com} used in the Lagrangian will be:

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

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, the Euler-Lagrange equations can be realized in the following manner:

$$D_{com}(\theta)\ddot{\theta} + H(\theta, \dot{\theta}) = B(\theta)u, \qquad (2.2)$$

where $u \in R^6$ 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

$$\dot{x} = f(x) + g(x)u, \qquad (2.3)$$

where $x = (\theta, \dot{\theta})$ and,

$$f(x) = \begin{bmatrix} \dot{\theta} \\ -D^{-1}(\theta)H(\theta, \dot{\theta}) \end{bmatrix} \qquad g(x) = \begin{bmatrix} 0 \\ D^{-1}(\theta)B(\theta) \end{bmatrix}$$

2.2 Discrete Dynamics

The domain, X, describes the allowable configuration of the system restricted by the guard function h_{nsf} is given by

$$X = \left\{ (\theta, \dot{\theta}) \in TQ : h_{nsf}(\theta) \geq 0 \right\},$$

where, TQ is the tangent space of the configuration space Q. The guard function specifies that the non-stance foot must be above the ground, i.e., the height of non-stance foot, $h_{nsf} \geq 0$.

The guard is just the boundary of the domain with the additional assumption that h_{nsf} is decreasing:

$$S = \left\{ (\theta, \dot{\theta}) \in TQ : h_{nsf}(\theta) = 0 \text{ and } \frac{\partial h(\theta)}{\partial \theta}\, \dot{\theta} < 0 \right\}.$$

When the non-stance foot impacts the ground, the angular velocities will change instantaneously while the joint angle won't change. Hence we define a reset map which calculates the post impact velocities in terms of the pre-impact velocities. In addition, the "stance" and "non-stance" legs are swapped for simplicity. The reset map Δ is then given by:

$$\Delta(\theta, \dot{\theta}) = \begin{bmatrix} \Delta_{\theta}\theta \\ \Delta_{\dot{\theta}}(\theta)\dot{\theta} \end{bmatrix}, \tag{2.4}$$

where Δ_{θ} is the relabeling matrix which switches the stance and non-stance legs at impact (by appropriately changing the angles). Here, $\Delta_{\dot{\theta}}$ determines the change in velocity due to impact. This thesis forgos the detailed discussion on its computation, but more descriptions can be found in [11] and [3].

3. 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.

3.1 Human-Inspired Outputs

To achieve human-like walking, we begin with seeking "outputs" of the human locomotion data, more detail about the human walking experimental research can be found in [3]. For this thesis, as it was denoted in Fig. 2.1b, six outputs are considered for the 6-DOF robot:

1. The linearized x-position of the hip:

$$\delta p_{\text{hip}}(\theta) = L_c(-\theta_{sf}) + L_t(-\theta_{sf} - \theta_{sk}), \tag{3.1}$$

2. The angle of stance knee θ_{sk},

3. The angle of non-stance knee θ_{nsk},

4. The linearized slope of the non-stance leg m_{nsl}: the tangent of the angle between the z-axis and the line on the non-stance leg connecting the ankle and hip, given by:

$$m_{nsl} = -\theta_{sf} - \theta_{sk} - \theta_{sh} + \theta_{nsh} + \frac{L_c}{L_c + L_t}\theta_{nsk}, \tag{3.2}$$

5. The torso angle from vertical:

$$\theta_{tor}(\theta) = \theta_{sa} + \theta_{sk} + \theta_{sh}, \tag{3.3}$$

6. The angle of the non-stance foot w.r.t the horizontal:

$$\theta_{nsf} = \theta_{sa} + \theta_{sk} + \theta_{sh} - \theta_{nsh} - \theta_{nsk} - \theta_{nsa}. \tag{3.4}$$

Analysis of the chosen outputs data indicates that, the linearized hip position is a linear function of time:

$$\delta p^d_{\text{hip}}(t, v) = v_{hip} t, \tag{3.5}$$

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):

$$y_H(t, \alpha) = e^{-\alpha_1 t}(\alpha_2 \cos(\alpha_3 t) + \alpha_4 \sin(\alpha_3 t)) + \alpha_5, \tag{3.6}$$

where $\alpha_1 = c_0$, $\alpha_3 = c_1$ are determined by the initial condition of the system, $\alpha_2 = \omega_d$ with $\omega_d = \omega_n \sqrt{1 - \xi^2}$ the damped natural frequency, ξ is the damping ratio, $\alpha_4 = \xi \omega_n$, with ω_n the natural frequency, and $\alpha_5 = g$. Based on the linear fashion of the linearized hip position, we parameterized the time as:

$$\tau(\theta) = \frac{\delta p^R_{\text{hip}}(\theta) - \delta p^R_{\text{hip}}(\theta^+)}{v_{\text{hip}}}, \tag{3.7}$$

which removes the dependence of time in (3.6) and renders an autonomous system [25]. 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:

$$y_\alpha(\theta,\dot{\theta}) = \begin{bmatrix} y_1(\theta,\dot{\theta}) \\ y_2(\theta) \end{bmatrix} = \begin{bmatrix} y_{a,1}(\theta,\dot{\theta}) - v_{hip} \\ y_{a,2}(\theta) - y_{d,2}(\tau(\theta),\alpha) \end{bmatrix}, \tag{3.8}$$

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:

$$y_{d,2}(t,\alpha) = \begin{bmatrix} y_H(t,\alpha_{sk}) \\ y_H(t,\alpha_{nsk}) \\ y_H(t,\alpha_{nsl}) \\ y_H(t,\alpha_{tor}) \\ y_H(t,\alpha_{nsf}) \end{bmatrix}, \quad y_{a,2}(\theta) = \begin{bmatrix} \theta_{sk} \\ \theta_{nsk} \\ \delta m_{nsl}(\theta) \\ \theta_{tor}(\theta) \\ \theta_{nsf} \end{bmatrix}, \tag{3.9}$$

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 $y_{a,2}(\theta) = H\theta$.

3.2 Control Law Construction

To drive $y_1 \to 0$ and $y_2 \to 0$, this thesis utilized input/output feedback linearization controller for simulations. In other words, the goal is to drive the dynamics of the system to the *zero dynamic surface*:

$$\mathbf{Z}_\alpha = (\theta,\dot{\theta}) \in \mathbf{TQ} : \mathbf{y_1}(\theta,\dot{\theta}) = \mathbf{0}, \mathbf{y_2}(\theta) = \mathbf{0}, \dot{\mathbf{y}}_\mathbf{2}(\theta,\dot{\theta}) = \mathbf{0})$$

We now differentiate the relative degree 1 once and the relative degree 2 twice, yields:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} L_f y_1(\theta, \dot{\theta}) \\ L_f^2 y_2(\theta) \end{bmatrix}}_{L} + \underbrace{\begin{bmatrix} L_g y_1(\dot{\theta}) \\ L_g L_f y_2(\theta) \end{bmatrix}}_{A} u \tag{3.10}$$

with L the Lie derivative, is nonsingular and A the decoupling matrix. Therefore, we can define the following torque controller:

$$u = -A^{-1}(\theta, \dot{\theta})(\begin{bmatrix} 0 \\ L_f^2 y_1(\theta, \dot{\theta}) \end{bmatrix} + \begin{bmatrix} L_f y_1(\theta, \dot{\theta}) \\ 2\varepsilon L_f y_2(\theta) \end{bmatrix} + \begin{bmatrix} \varepsilon y_1(\theta, \dot{\theta}) \\ \varepsilon^2 y_2(\theta) \end{bmatrix}).$$

In other words, we can apply feedback linearization to obtain a linear system on the human-inspired output. This system is exponentially stable, implying that for $\varepsilon > 0$ the control law u drives $y_1 \to 0$ and $y_2 \to 0$ as $t \to \infty$. Applying the feedback control law in (3.11) to the hybrid control system $\mathscr{HC} = (X, u, S, \Delta, f, g)$, yields a hybrid system:

$$\mathscr{H}^{(\alpha,\varepsilon)} = (X, S, \Delta, f^{(\alpha,\varepsilon)}), \tag{3.11}$$

where, X, S, and Δ are defined as for $\mathscr{HC}$, and

$$f^{(\alpha,\varepsilon)}(\theta, \dot{\theta}) = f(\theta, \dot{\theta}) + g_v(\theta, \dot{\theta})u.$$

Before the specific method for determining the parameters α that result in robotic walking is presented, we need to introduce *Partial Hybrid Zero Dynamics* first. 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*:

$$\mathbf{PZ}_\alpha = \{(\theta, \dot{\theta}) \in TQ : y_2(\theta) = \mathbf{0},\ L_f y_2(\theta) = \mathbf{0}\}. \tag{3.12}$$

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 the 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.

