

From Formal Methods to Algorithmic Implementation of Human Inspired Control on Bipedal Robots

Shishir Nadubettu Yadukumar, Murali Pasupuleti and Aaron D. Ames

Abstract This paper presents the process of translating formal theory and methods to efficient algorithms in the context of human-inspired control of bipedal robots, with the end result being experimentally realized robust and efficient robotic walking with AMBER. We begin by considering human walking data and find outputs (or virtual constraints) that, when calculated from the human data, are described by simple functions of time (termed canonical walking functions). Formally, we construct a torque controller, through model inversion, that drives the outputs of the robot to the outputs of the human as represented by the canonical walking function; while these functions fit the human data well, they do not *a priori* guarantee robotic walking (due to the physical differences between humans and robots). An optimization problem is presented that determines the best fit of the canonical walking function to the human data, while guaranteeing walking for a specific bipedal robot; in addition, constraints can be added that guarantee physically realizable walking. We consider a physical bipedal robot AMBER and define a simple voltage based control law—utilizing only the human outputs and canonical walking function with parameters obtained from the optimization—for which we obtain walking in simulation. Since this controller does not require model inversion, it can be implemented efficiently in software. Moreover, applying this methodology to AMBER experimentally results in robust and efficient “human-like” robotic walking.

1 INTRODUCTION

Humans intrinsically display the following five major characteristics during walking: efficiency, naturalism, stability, simplicity, and versatility. Though human walking is a result of complex neuro-muscular interactions, it seems that the aforementioned high-dimensional behavior can be characterized by low-dimensional repre-

Texas A&M University, College Station, TX 77843
e-mail: {shishirny, muralikris, aames}@tamu.edu

sentation; for example, flat ground human walking behavior appears to be controlled by central pattern generators in spinal cord [11, 16]. This special property motivates the construction of a human-inspired controller for bipeds, which can help robots achieve human-like walking and thus paving way for the transition of bipeds from research labs to real environments. The philosophy behind our work is “simplicity implies robustness”. So the main objective of this paper is to develop a framework which can seamlessly integrate human walking data to design control algorithms which are simple, computationally tractable and easily realizable in physical robots.

Numerous approaches which aim to find the underlying “simplicity” in bipedal walking have been explored. Some of the first fundamental work in this area was by Marc Raibert, with the idea of achieving locomotion through the use of inverted pendulum models to create single-legged hoppers [20], and Tad Mcgeer who introduced the concept of passive walking [14] (which has also been realized on robots with efficient actuation [7]). Passive walking lead to the notion of controlled symmetries which allows for low energy walking [21], and the Spring Loaded Inverted Pendulum (SLIP) models [10, 19] for running robots. Walking has also been looked as a learning process [15] where the learning algorithm determines an optimal control policy by going through a collection of training sets. This method inherently requires several successful walking trials or good training examples in order to learn walking. In addition to these approaches, several methods have been proposed to directly bridge the gap between biomechanics and control theory by looking at human walking data to build models for bipedal robotic walking (see [8, 22] to name a few). Finally, combining many of the above approaches, significant strides have been made in underactuated bipedal walking (without feet) by using the idea of virtual constraints and hybrid zero dynamics (HZD) [23, 13], which resulted in amazingly robust walking even on rough terrain. HZD’s has indeed represented bipedal walking in a very elegant fashion, but implementing a HZD controller on a biped involves the determination of the parameters of the robot through identification experiments [17] which are not only very exhaustive and time consuming but are also not scalable to changes in hardware or robot structure.

This paper attempts to overcome the limitations posed by a HZD controller by using outputs and canonical walking functions which intrinsically capture the major characteristics of human walking behavior; this human-inspired control approach thus aims to further bridge the gap between robotics and control by using human walking data to formally design controllers (as first discussed in [3]). Specifically, by considering human walking data obtained through motion capture of subjects walking on flat ground, we find that certain outputs (or virtual constraints) of the human as calculated from this data can be represented by a special class of functions, termed canonical walking functions, characterized as time response of a linear spring-mass-damper system. Thus, humans appear to act like linear spring-mass-damper systems when walking on flat ground. By forming an optimization algorithm, where the cost is the least squares fit of the canonical walking functions to the human walking data, we obtain parameters for a human-inspired controller that provably results in stable underactuated robotic walking that is as close as possible to human walking. The goal of this paper is to translate these formal methods in human-inspired control

to efficient algorithms which are experimentally realizable on a physical robot; and specifically AMBER, a 2D underactuated bipedal robot (see Fig. 1).

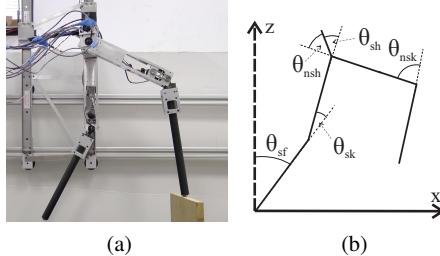


Fig. 1: The bipedal robot AMBER (a) and the angle convention used (b).

The main idea behind algorithmically implementing the formal ideas underlying the human-inspired control approach to bipedal robotic walking is that the inherent robustness present in the human-outputs that are chosen can be utilized to create simple and efficient feedback control strategies. Using the parameters of outputs obtained from the formal optimization problem that provably results in stable robotic walking, we define a simple voltage-based proportional (P) feedback control law on the human-inspired outputs (similar ideas have been explored for robotic manipulators [6, 12]). Since the actuators of AMBER are powered by DC motors, this naturally lends itself to simple implementation on the physical robot. The end result is that the voltage applied to the motors is directly proportional to the error between the desired and actual outputs of the robot, as represented by the canonical walking functions. The algorithmic implementation of this controller is, therefore, very efficient requiring less than 100 lines of pseudo-code. Therefore, the authors conclude that this simplistic algorithm becomes the foundation for this walking robot.