3.3 Human-Inspired Optimization

Having constructed the human-inspired controller, the objective of this section is to find the controller parameters, α, which deliver provable walking with the robot. However, in order to implement the results in the robot, physically realizable constraints need to be imposed. We will begin with describing each constraint explicitly, and then present the optimization problem.

Partial Hybrid Zero Dynamics Constraints. As discussed in former section, in order to realize exponentially stable orbits in hybrid systems, PHZD constraints need to be imposed on the objective function:

$$y_{2,\alpha}(\vartheta(\alpha)) = \mathbf{0}, \tag{C1}$$

$$dy_{2,\alpha}(\Delta_\theta \vartheta(\alpha)), \Delta_{\dot{\theta}}(\vartheta(\alpha))\dot{\vartheta}(\alpha) = \mathbf{0}, \tag{C2}$$

$$dh_{nsf}(\vartheta(\alpha))\dot{\vartheta}(\alpha) < 0, \tag{C3}$$

where $\vartheta(\alpha)$, $\dot{\vartheta}(\alpha)$ are the configuration and velocities respectively of the biped at the beginning of the step, and can be computed purely from the parameters, α

(More on the computation of $\vartheta(\alpha)$ can be found in [4].) (C1) and (C2) enforce the PHZD constraints, and (C3) ensures that the non-stance foot strikes the guard “transversally”. The end results of these constraints, therefore, is PHZD [3].

Torque Constraints. Torques acting on the joints are limited by the capacity of the motors and the modules. Therefore, constraints were imposed on not just the maximum torque from the linearizing controller, but also the change in torque demands due to impacts. In fact, achieving steady state walking was stymied by the frequent occurrence of current spikes in the non-stance hip and non-stance knee motors during impacts. Therefore, besides minimizing maximum allowable torques, constraints were imposed to ensure smooth torque demands in the non-stance knee and non-stance hip actuators.

$$\max_{0 \leq u(\theta(\alpha))} \|u(\theta^e(t,\alpha), \dot{\theta}^e(t,\alpha))\| < Torque_{MAX} \tag{C4}$$

$$u_{nsh}(\vartheta(\alpha), \dot{\vartheta}(\alpha), \alpha) = u_{nsh}(\Delta_\theta \vartheta(\alpha), \Delta_{\dot{\theta}}(\vartheta(\alpha)), \alpha) \tag{C5}$$

$$u_{nsk}(\vartheta(\alpha), \dot{\vartheta}(\alpha), \alpha) = u_{nsk}(\Delta_\theta \vartheta(\alpha)), \Delta_{\dot{\theta}}(\vartheta(\alpha)), \alpha) \tag{C6}$$

where $Torque_{MAX}$ is the maximum torque the motor can supply with the efficiency consideration and u are the actual torques in the simulation.

Foot Scuffing Conditions. There needs to be sufficient swing foot height clearance and stride length during the swing phase of walking to prevent scuffing. The clearance must be sufficient enough to avoid scuffing amidst sensor noise, tracking error, uneven ground and even imperfection in the mechanical design. Therefore, the following constraint is proposed:

$$\max_{0\leq\tau(\theta(\alpha))}(h_{nsf}(\theta^e(t,\alpha)) - h_{quad}(\theta^e(t,\alpha), hmax)) > 0 \tag{C7}$$

$$\max_{0\leq\tau(\theta(\alpha))} l_{nsf}(\theta^e(t,\alpha)) - MIN_{steplength} > 0, \tag{C8}$$

where h_{quad} is a quadratic polynomial above which the non-stance foot (h_{nsf}) must remain over the course of a step. h_{quad} can be computed from the parameters α and the user specified peak height of the quadratic ($hmax$). The stride length l_{nsf} is constrained to be above a minimum specified stride length, $MIN_{steplength}$. Simultaneously, the strike length is also optimized to guarantee better walking gait.

Human-Inspired Optimization. Having specified all the constraints, we now consider the human-inspired optimization problem with using the human-data-based cost function,

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

$$\text{s.t} \quad \text{(C1)} - \text{(C8)} \tag{C}$$

The cost function (HIO) is the least squares fit between the human experimental data and the CWF representations [5], i.e., we try to find the α parameters that best fit human-walking data which enforcing the desired constraints. The end result of this optimization is that the control law (3.11) results in provable stable robotic walking for the hybrid system model of AMBER2 (3.11) (see [5] for a proof which easily extends to the case of AMBER2).

4. CONTRLLER DESIGN

Having constructed the human-inspired trajectory from optimization, the objective of this section is to design the appropriate controller that 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 unified PD impedance controller.

4.1 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:

$$\begin{aligned} \xi_1 &= \delta p_{hip}^R(\theta) := c\theta \\ \xi_2 &= y_1^a(\theta, \dot{\theta}) := \delta \dot{p}_{hip}^R(\theta) := c\dot{\theta} \end{aligned} \tag{4.1}$$

where c is obtained from (3.7). Since ξ_1 is the linearized position of the hip, which is used to parameterize time as (3.7), 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:

$$\begin{aligned} \eta_1 &= y_{2,a} = H\theta \\ \eta_2 &= L_{f^R} y_{2,a}(\theta, \dot{\theta}) = H\dot{\theta} \end{aligned} \tag{4.2}$$

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 joints angles and velocities and the desired outputs of the robot:

$$\begin{aligned}\theta_d(\tau) = \Psi(\xi_1, \alpha) &= \begin{bmatrix} c \\ H \end{bmatrix}^{-1} \begin{pmatrix} \xi_1 \\ y_{d,2}(\xi_1, \alpha) \end{pmatrix} \\ \dot{\theta}_d(\tau) = \Phi(\xi_1, \xi_2, \alpha) &= \begin{bmatrix} c \\ H \end{bmatrix}^{-1} \begin{pmatrix} v_{hip} \\ \frac{\partial y_{d,2}(\xi_1,\alpha)}{\partial \xi_1}\xi_2 \end{pmatrix}\end{aligned} \tag{4.3}$$

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:

$$\begin{aligned}\dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= -\varepsilon(\xi_2 - v_{hip})\end{aligned} \tag{4.4}$$

Having known ξ_1, ξ_2, the desired angles and velocities are obtained from (4.3). 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 constraints of the dynamics to the partial zero dynamics surface still maintain.

4.2 Feedback PD Control

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

$$u^f_{PD} = K_p(\theta_a - \theta_d) + K_d(\dot{\theta}_a - \dot{\theta}_d) \tag{4.5}$$

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

4.3 Feed-forward Impedance Control

Impedance control benefited from its simplicity and passive nature, is one of the 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 [12], the torque at each joint during a single step can be represented in a piecewise fashion by a series of passive impedance functions [22] in the following form:

$$u = k(\theta - \theta^e) + b\dot{\theta}. \tag{4.6}$$

Inspired by the previous work [2], analysis of AMBER2 experimental walking data (which is achieved by PD controller alone) shows that one gait cycle can be divided into two phases based on the knee joints, which are denoted as $p = 1, 2$. Specifically, each phase begins at time t_0^p and ends at t_f^p. The phase separation principle is similar to that in [2], but with values specific to the gait of AMBER2. The impedance torque for specific joint i during a phase $p \in \{1, 2\}$, can be represented by the following equation:

$$u_{i,p}^f = k_{i,p}(\theta_i(t) - \theta_{i,p}^e) + b_{i,p}\dot{\theta}_i(t), \tag{4.7}$$

where $\theta_i(t)$ and $\dot{\theta}_i(t)$ denote the 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. From the work in [2], it was shown 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 an experiment with a transfemoral prosthetic device. To extend these results to AMBER2, we utilize a similar method to estimate the impedance parameters by analysis the data of the best 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 $u^a_{i,p}$ obtained by utilizing the PD controller on AMBER2 in experiment, we can form the least square errors minimization problem as following:

$$\beta^*_{i,p} = \underset{\beta_{i,p}}{\operatorname{argmin}} \int_{t^p_0}^{t^p_f} (\tau^f_{i,p} - \tau^a_{i,p})^2 dt, \tag{4.8}$$

where $u^f_{i,p}$ is defined as (4.7) 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.

4.4 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:

$$u^f = u^f_{PD}(\theta_a, \theta_d, \dot{\theta}_a, \dot{\theta}_d, K_p, K_d) + u^f_{i,p}(\theta_a, \dot{\theta}_a, k, b, q_e) \tag{4.9}$$

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 does not cause significant distortions in tracking.

5. EXPERIMENTAL REALIZATION OF FLAT-FOOT WALKING

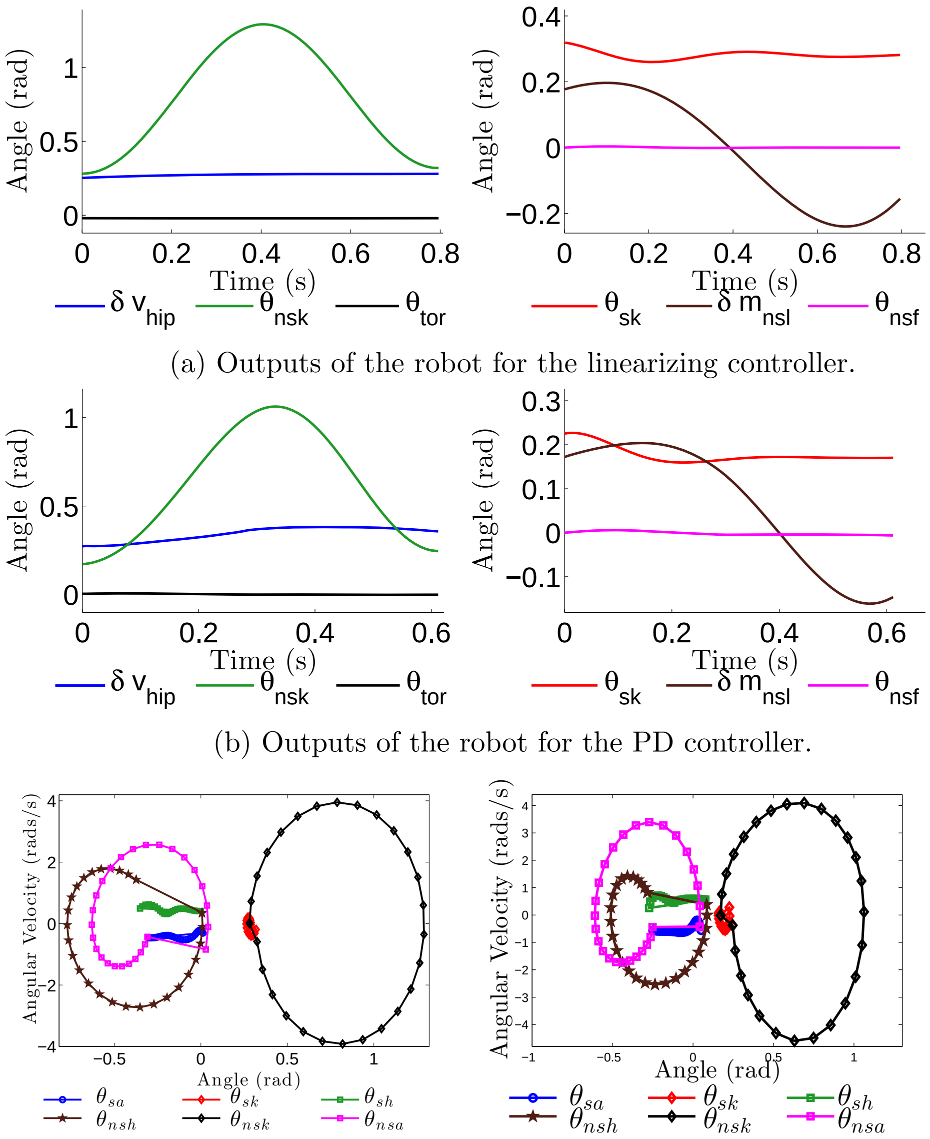
To realize real world walking on the physical robot AMBER2, we use LabView2011 as our IDE (Integrated Development Environment) to develop the code and control the robot. The controller for AMBER2 has two levels: high level controller, which is realized by RT (Real Time) control, and low level controller realized by FPGA (Field-Programmable Gate Array). The objective of this section is to introduce the control structure of AMBER2.

5.1 High Level Controller

The Real Time control has the following major functionality incorporated:

- Interface with the FPGA, read joint angles and angular velocities, send torque commands to low level controller, enable/disable motors, logging data.
- Compute the time parameter τ using (3.7).
- Compute torque command based on $u_d, \theta_d, \dot{\theta}_d, \theta_a, \dot{\theta}_a$. Then by applying PD with impedance control law to corresponding motors, the desired torque input u_{in}^f is fed into the FPGA.

Note that for AMBER2, the sample rate and command rate are both 143Hz. Due to the complexity of the controller itself, labview build-in code is not efficient enough to control the robot. Instead, the high level controller is coded into customized shared libraries to interface with C++. The NI9144 EhterCAT Slave chassis is connected to the cRIO by EtherCAT to extend the capacity of compactRIO. For this configuration, each chassis is in charge of one leg. The pseudo-code running in RT is shown in Algorithm 1 and the Labview code is shown in 5.2.



(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).

Figure 5.1: Simulation results for the feedback linearizing and PD voltage controllers.