Implementing the algorithms developed on AMBER experimentally results in bipedal robotic walking that is efficient, robust and “human-like.” We argue that this is a direct result of the combination of formal methods with simple realization through the proportional voltage control on outputs that are directly inspired by human walking. In particular, we find good agreement between the simulation and experimental data. This indicates a direct connection between the formal methods and implementation. The experimental output data of the robot can also be related back to the human output data from which the controller was derived, for which there is a strong similarity showing that “human-like” walking is achieved. In addition the robot exhibits robustness in walking even under the influence of a wide variety of disturbances like push-pull, knee strike, tripping, obstacles (as high as 6cm) and even with hits from wooden blocks (see [2]). Moreover, this was achieved by using extremely low power DC motors (11 W). Hence simplicity of the voltage-based P-control on human-inspired outputs adheres closely to the philosophy that “simplicity implies robustness”, thereby rendering our walking algorithm, and the resulting robotic walking, efficient and robust.

2 Formal Methods for Bipedal Robot Modeling and Control

AMBER (short for **A & M Bipedal Experimental Robot**) is a 2D bipedal robot with 5 links (2 calves, 2 thighs and a torso, see Fig. 2). AMBER is 61cm tall with a total mass of 3.3kg (see Table 3b). It is made from aluminum with carbon fiber calves, powered by 4 DC motors and controlled through LabView software by National Instruments. The robot has point feet, and is thus underactuated at the ankle. In addition, since this robot is built for only 2D walking, it is supported in the lateral plane via a boom; this boom does *not* provide support to the robot in the sagittal plane. This means that the torso, through which the boom supports the robot, can freely rotate around the boom. The boom is fixed rigidly to a sliding mechanism (see Fig. 2), which allows the boom and consequently the biped, to move its hip front, back, up and down with minimum friction. The sliding mechanism is rested on a pair of parallel rails.

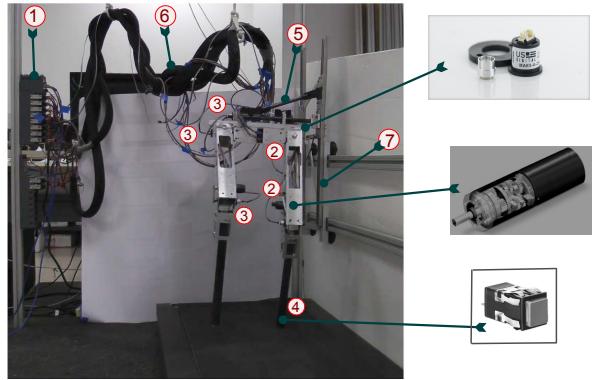


Fig. 2: Amber Experimental Setup. Parts marked are (1): NI cRIO, (2): Maxon DC Motors located in the calf and the torso, (3): Encoders on boom and the joints, (4): Contact switch at the end of the foot, (5): Boom, (6): Wiring with sheath protection, (7): Slider for restricting the motion to the sagittal plane

Let L_c , L_t , L_{tor} be the lengths of the calf, thigh and torso respectively (values are given in Fig. 3(b)) and $\theta = (\theta_{sf}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk})^T$ be the angles of stance foot (foot of the stance leg), stance knee (knee of the stance leg), stance hip, non-stance (swing leg) hip and non-stance knee respectively. These variables form the configuration space of the robot, Q_R , and are shown in Fig. 1(b). Note that every time the swing foot hits the ground, the stance and non-stance nomenclatures are switched in the physical biped. Formally, we represent the bipedal robot as a hybrid system (see [3, 4] for a formal definition):

$$\mathcal{H}\mathcal{C}_R = (X_R, U_R, S_R, \Delta_R, f_R, g_R), \quad (1)$$

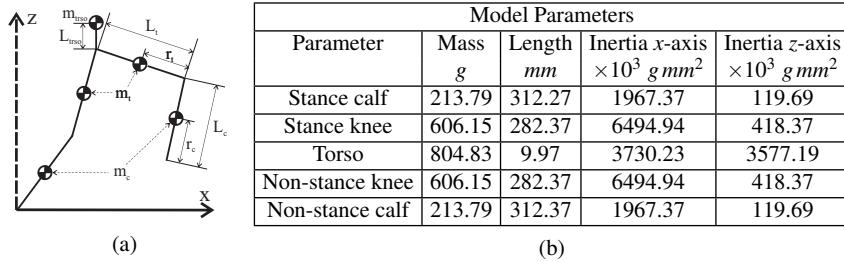


Fig. 3: (a) shows the notations used for masses and lengths for the links. (b) contains the masses, lengths and inertia of the 5 links of AMBER.

where $X_R \subset TQR$ is the domain given by the constraint $h_R(\theta) \geq 0$, where h_R is the height of the swing foot, $U_R \subset \mathbb{R}^4$ is the set of admissible controls, $S_R \subset X_R$ is the guard given by $h_R = 0$, Δ_R is the reset map which provides an instantaneous change in velocity at foot strike, and $\dot{x} = f_R(x) + g_R(x)u$, with $x = (\theta^T, \dot{\theta}^T)^T \in \mathbb{R}^{10}$ and u the torque input, is a control system obtained from the Lagrangian of the robot (which includes the mass and inertia of all links, the motors and the boom [18]). Since the robot is controlled by DC motors, we will also be interested in the control system which includes the DC motor models which results in the control system $\dot{x} = f_{R_V}(x) + g_{R_V}(x)V_{in}$ with voltage, V_{in} , being the control input.

Human-Inspired Functions. By considering human walking data (as described in [3]), we discover that certain outputs (or virtual constraints), computed from the human joint data, display simple behavior; this core observation will be central to the design of human-inspired controllers. With the goal of picking outputs that elucidate the underlying structure of walking through a low-dimensional representation, or “virtual model,” we pick outputs that represent the human (and bipedal robot) as a compass-gait biped [7, 14] and the SLIP model [10]. In particular, the following collection of outputs yields such a representation (as illustrated on the right): the linearization of the x -position of the hip, p_{hip} , given by:

$$\delta p_{\text{hip}}(\theta) = L_c(-\theta_{sf}) + L_t(-\theta_{sf} - \theta_{sk}), \quad (2)$$

the linearization of the 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:

$$\delta m_{nsl}(\theta) = -\theta_{sf} - \theta_{sk} - \theta_{sh} + \theta_{nsh} + \frac{L_c}{L_c + L_t} \theta_{nsk}, \quad (3)$$

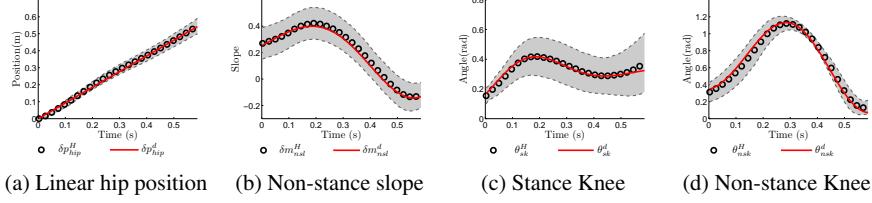


Fig. 4: The black circles indicate the mean of the human output data (see [5]). The grey shaded area indicates the standard deviation from the mean trajectory. The red solid lines are the fits of the canonical functions to the mean human data.

the angle of the stance knee, θ_{sk} , the angle of the non-stance knee, θ_{nsk} , the angle of the torso from vertical, $\theta_{tor}(\theta) = \theta_{sf} + \theta_{sk} + \theta_{sh}$. It is important to note that the linearized form of these outputs, rather than their original nonlinear form [3], is considered to allow for simpler implementation. Inspection of these outputs, as computed from the human data and shown in Fig. 4, reveals that they appear to display very simple behavior. In the case of the (linearized) position of the hip, it appears to essentially be a linear function of time:

$$\delta p_{\text{hip}}^d(t, v) = v_{\text{hip}} t, \quad (4)$$

The remaining outputs, (the non-stance slope δm_{nsl} , the stance knee θ_{sk} , the non-stance knee θ_{nsk} and the torso angle θ_{tor}) appear to act like a second order linear system. This motivated the introduction of the *canonical walking function* [3, 4]:

$$y_H(t, \alpha) = e^{-\alpha_4 t} (\alpha_1 \cos(\alpha_2 t) + \alpha_3 \sin(\alpha_2 t)) + \alpha_5. \quad (5)$$

which is simply the time solution to a linear mass-spring damper system, with $\alpha_4 = \zeta \omega_n$, where ζ is the damping ratio and ω_n is the natural frequency, $\alpha_2 = \omega_d$, where $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ is the damped natural frequency, $\alpha_1 = c_0$ and $\alpha_3 = c_1$, where c_0, c_1 are determined by the initial conditions of the system and $\alpha_5 = g$, where g is the gravity related constant. Performing a least squares fit of the human output data with these functions results in near unity correlations (see [5]); implying that for the specific outputs chosen humans appear to act like linear mass-spring-damper systems. This is an important conclusion because it illustrates the simplicity in behavior that humans display when walking.

Human-Inspired Outputs. Having obtained outputs from the inspection of human data, the goal will be to construct a controller that drives the outputs of the robot to the outputs of the human, $y_a(\theta(t)) \rightarrow y_d(t, \alpha)$, with:

$$y_d(t, \alpha) = \begin{bmatrix} y_H(t, \alpha_{nsl}) \\ y_H(t, \alpha_{sk}) \\ y_H(t, \alpha_{nsk}) \\ y_H(t, \alpha_{tor}) \end{bmatrix}, \quad y_a(\theta) = \begin{bmatrix} \delta m_{nsl}(\theta) \\ \theta_{sk} \\ \theta_{nsk} \\ \theta_{tor}(\theta) \end{bmatrix}, \quad (6)$$

where $y_H(t, \alpha_i)$, $i \in \{nsl, sk, nsk, tor\}$ is the canonical walking function (5) but with parameters, α_i specific to the output being considered. By grouping these parameters with the velocity of the hip, v_{hip} , that appears in (4) results in the vector of parameters $\alpha = (v_{hip}, \alpha_{nsl}, \alpha_{sk}, \alpha_{nsk}, \alpha_{tor}) \in \mathbb{R}^{21}$.

In order to remove the dependence of $y_d(t, \alpha)$, we introduce a parameterization of time based upon the fact that the (linearized) position of the hip is accurately described by a linear function of time. This motivates the following parameterization:

$$\tau(\theta) = \frac{\delta p_{\text{hip}}^R(\theta) - \delta p_{\text{hip}}^R(\theta^+)}{v_{\text{hip}}}, \quad (7)$$

where $p_{\text{hip}}^R(\theta^+)$ is the hip-position of the robot at the beginning of a step, where θ^+ is a point where the height of the non-stance foot is zero, i.e., $h_R(\theta^+) = 0$. Using this parameterization and (6), we define the following *human-inspired output*:

$$y_\alpha(\theta) = y_a(\theta) - y_d(\tau(\theta), \alpha). \quad (8)$$

Human-Inspired Control. Consider again the affine control system (f_R, g_R) associated with the hybrid model of AMBER (1). The human outputs were explicitly chosen so that the decoupling matrix, $A(\theta, \dot{\theta}) = L_{g_R} L_{f_R} y_\alpha(\theta, \dot{\theta})$ with L (Lie derivative) being nonsingular. Therefore, the human-inspired outputs are (vector) relative degree 2 and we can define the following torque controller:

$$u_{(\alpha, \varepsilon)}(\theta, \dot{\theta}) = -A^{-1}(\theta, \dot{\theta}) (L_{f_R}^2 y_\alpha(\theta, \dot{\theta}) + 2\varepsilon L_{f_R} y_\alpha(\theta, \dot{\theta}) + \varepsilon^2 y_\alpha(\theta)). \quad (9)$$

In other words, we can apply input/output linearization to obtain the linear system on the human-inspired outputs: $\ddot{y}_\alpha = -2\varepsilon\dot{y}_\alpha - \varepsilon^2 y_\alpha$. This system is exponentially stable, implying that for $\varepsilon > 0$ the control law $u_{(\alpha, \varepsilon)}$ drives $y_\alpha \rightarrow 0$. More generally, it renders the *zero dynamics surface*:

$$\mathbf{Z}_\alpha = \{(\theta, \dot{\theta}) \in TQ_R : y_\alpha(\theta) = \mathbf{0}, L_{f_R} y_\alpha(\theta, \dot{\theta}) = \mathbf{0}\} \quad (10)$$

invariant and exponentially stable for the *continuous dynamics*. Yet this property does not hold for the hybrid dynamics since discrete impacts in the system cause the state to be “thrown” off of the zero dynamics surface. Therefore, the goal is to achieve *hybrid zero dynamics*: $\Delta_R(S_R \cap \mathbf{Z}_\alpha) \subset \mathbf{Z}_\alpha$, i.e., render the zero dynamics surface invariant through impact. This will imply that the behavior of the robot will be characterizable by the “virtual model” that motivated the output functions under consideration, and will thus allow us to guarantee the existence of walking gaits.