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: $\theta_{abs}$
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;
Enable Motor Drives;
repeat
    Wait till all motor drives are Enabled
until ( Drive-Status == Enable )
while ( $\neg$ Stop-RT ) do
    Reform $\theta, \dot{\theta}$ from Left/Right($\theta_{LR}$) to Stance/nonStance($\theta_{SnS}$);
    Calculate actual time parameter $\tau_a$ ;
    Desired $\tau_d = \tau_a + \delta T$;
    Calculate( $\xi_1$, $\xi_2$);
    Calculate( $y_d$, $\dot{y}_d$ );
    Calculate( $\theta_d$, $\dot{\theta}_d$ );
    Apply PD Controller:
      $u^f_{PD} = K_p(\theta_a - \theta_d) + K_d(\dot{\theta}_a - \dot{\theta}_d)$;
    Based on $\tau_a$ and $\tau_{flag}$, determine $Impedance\ Phase$;
    Apply Impedance Controller:
      $u^f_{imp} = Impedance(\theta,\ \dot{\theta},\ K,\ \theta_e,\ b_i)$;
    Control Law Constructed:
      $u^f = u^f_{PD} + u^f_{imp}$;
    Change $u^f$ 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;
```

5.2 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:

- *Angular velocity measurement*: Single-ended incremental quadrature encoders attached to every rotor on the motor end is used to measure angler velocity using the following equation:

$$\omega = \frac{2\pi/(n_0 r)}{T} \times n,$$

where n_0 is the total counts per round of the rotor, r is the gear ration from rotor to joint end, T is the sampling period, and n is the counts measured by the encoder.

- *Joint Angle Measurement*: Measure joint angles by using PWM absolute encoders mounted on the joint end for home position and integrating velocity data for angle increment/decrement. In particular, both absolute encoders and incremental encoders operate at 40MHz. Note that the high value of r of incremental encoder results in 5 times more precision than absolute encoder. The angle reading is done by the following manner

$$\theta = \theta_{abs} + \int_0^t \omega dt \tag{5.1}$$

with θ_{abs} provided by absolute encoder as home position of incremental encoder.

- *Stance foot detection*: There are two switches placed on each foot, one is in front and the other one is in the back. The way the stance and non-stance phase of each leg is decided is given by the state diagram shown in Fig. 5.4.

- *Hardware protection*: Execute hardware protection logic when any joint is trying to go beyond its working space, which is done by resetting torque command to zero.

- *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. 5.3, the torque is translated to the desired current in the manner of:

$$I_d = \frac{\tau^f}{K_t}.$$

Then the flux angle is computed from the hall sensor and incremental encoder data, which are initialized by auto-phasing. The flux angles are used in Park (inverse) and Clarke (inverse) transform, which are transforming the reference frames of three-phase (u, v, w) system to two-phase (direct-quadrature) system, more information can be found in [8]. Finally, by applying a PI controller on the quadrature and direct current, the desired three-phase voltage command are computed and the motors are actuated accordingly by sending corresponding PWM signal.

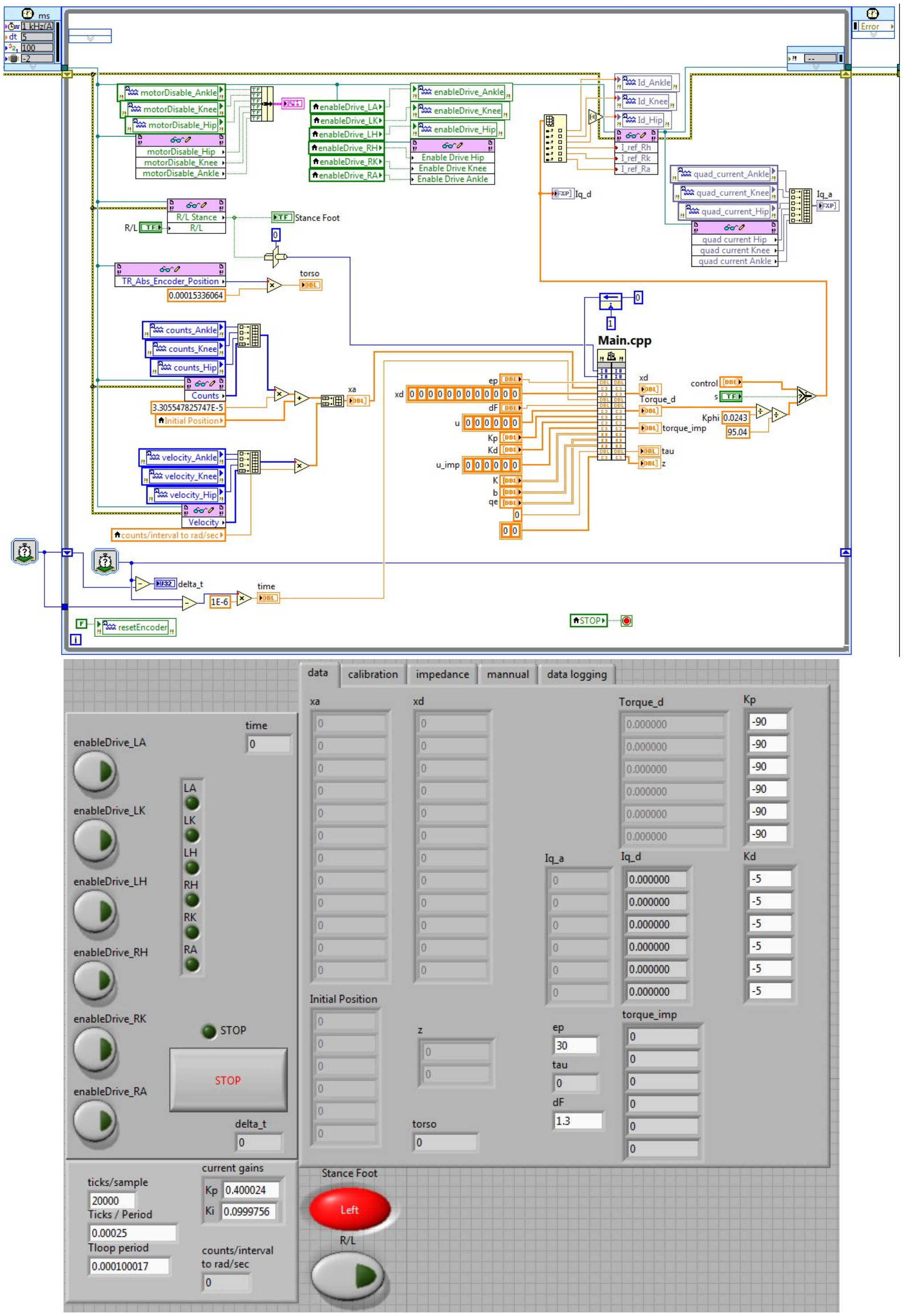


Figure 5.2: The labview code with controller coded into C++ block diagram (upper) and the front panel of it (down).

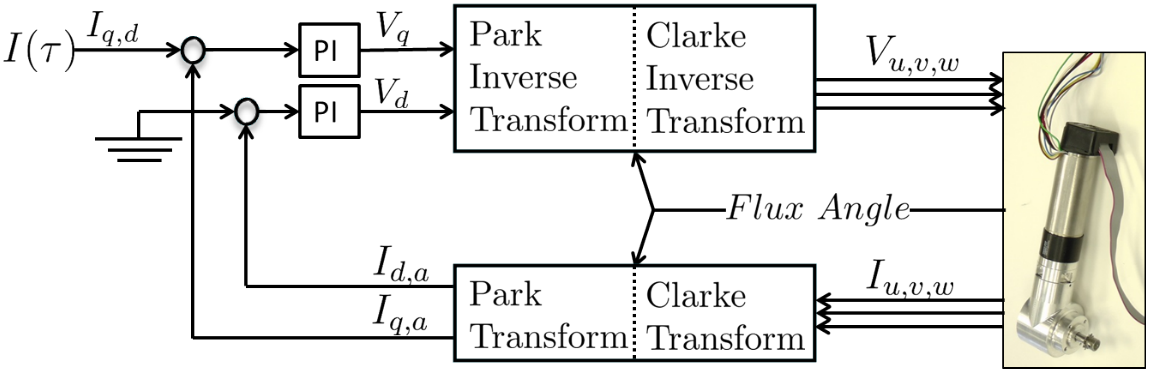


Figure 5.3: Field-oriented control block diagram

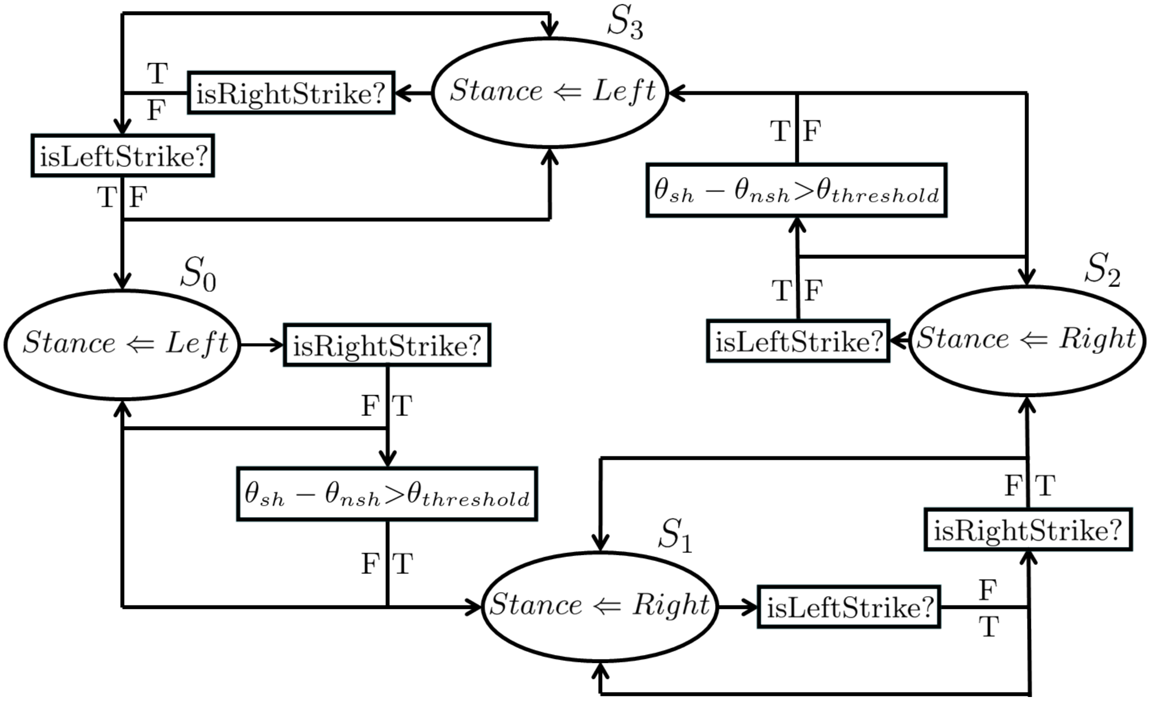


Figure 5.4: State machine showing the foot contact and the logic used to determine the stance leg.

5.3 Experimental Setup and Data Logging

In addition, other experimental setup is also included:

- *The calibration of absolute encoders.* By manually setup the zero position of absolute encoders, we can have the home position for position sensing.

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. 5.3) // Operation Frequency: 40MHz
    PWM signal Generation logic;
end loop
loop
    Guard and Stance Leg Detection Logic using foot contact switches (shown in Fig. 5.4);
    if ( Left Leg stance ) then
        L/R stance ← 0;
    else if ( Right Leg stance ) then
        L/R stance ← 1;
    end if
end loop
```