Optimization Theorem. We now present the main theorem (originally introduced in [3, 4, 5]) that will be used to generate the control parameters and experimentally implemented on AMBER to obtain robotic walking. From the mean human walking data, we obtain discrete times, $t^H[k]$, and discrete values for the human output data, $y_i^H[k]$ and the canonical walking functions, $y_i^d(t, \alpha_i)$ for $i \in \text{Output} = \{\text{hip}, \text{nsl}, \text{sk}, \text{nsk}, \text{tor}\}$; for example, $y_{nsl}^H[k] = y_H(kT, \alpha_{nsl})$, where T is the discrete

time interval and $k \in \mathbb{Z}$. We can now define the following human-data cost function:

$$\text{Cost}_{\text{HD}}(\alpha) = \sum_{k=1}^K \sum_{i \in \text{Output}} \left(y_i^H[k] - y_i^d(t^H[k], \alpha_i) \right)^2 \quad (11)$$

which is simply the sum of squared residuals. To determine the parameters for the human walking functions, we need only solve the optimization problem:

$$\alpha^* = \underset{\alpha \in \mathbb{R}^{21}}{\operatorname{argmin}} \text{Cost}_{\text{HD}}(\alpha) \quad (12)$$

which yields the least squares fit of the mean human output data with the canonical walking functions. While this provides an α^* that yields a good fit of the human data (see Fig. 4), these parameters will not result in robotic walking due to the differences between the robot and a human. Therefore, the goal is to determine these parameters which provide the best fit of the human data while simultaneously guaranteeing stable robotic walking for AMBER¹. This motivates the following theorem:

Theorem 1. *The parameters α^* solving the constrained optimization problem:*

$$\alpha^* = \underset{\alpha \in \mathbb{R}^{21}}{\operatorname{argmin}} \text{Cost}_{\text{HD}}(\alpha) \quad (13)$$

$$\text{s.t } y(\vartheta(\alpha)) = \mathbf{0} \quad (\text{C1})$$

$$dy_\alpha(\Delta_\theta \vartheta(\alpha)) \Delta_{\dot{\theta}}(\vartheta(\alpha)) \dot{\vartheta}(\alpha) = \mathbf{0} \quad (\text{C2})$$

$$dh_R(\vartheta(\alpha)) \dot{\vartheta}(\alpha) < 0 \quad (\text{C3})$$

$$\mathcal{D}_Z(\vartheta(\alpha)) < 0 \quad (\text{C4})$$

$$0 < \Delta_Z(\vartheta(\alpha)) < 1 \quad (\text{C5})$$

yield hybrid zero dynamics: $\Delta_R(S_R \cap \mathbf{Z}_{\alpha^*}) \subset \mathbf{Z}_{\alpha^*}$. Moreover, there exists an $\hat{\varepsilon} > 0$ such that for all $\varepsilon > \hat{\varepsilon}$ the hybrid system $\mathcal{H}_R^{(\alpha^*, \varepsilon)}$, obtained by applying the control law (9) to the hybrid control system (1), has a stable periodic orbit with fixed point $(\theta^*, \dot{\theta}^*) \in S_R \cap \mathbf{Z}_{\alpha^*}$ that can be explicitly computed.

It is not possible to introduce all of the elements utilized in Theorem 1 due to space constraints but a detailed explanation can be found in [4]. Of particular importance is the point $(\vartheta(\alpha), \dot{\vartheta}(\alpha)) \in S_R \cap \mathbf{Z}_{\alpha^*}$ on the intersection of the zero dynamics

¹ It is important to note that [22] also presents an optimization problem that results in the least squares fit of human walking data subject to constraints that ensure HZD. Yet the theorem presented here is a substantial departure from the results in [22] in several important ways: [22] considers human joint angles, while we consider output functions, [22] fits high degree (9th order) polynomials to this data to create virtual constraints while we utilize the canonical walking function (which is nonlinear, and has far fewer parameters), [22] defines a configuration at the end of the step a priori and uses these to constrain the parameters of the outputs to ensure HZD while we define the point in terms of the parameters and allow it to change with the parameters as a result. All of these considerations require the use of different methods and theory and, fundamentally, changes the walking achieved.

surface and the guard that can be explicitly computed in terms of the parameters α (this point will later be used in additional constraints that will yield physically realizable walking). In other words, the configuration and velocities at the beginning and end of a step can change with the parameters allowing for a better translation of the outputs to robots which have different mass and length parameters from humans. In addition, $(\vartheta(\alpha), \dot{\vartheta}(\alpha))$ allows for the constraints in the optimization to be framed only in terms of the parameters, α . For these constraints, (C1) and (C2) (when coupled with the way $(\vartheta(\alpha), \dot{\vartheta}(\alpha))$ is computed from the outputs) ensure that the state of the robot is restricted to the zero dynamics surface even through the impacts. (C3) ensures that the non-stance foot intersects the guard transversally. (C4) and (C5) imply the existence and stability of a periodic orbit in the hybrid zero dynamics. In particular, $\mathcal{D}_Z(\vartheta(\alpha))$, which is a function of the energy contained in the zero dynamics, determines the existence of a step (which in turn determines the existence of the periodic orbit). $\Delta_Z(\vartheta(\alpha))$, gives the post impact velocity in the zero dynamics from pre-impact velocity, and therefore (C5) indicates the stability of the resulting periodic orbit. Finally, following from the results in [23], the existence and stability of a periodic orbit in the hybrid zero dynamics implies the stability of a periodic orbit in the full-order dynamics for sufficiently large ε , i.e., the end result is a stable walking gait.

Additional Constraints. The walking that we achieve using Theorem 1 should be physically realizable, which necessitates the additional constraints that ensure that the resulting control parameters will experimentally result in walking with AMBER:

(C6) Foot scuff prevention: The height of the swing foot at any point of time, must be such that it is greater than a quadratic polynomial, $h_R(\theta) > P(\theta)$, where $P(\theta) = ax_f(\theta)^2 + bx_f(\theta) + c$ with $x_f(\theta)$ being the horizontal position of the swing foot w.r.t. the stance foot and

$$a = -\frac{4h_{max}}{SL(\alpha)^2}, \quad b = \frac{4h_{max}SL(\alpha)}{SL(\alpha)^2}, \quad c = -\frac{4h_{max}x_f(\vartheta(\alpha))x_f(\Delta_R(\vartheta(\alpha)))}{SL(\alpha)^2},$$

where $SL(\alpha) = x_f(\vartheta(\alpha)) - x_f(\Delta_R(\vartheta(\alpha)))$ is the step length of the robot, computed from α through $\vartheta(\alpha)$. These constants, therefore, can be adjusted based on the required maximum stance foot height, h_{max} , and step length, $SL(\alpha)$.

(C7) Peak torque: The maximum torque delivered by the motors is limited. Therefore, the peak torque during a walking gait must be: $\max(u_{(\alpha,\varepsilon)}(\theta, \dot{\theta})) < u_{max}$. Here $u_{(\alpha,\varepsilon)}$ is dependent on the parameters α and ε , given in (9) and u_{max} is the maximum torque of the motors (for AMBER, $u_{max} = 2\text{Nm}$).

(C8) Hip-Velocity: The desired hip velocity of the biped must be within reasonable limits. Therefore, we introduce the constraint: $v_{min} < v_{hip} < v_{max}$.

For AMBER, $v_{min} = 0.1\text{m/s}$, $v_{max} = 0.6\text{m/s}$.

(C9) Angular velocities of joints: The maximum angular velocities with which the joints can turn are limited by the maximum angular velocities of the motors. The motors used in AMBER have a maximum angular velocity of 6.5rad/s .

3 Algorithmic Implementation and Experimental Results

The control law proposed in the previous section requires us to linearize the dynamics of AMBER through model inversion, which requires exact values of masses, inertias and dimensions of the robot. This is not only complex to implement but realizing the control law (9) on AMBER could potentially consume both time and resources, and achieving walking may still not be guaranteed due to a potentially inexact model. We, therefore, take a different approach by arguing that due to the “correct” choice of output functions—and specifically the human-inspired outputs—it is possible to obtain walking through simple controllers that are easy to implement and inherently more robust. Specifically, we present a proportional voltage controller on the human-inspired outputs, and demonstrate through simulation that robotic walking is obtained on AMBER. The simplicity of this controller implies that it can be efficiently implemented in software, and the details of this implementation are given. Finally, experimental results are presented showing that bipedal robotic walking is obtained with AMBER that is both efficient and robust.

Human-Inspired Voltage Control. Even if walking is obtained formally through input/output linearization, i.e., model inversion, the controllers are often implemented through PD control on the torque (see, for example, [17]). Since AMBER is not equipped with torque sensors, we sought an alternative method for feedback control implementation. Because AMBER is powered by DC motors, the natural input to consider is voltage, V_{in} , which indirectly affects the torques acting on the joints. Let V_{nsl} , V_{sk} , V_{nsk} and V_{tor} be the voltage input to the motors at the non-stance hip, stance knee, non-stance knee and stance hip, respectively. Define the following human-inspired proportional (P) voltage control law:

$$V_{in} = \begin{bmatrix} V_{nsl}(\theta) \\ V_{sk}(\theta) \\ V_{nsk}(\theta) \\ V_{tor}(\theta) \end{bmatrix} = -K_p y_\alpha(\theta), \quad (14)$$

where K_p is the constant matrix with its diagonal entries being the proportional gains for each of the motors and its non-diagonal entries being zero since the motors are controlled independently. This controller can be applied to the control system $\dot{x} = f_{R_v}(x) + g_{R_v}(x)V_{in}$ modeling the bipedal robot in conjunction with the motors. It can be seen that the control law (proportional control) solely depends on the generalized coordinates of robot (angles), θ , and not on the angular velocities. This marks a drastic change from the traditional ways of computing control. Evidently, and importantly, this avoids computation of angular velocities of the joints, which would have been computationally expensive and inaccurate.

It is important to note that the P voltage control law (14) is equivalent to a PD torque controller, where the derivative (D) constant is specified by the properties of the motor:

$$V_{in} = -K_p y_\alpha(\theta) = R_a i_a + K_\omega \omega \implies u(\theta, \dot{\theta}) = -K_\phi R_a^{-1} K_p y_\alpha(\theta) - K_\phi R_a^{-1} K_\omega \dot{\theta}$$

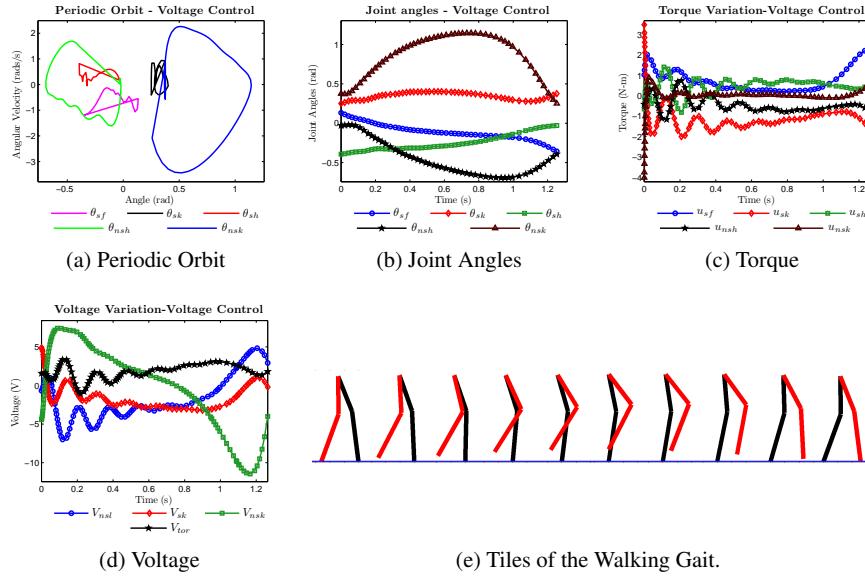
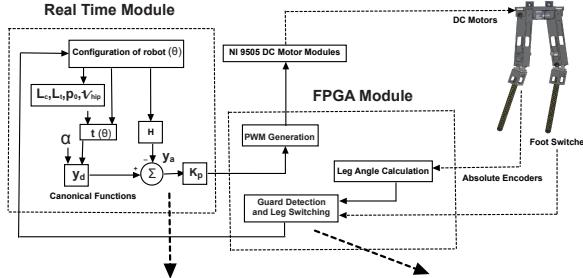


Fig. 5: Walking gait for AMBER obtained in simulation through P-voltage control.

where K_φ is the torque constant matrix, and K_ω is the motor constant matrix. Hence, the control being applied is, in the end, related to the conventional torque PD control methods adopted in literature (see [9]).