- *Auto-phasing BLDC Motors.* Determine the motor initial setup including the hall effect sensor phasing values, index angles and step angle increment of incremental encoders by turning the motor once at a very low speed.

- *Data Logging.* Remote host computer connected with AMBER2 through Ether-

CAT cable is in charge of data saving. To avoid data lost, we package data of every ten sampling period. The data packaging in the real time and unpacking in the host is shown as Fig. 5.5 .

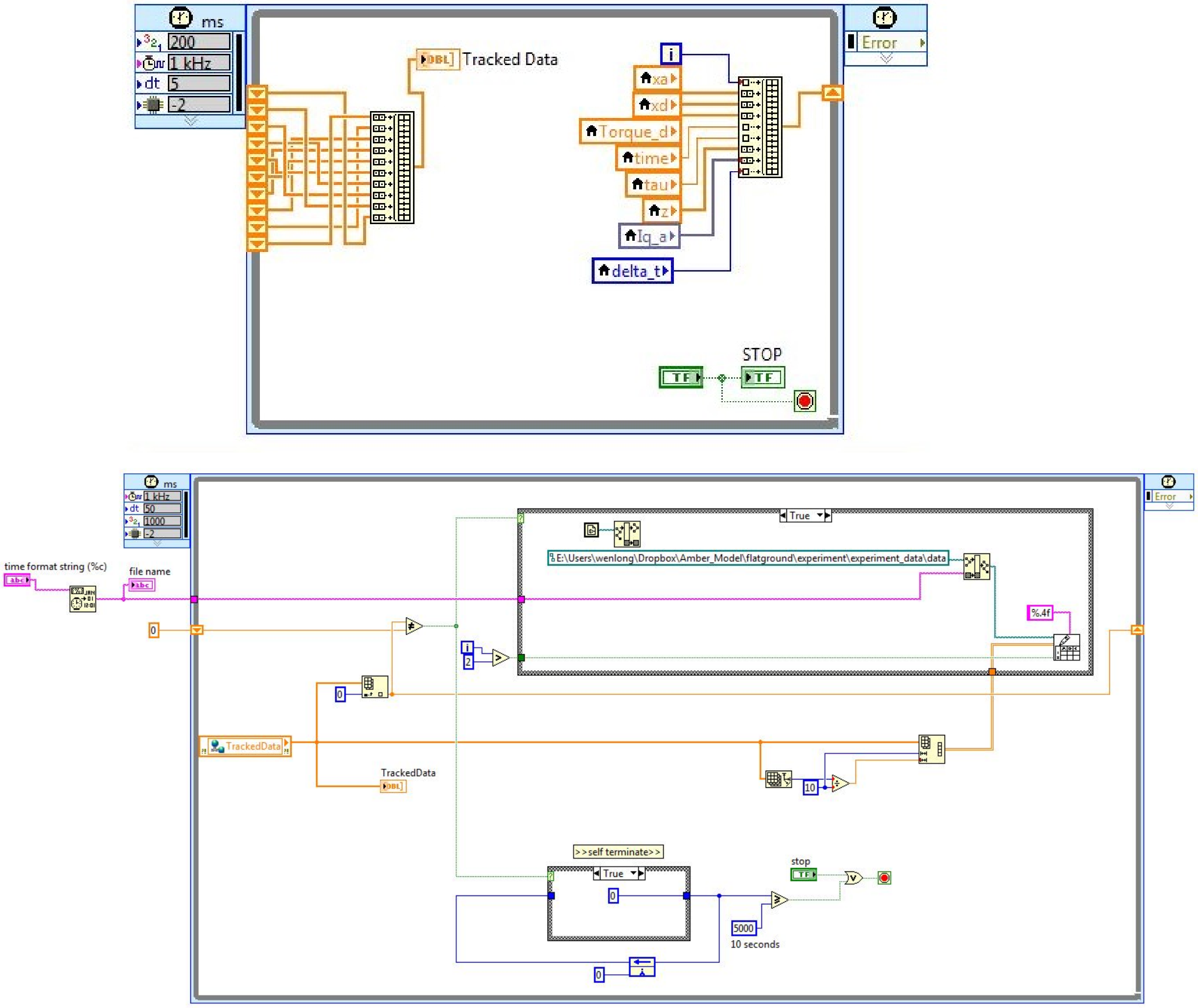


Figure 5.5: The Labview code for data packaging (upper) and data logging (down).