Simulation Results. To obtain walking in simulation with AMBER through the formal methods discussed in Sect. 2, we begin with the hybrid model of the robot $\mathcal{H}C_R$ given in (1). Applying the human-inspired controller (9) and solving the optimization problem in Theorem 1 subject to the additional constraints (C6)-(C9) results in a hybrid system $\mathcal{H}_R^{(\alpha^*, \varepsilon)}$ that provably has a stable periodic orbit, i.e., a stable walking gait. The parameters α^* are then used in the P-voltage control, and the resulting closed loop system, which includes the mechanical and electrical models of AMBER is simulated. This results in a new periodic orbit (that is “near” the periodic orbit for the human-inspired torque controller). The resulting walking that is obtained through P-voltage control is shown in Fig. 5, along with the periodic orbit, joint angles, torques and voltages. The torque and voltage figures show oscillations. This is partly due to the application of a single order linear controller on a higher order nonlinear system.

Experimental Setup. AMBER’s experimental set up consists of three main segments: controller, actuators and sensors. Fig. 2 shows the locations of various sensors and actuators on AMBER. The Real Time (RT) processor and the FPGA in the cRIO form the controller, the DC motors powered by the H-bridge modules form the actuators, and the encoders connected to the joints along with contact switches at the feet form the sensors. The RT processor is connected via ethernet to a host

**Algorithm 1** Real Time Module

```

Input: Amber Parameters: Calf Length( $L_c$ ), Thigh Length( $L_t$ );
Input: Optimization Parameters:  $\delta p_{\text{hip}}^R(\theta^+)$ ,  $v_{\text{hip}}$ ,  $\alpha$ ;
Input: Proportional Gain ( $K_p$ );
Input:  $\theta_{sf}$ ,  $\theta_{sk}$ ,  $\theta_{sh}$ ,  $\theta_{nsk}$ ,  $\theta_{nsh}$ ;
Input: L / R stance; // Says which leg is supporting
Input: Encoder Not Working;
Input: Drive-Status;
Output: Enable Motors, Disable Motors ;
Output: PWM Duty Cycles for Voltages ;
1: Run Real Time (RT) VI;
2: Enable Motor ;
3: repeat
4:   Wait till all motors are Enabled
5: until ( Drive-Status == Enable )
6: while ( Stop-RT (or) Encoder-Error != 1 ) do
7:   if ( (Any Joint Angle Exceeds Limits)
        (or) (Encoder Not Working) ) then
8:     Encoder-Error ← 1;
9:   else
10:    Calculate  $\tau(\theta)$  ;
11:    PWM_Count1 ←  $K_p(y_H(\tau(\theta), \alpha_{nsk}) - \delta m_{nsl}(\theta(t)))$ ;
12:    PWM_Count2 ←  $K_p(y_H(\tau(\theta), \alpha_{sk}) - \theta_{sk}(t))$ ;
13:    PWM_Count3 ←  $K_p(y_H(\tau(\theta), \alpha_{nsh}) - \theta_{nsh}(t))$ ;
14:    PWM_Count4 ←  $K_p(y_H(\tau(\theta), \alpha_{tor}) - \theta_{tor}(\theta(t)))$ ;
15:    if (L / R stance) then
16:       $V_{LHIP} \leftarrow$  PWM.Count1;
17:       $V_{LKNEE} \leftarrow$  PWM.Count3;
18:       $V_{RHIP} \leftarrow$  PWM.Count4;
19:       $V_{RKNEE} \leftarrow$  PWM.Count2;
20:    else
21:       $V_{LHIP} \leftarrow$  PWM.Count4;
22:       $V_{LKNEE} \leftarrow$  PWM.Count2;
23:       $V_{RHIP} \leftarrow$  PWM.Count1;
24:       $V_{RKNEE} \leftarrow$  PWM.Count3;
25:    end if
26:  end if
27: end while
28: Disable Motor Drives;
29: Report Errors and Stop the Real Time VI;

```

Algorithm 2 FPGA Module

```

Input: Enable Motors, Disable Motors;
Input: PWM Duty Cycles for Voltages;
Output:  $\theta_{sf}$ ,  $\theta_{sk}$ ,  $\theta_{sh}$ ,  $\theta_{nsh}$ ,  $\theta_{nsk}$  ;
Output: L / R stance; // Says which leg is supporting
Output: Encoder Not Working; Drive-Status;
1: loop
2:   if ( Enable Motors (or) !Disable Motors ) then
3:     Enable all NI9505 DC Motors;
4:   end if
5:   if ( !Enable Motors (or) Disable Motors ) then
6:     Disable all NI9505 DC Motors;
7:   end if
8:   if ( PWM Duty Cycle > 75% ) then
9:     PWM Duty Cycle ← 75%;
10:   end if
11:   if ( Left Leg stance ) then
12:     L / R stance ← 0;
13:   else if ( Right Leg stance ) then
14:     L / R stance ← 1;
15:   end if
16:   if ( Signal low for 2 periods of encoder pulse) then
17:     Encoder Not Working ← 1;
18:   else
19:     Encoder Not Working ← 0;
20:   end if
21:   if (L / R stance ) then
22:      $\theta_{nsh} \leftarrow$  LHIP_ANGLE ;
23:      $\theta_{nsk} \leftarrow$  LKNEE_ANGLE ;
24:      $\theta_{sh} \leftarrow$  RHIP_ANGLE ;
25:      $\theta_{sk} \leftarrow$  RKNEE_ANGLE ;
26:   else
27:      $\theta_{sh} \leftarrow$  LHIP_ANGLE ;
28:      $\theta_{sk} \leftarrow$  LKNEE_ANGLE ;
29:      $\theta_{nsh} \leftarrow$  RHIP_ANGLE ;
30:      $\theta_{nsk} \leftarrow$  RKNEE_ANGLE ;
31:   end if
32:    $\theta_{sf} \leftarrow \theta_{tor} - \theta_{sk} - \theta_{sh}$  ;
33: end loop

```

Fig. 6: Controller overview with Pseudo Algorithms for RT and FPGA

PC for data logging. The Real Time processor carries out floating point operations, while the FPGA interacts with I/O devices and provides hardware level interaction with the actuators and the sensors. AMBER walks on a treadmill, so the treadmill speed (resolution of 0.1 mph) is adjusted to be roughly equal to the desired average

speed of walking (0.8 mph) and then fine-tuned by using an autotransformer. In this manner, we can bring the speed very close to the value predicted in simulation. The robot is then powered on and slowly lowered down to the treadmill; after a couple of steps, the robot steadily falls into a limit cycle. The rest of this section is devoted to explaining the implementation of control law algorithms and the experimental results.

Implementation of Feedback Control Law. The schematic depicting the implementation of voltage based P-control on human-inspired outputs for AMBER is presented in Fig. 6. Contact switches are used to detect the guard and initiate the discrete dynamics for the robot. The variable, \bar{L}/R Stance, is used to identify which of the two legs is a stance leg. When the left leg is in stance phase, the watchdog in the controller checks every 5ms for right foot strike and vice-versa. Debounce logic is implemented, which discards any swing leg contact happening in less than 0.2s from previous foot strike. This eliminates accidental switchings when swing leg lifts off from the ground behind the stance leg.

The pseudo-algorithms used to implement the control law for Labview (RT and FPGA modules) are presented in Fig. 6. It is evident from the size of the code, that a complex task of human-like robotic walking was achieved using not more than 100 lines of pseudo code and minimal resources for hardware implementation (Fig. 7) with voltage-based P control on the human-inspired outputs. This highlights the effectiveness and simplicity of the algorithm used to implement the control law and we claim that, this potentially is one of the reasons why the walking obtained was robust.

Experimental Results. Implementing the algorithm in AMBER results in bipedal robotic walking (see [1] for a video of AMBER walking and responding to external disturbances). Walking tiles from the experiment and the simulation are shown in Fig. 8, and are in good agreement with each other. The similarity between the ex-

Component	Total	Used (%)
Slices	7200	39
Slice Registers	28800	20.8
Slice LUTs	28800	28.1
DSP48s	48	20.8
Block RAMs	48	0

Fig. 7: FPGA Resource Used

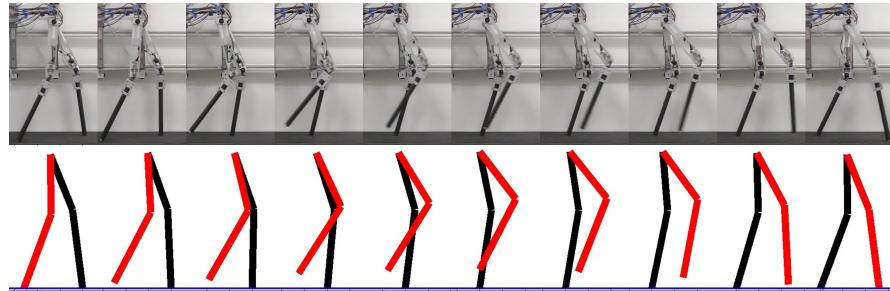


Fig. 8: Experimental walking tiles of AMBER (top) compared with the walking obtained from the simulation (bottom). See [1] for the video of AMBER walking.

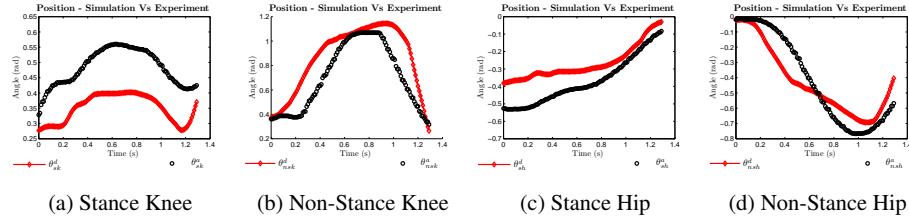


Fig. 9: Experimental vs. simulated angles over one step.

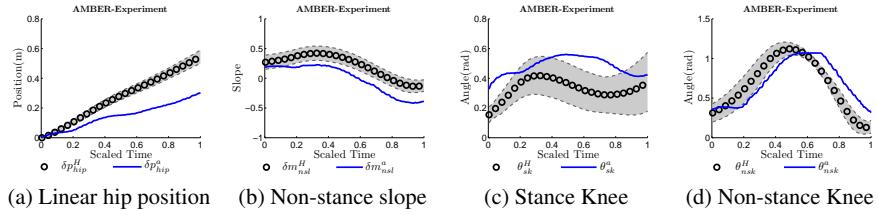


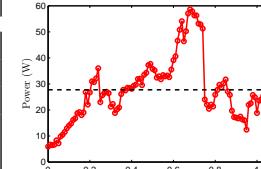
Fig. 10: Experimental robotic output data compared to human output data.

perimental and simulated behavior can be further seen by comparing the trajectories of the joint angles with the simulation, as shown in Fig. 9.

The human-inspired controller design has made the resulting walking efficient, robust and “human-like”. The specific cost of transport (electrical) for AMBER walking at 0.44 m/s is 1.88 using an average power of 27 W, which is very low compared to commercial robots like Honda [7] and it also has the least installed power to weight ratio (W/kg) among robots with no compliance as indicated in Table 1. In addition, the walking achieved with AMBER is incredibly robust; with no changes to the controller, AMBER is able to successfully handle disturbances like push-pull, hitting, tripping and even rough terrains with ease (see video: [2]). Finally, comparison between AMBER outputs with the human data, from which the controller was originally derived (see Fig. 10), demonstrate that the walking is remarkably “human-like” despite the vast differences between AMBER and humans.