6. 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. 5.1a, we can see that the unified PD-impedance controller has achieved similar performance as seen in Fig. 5.1b. 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. 6.1, show good agreement between theory and simulation, and the comparison between actual and desired values of different joints are shown in Fig. 6.3 and Fig. 6.5. 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. 6.2 and Fig. 6.4. 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 produces better tracking performance but also better robustness. In fact, after impedance control was imposed, the walking is more easily repeatable and easier to start up by simply landing it on the ground and releasing it. It is very 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

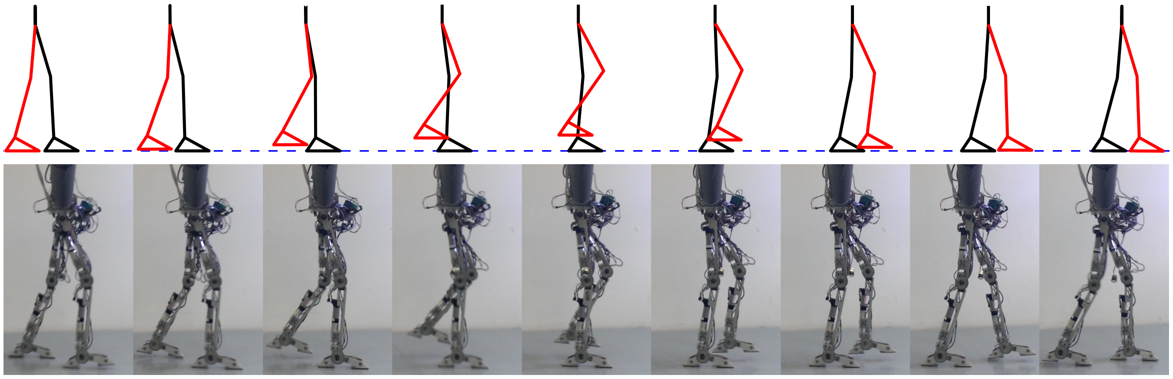
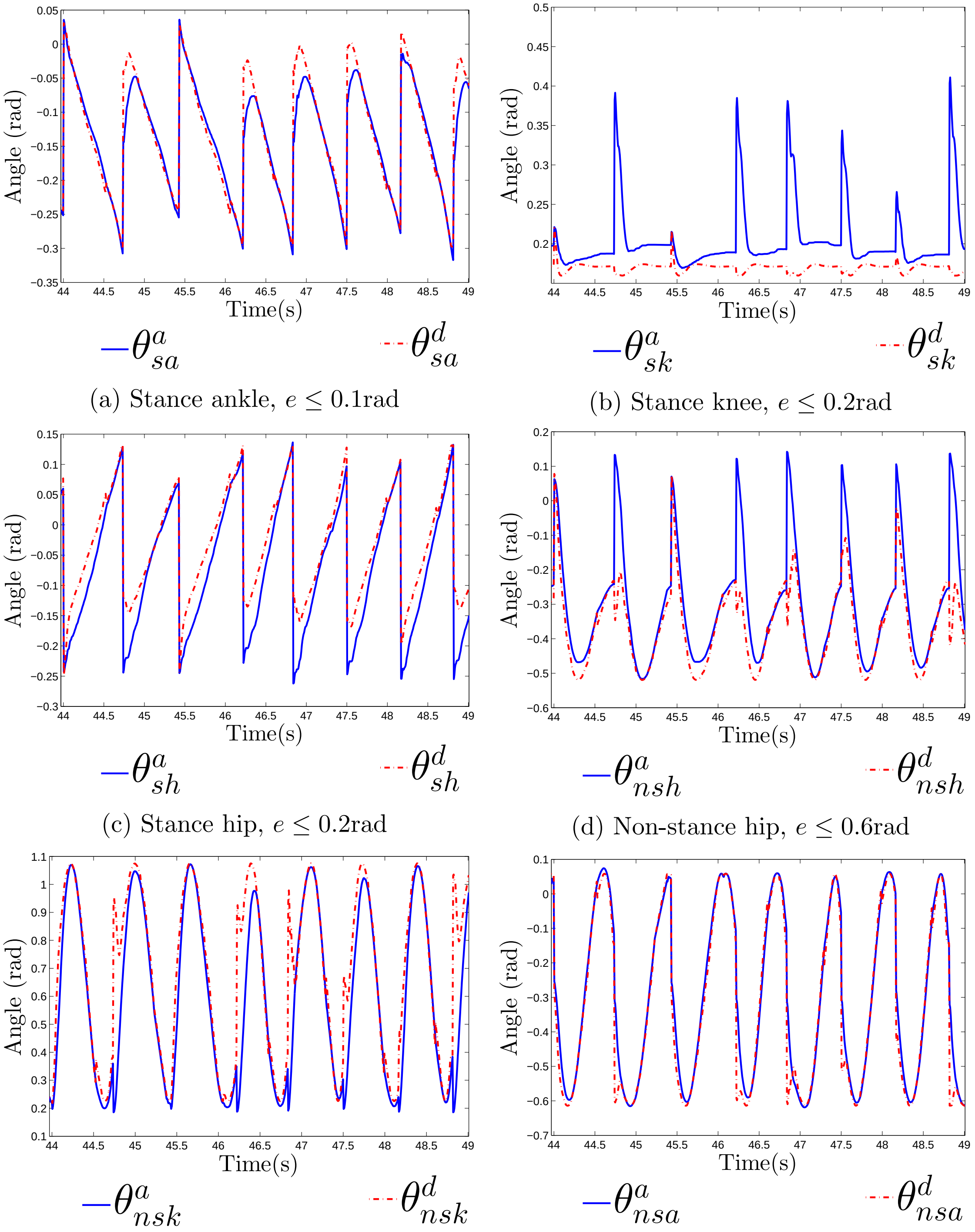


Figure 6.1: Comparison of walking tiles of simulated and experimental walking with the unified PD, impedance control.

input for ankle, knee and hip motors are 5Nm, 5Nm, 10Nm accordingly (Fig. 6.6). 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, $e \leq 0.1$rad