4 Conclusions

This paper successfully translated formal methods in human-inspired control to efficient algorithmic implementation and finally experimentally realized walking on AMBER. Specifically, formal methods utilizing model inversion were presented that provably result in walking; these were translated to implementable control strategies through voltage-based P-control on the human-inspired outputs. The simplicity

Power Plot	Robot	Compliance	Value	Reference
	AMBER	No	13.2	44W, 3.33kg
	ERNIE	Yes	53.7	1 kW ¹ , 18.6kg [23]
	RABBIT	No	93.2	2.98 kW ² , 32kg [23]
	NAO H25 V3.3	No	35.4	177W ³ , 5kg
	DARWIN	No	27.6	80W ⁴ , 2.9kg
	MABEL	Yes	58.6	3.28 kW ⁵ , 56kg [9]
	Cornell Biped	Yes	1.5	19W ⁶ , 12.7kg [7]

¹ EC 45-136212 (250 W), so for 4 motors total power: 1 kW

² RS 420J performance curves indicate 1 HP motor, for 4 motors total power: 2.98 kW

³ Type1: RE-MAX 17 (4W), Type2: RE-MAX 24 (11W). Type1 Motors on Head-2, Type1 Motors on Arms -12, Type2 Motors on Legs- 11. Total power for 25 motors: 177W

⁴ Specifications of Dynamixel RX-28 at 12 V has values of RE-MAX 17 motor(4W) with 1:193 gear ratio, so for 20 motor modules total power: 80 W

⁵ QBO5600-X0X (843.892 W)- 2, QBO5601-X0X(798.605 W) - 2, so for 4 motors total power: 3.28 KW

⁶ For two 9.5 Watt 6.4 oz MicroMotors, total power: 19 W

Table 1: Power to Weight Ratio (W/kg)

of the algorithmic implementation of this control law resulted in low computation overhead thereby enabling us to use a time step of 5ms for each calculation and minimal hardware resources (39% of the FPGA). With no feet actuation, the overall energy efficiency is enhanced, which enabled us to have the lowest installed W/kg when compared with the contemporary robots, as shown in Table 1. While it must be pointed out that some of the robots also carry support electronics and some are meant for running (MABEL) which has resulted in them requiring higher power; nevertheless, we claim that the proposed method of voltage-based P-control on human-inspired outputs can result in robust walking with a very good cost of transport. While achieving a walking gait that is very close to the natural human walking gait, the biped is also tolerant to changes in terrain (6cm), change of treadmill speeds (12.5%) and even force disturbances on all of the links of the robot. It is important to highlight that the proposed voltage-based control law is dependent only on the configuration variables as opposed to using speed and acceleration feedback, which constitute the inherent simplicity and advantage of indirectly affecting the torque produced at a joint through voltage at the motor level. The end result is robust walking both in simulation as well as in the experiment.

Acknowledgements The authors would like thank Michael Zeagler and Jordan Lack for their support in machining and assembling of AMBER. We also thank National Instruments for their support for the project by providing the necessary hardware and software, and specifically Andy Chang for numerous discussions on implementation. This work is supported by NSF grants CNS-0953823 and CNS-1136104, NHARP award 00512-0184-2009, NASA grant NNX12AB58G.

References

1. Amber walking and undergoing robustness tests. <http://youtu.be/SYXWoNU8QUE>.
2. Robustness tests conducted on AMBER. <http://youtu.be/RgQ8atV1NW0>.
3. A. D. Ames. *First Steps Toward Automatically Generating Bipedal Robotic Walking from Human Data*, volume 422, chapter Robot Motion and Control, Lecture Notes in Control and Information Sciences. K. Kozlowski eds, 2012.
4. A. D. Ames. First steps toward underactuated human-inspired bipedal robotic walking. In *2012 IEEE Conference on Robotics and Automation*, 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 *Hybrid Systems: Computation and Control, 2012*, Beijing, China, April 2012.
6. T. Burg, D. Dawson, J. Hu, and M. de Queiroz. An adaptive partial state-feedback controller for RLED robot manipulators. *Automatic Control, IEEE Transactions on*, 41(7):1024 –1030, jul 1996.
7. S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, February 2005.
8. H. Geyer and H. Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(3):263 –273, june 2010.
9. J. W. Grizzle, J. Hurst, B. Morris, H. Park, and K. Sreenath. MABEL, a new robotic bipedal walker and runner. In *American Control Conference*, pages 2030–2036, St. Louis, MO, USA, June 2009.
10. P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Rev.*, 48:207–304, February 2006.
11. A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, May 2008.
12. C. Liu, C. C. Cheah, and J. E. Slotine. Adaptive jacobian tracking control of rigid-link electrically driven robots based on visual task-space information. *Automatica*, 42(9):1491 – 1501, 2006.
13. I. R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, March 2011.
14. T. McGeer. Passive dynamic walking. *Intl. J. of Robotics Research*, 9(2):62–82, April 1990.
15. J. Morimoto, G. Cheng, C. Atkenson, and G. Zeglin. A simple reinforcement learning algorithm for biped walking. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA, May, 2004.
16. J. Bo. Nielsen. How we walk: Central control of muscle activity during human walking. *The Neuroscientist*, 9(3):195–204, 2003.
17. H-W. Park, K. Sreenath, J. Hurst, and J. W. Grizzle. Identification of a bipedal robot with a compliant drivetrain: Parameter estimation for control design. *IEEE Control Systems Magazine*, 31(2):63–88, April 2011.
18. M. Pasupuleti, S. Nadubettu Yadukumar, and A. D. Ames. Human-inspired underactuated bipedal robotic walking with amber on flat-ground, up-slope and uneven terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Algarve, Portugal, 2012.
19. I. Pouliquen 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.
20. M. H. Raibert. Legged robots. *Communications of the ACM*, 29(6):499–514, 1986.
21. M. W. Spong and F. Bullo. Controlled symmetries and passive walking. *IEEE TAC*, 50(7):1025–1031, 2005.
22. S. Srinivasan, I. A. Raptis, and E. R. Westervelt. Low-dimensional sagittal plane model of normal human walking. *ASME J. of Biomechanical Eng.*, 130(5), 2008.
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, June 2007.