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

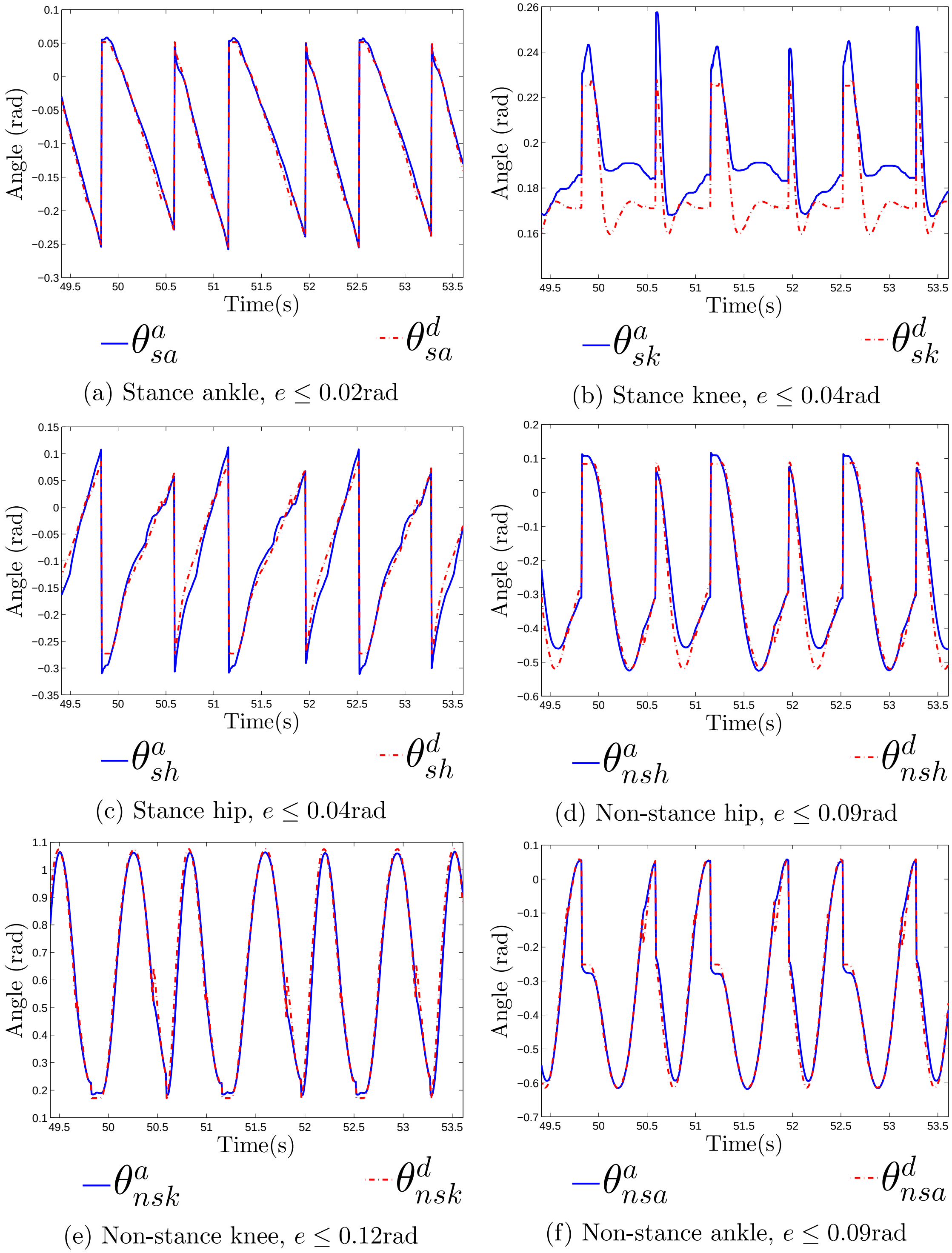
(c) Stance hip, $e \leq 0.2$rad

(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

Figure 6.2: 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

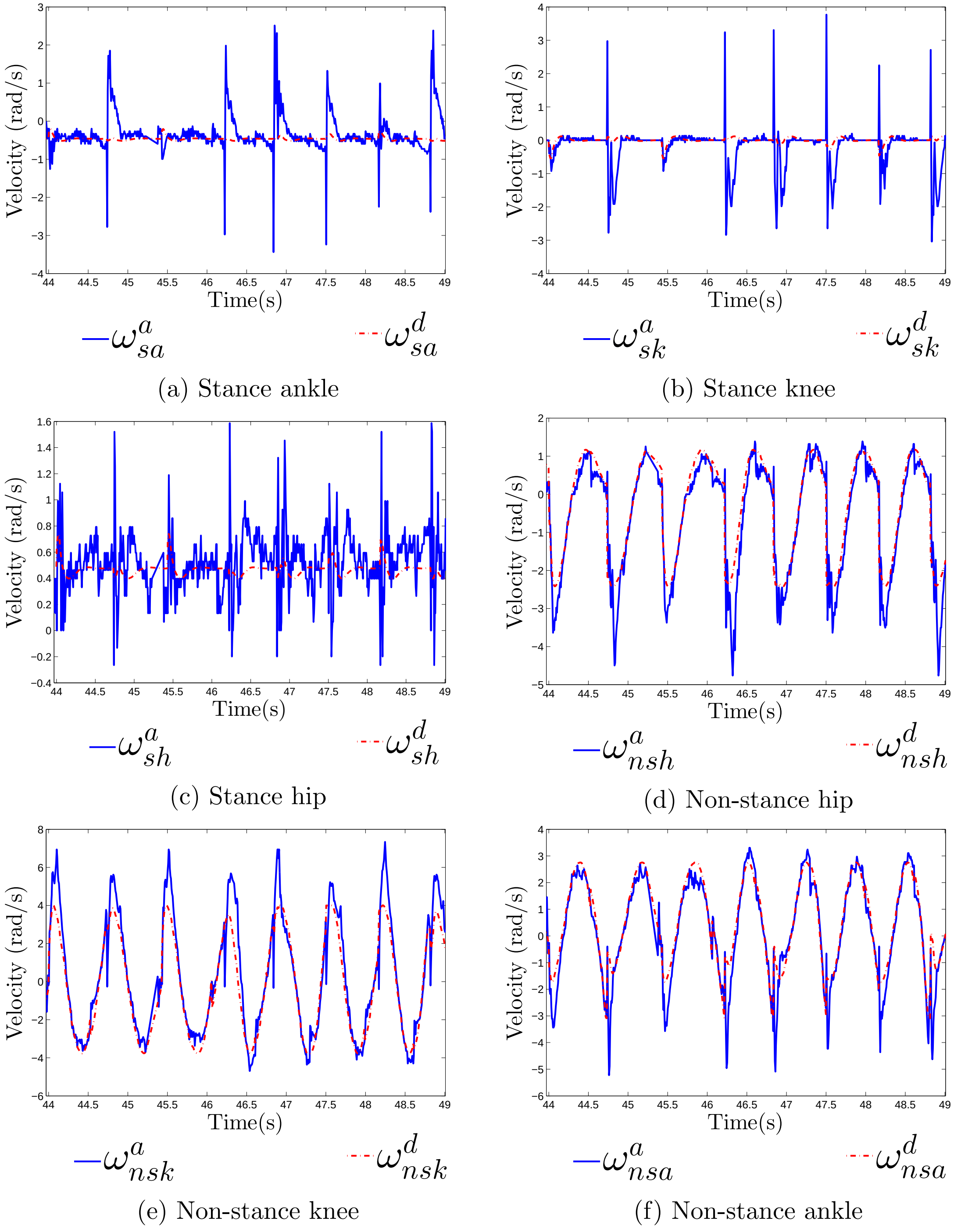
(c) Stance hip, $e \leq 0.04$rad

(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

Figure 6.3: Actual vs. desired joint angles logged during AMBER2 walking with the unified control law, with e the tracking error.



(a) Stance ankle

(b) Stance knee

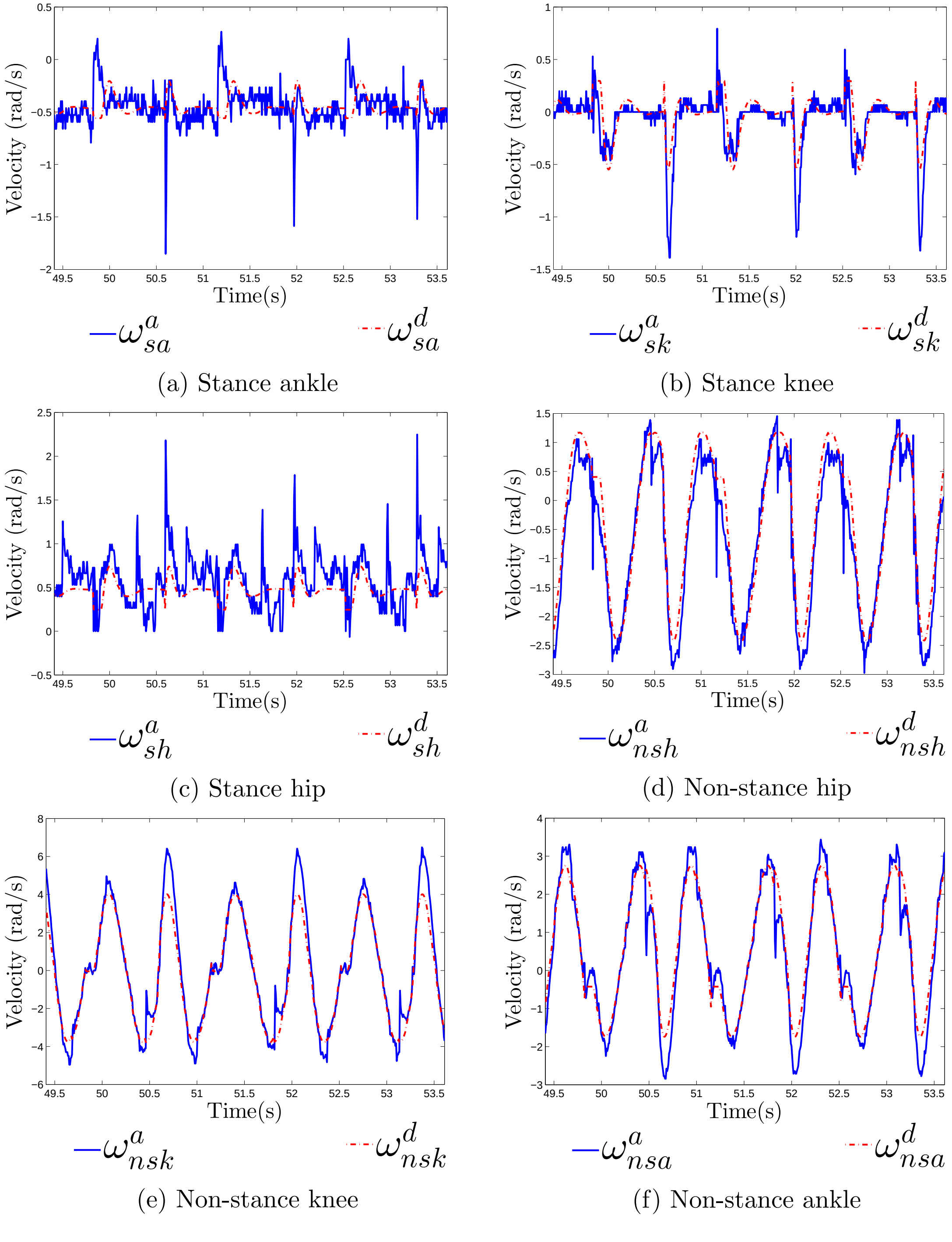
(c) Stance hip

(d) Non-stance hip

(e) Non-stance knee

(f) Non-stance ankle

Figure 6.4: Actual vs. desired joint angular velocities logged during AMBER2 walking with PD control.



(a) Stance ankle

(b) Stance knee

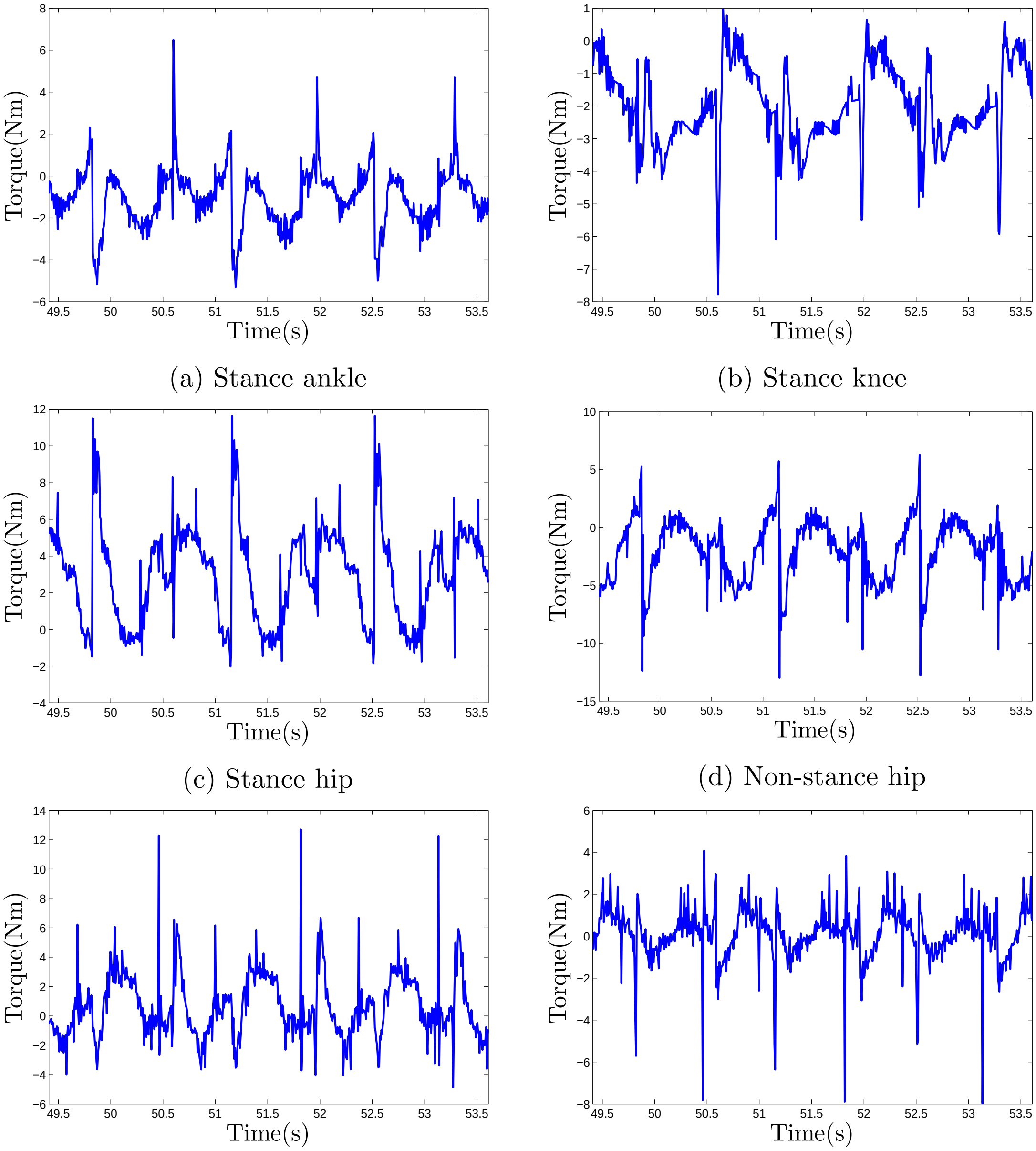
(c) Stance hip

(d) Non-stance hip

(e) Non-stance knee

(f) Non-stance ankle

Figure 6.5: Actual vs. desired joint angular velocities 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

Figure 6.6: Joint torque inputs logged during AMBER2 walking with PD, impedance control.

BIBLIOGRAPHY

[1] AMBER-Lab. Sustained Robotic Walking of AMBER2. `http://youtu.be/d6oM5sLI9vA`, 2013.

[2] N. Aghasadeghi, H. Zhao, L. J. Hargrove, A. D. Ames, E. J. Perreault, and T. Bretl. Learning Impedance Controller Parameters for Lower-Limb Prostheses. *International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[3] A. D. Ames. First Steps Toward Automatically Generating Bipedal Robotic Walking from Human Data. In *Robotic Motion and Control*, 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] C. Brosilow and B. Joseph. *Techniques of Model-Based Control*, chapter 9, Feedforward Control. Prentice Hall, 2002.

[8] S. Campbell and H. A . Toliyat. *DSP-Based Electromechanical Motion Control.* CRC Press, 2003.

[9] 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, June 2005.

[10] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient Bipedal Robots Based on Passive-Dynamic Walkers. *Science*, 307:1082–1085, 2005.

[11] 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.

[12] N. Hogan. Impedance Control: An Approach to Manipulation. *American Control Conference, 1984*, pages 304–313, 1984.

[13] 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.

[14] F. Iida and R. Tedrake. Minimalistic Control of Biped Walking in Rough Terrain. *Autonomous Robots*, 28:355–355, 2009.

[15] 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.

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

[17] 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.

[18] S. Rutishause, A. Sprowitz, L. Righetti, and A. J. Ijspeert. Passive Compliant Quadruped Robot using Central Pattern Generators for Locomotion Control. *IEEE RAS and EMBS International Conference*, pages 710–710, October 2008.

[19] R. W. Sinnet and A. D. Ames. Extending Two-Dimensional Human-Inspired Bipedal Robotic Walking to Three Dimensions through Ggeometric Reduction. In *American Control Conference (ACC)*, pages 4831–4836, 2012.

[20] M. W. Spong. and F. Bullo. Controlled symmetries and passive walking. *IEEE Transactions on Automatic Control*, 50:1025–31, 2005.

[21] 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.

[22] 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.

[23] M. Vukobratović and B. Borovac. Zero-Moment Point – Thirty-Five Years of its Life. *Intl. J. of Humanoid Robotics*, 1(1):157–157, March 2005.

[24] M. Vukobratović, B. Borovac, D. Surla, and D. Stokic. *Biped Locomotion.* Springer-Verlag, Berlin, March 1990.

[25] 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.

[26] 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)*, Cambridge, MA, 2012.