ROBOT LOCOMOTION CONTROLLER GENERATION THROUGH HUMAN INSPIRED OPTIMIZATION

A Thesis

by

MATTHEW JOSEPH POWELL

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 Ames
Committee Members,   Sivakumar Rathinam
                        John Hurtado
                        Nancy Amato
Head of Department,   Andreas A. Polycarpou

December 2013

Major Subject: Mechanical Engineering

ABSTRACT

This thesis presents an approach to the formal design, optimization and implementation of bipedal robotic walking controllers, with experimental application on two biped platforms. Standard rigid-body modeling is used to construct a hybrid system model of robotic walking; this model estimates the motion of the robot hardware under a given control action. The primary objective of this thesis is the construction of a control law which effects, on the robot, a periodic "walking" behavior. The process begins with examination of human walking data—specifically outputs of human walking—which provide inspiration for the construction of formal walking control laws. These controllers drive the robot to a low-dimensional representation, termed the partial hybrid zero dynamics, which is shaped by the parameters of the outputs describing the human output data. The main result of this paper is an optimization problem that produces a low-dimensional representation that "best" fits the human data while simultaneously enforcing constraints that ensure a stable periodic orbit and constraints which model the physical limitations of the robot hardware. This formal result is demonstrated through simulation and utilized to obtain 3D walking experimentally with an Aldebaran NAO robot and NASA's prototype Leg Testbed robot.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Thesis: the state-based torque control law, $u(x)$, provided by the Human-Inspired Control approach to bipedal robotic walking evokes stable walking behaviors in a hybrid system model of a given anthropomorphic robotic biped.

(a) NAO
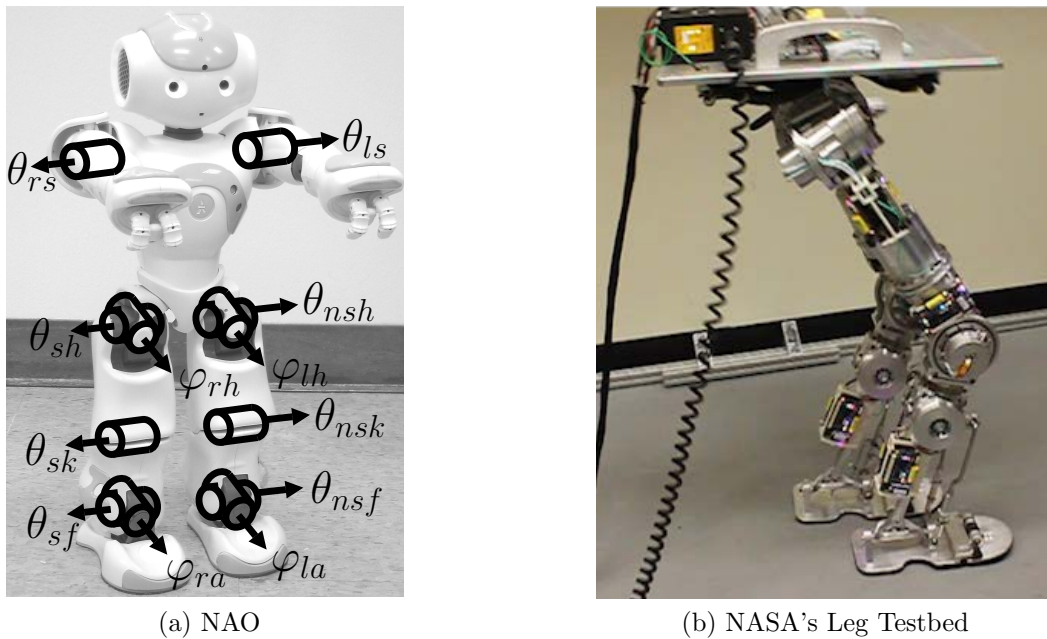
(b) NASA's Leg Testbed

Figure 1.1: Anthropomorphic bipedal robots: NAO and NASA's Leg Testbed.

The term "robot" in this thesis takes a specific meaning: a robot is an anthropomorphic biped which takes the form of a machine, such as the ones shown in Figure 1.1a and Figure 1.1b, comprised of rigid mechanical limb segments, connected by rotary or prismatic joints, and designed to resemble the human lower body. DC motors, hydraulics, or pneumatics are the primary source of mechanical input (force and torque) in robots; these actuators, together with the influence of gravity and forceful

1

interaction with the environment, dictate the motion of a robot's limbs. As control theorists and engineers, our task is to design actuator torque control laws (mathematical controllers implemented via computer and micro controller code) which use knowledge of the robot (in the form of a model of the structure, sensors and actuators) to accomplish a desired behavior in our robot, per project specifications. The behavior of interest in this thesis is stable walking and the purpose of this document is to present a method, termed the Human-Inspired Control method, of generating torque control laws which drive the robot into a stable walking gait.

1.1 Motivation for Walking Robots

Tasks that require humans to work in hazardous environments, e.g. those with high risk of exposure to toxic chemicals, radiation, extreme temperature, unstable terrain, or falling debris, are excellent potential applications for walking humanoid robots. The purpose of robots in this context would be to reduce the time human operators spend in such environments. Successful implementation of robots in these scenarios would give such human operators higher quality of life, by definition, and could potentially reduce the monetary cost associated with such tasks. Walking robots are preferred over their wheeled counterparts as legs provide a significant mobility advantage; and to completely replace a human operator, a robot would have to have mobility on par with that of humans. Furthermore, understanding of bipedal locomotion from a robotics perspective yields useful cross-industry knowledge. Discovering how to make a robot walk can help us understand how to build better prosthetic and rehabilitation devices [33, 15].

1.2 Notable Ideas in Bipedal Robotic Walking Research

This section highlights other researchers' work in the field of bipedal walking robots and in particular, those methods which most aided the control design of this

thesis. Specifically, the Zero-Moment Point control method for bipedal robotic walking is a popular paradigm for achieving conservative, robust walking and has been implemented successfully on numerous robots. Another approach in the literature is the Hybrid Zero Dynamics approach, in which formal guarantees are made about the stability of the modeled walking. The following sections briefly describe work done in each of these areas.

1.2.1 Planning Zero-Moment Point (ZMP) Trajectories

The most prevalent concept utilized in modern control design for bipedal robotic walking is the Zero-Moment Point (ZMP). Several (sometimes inconsistent) definitions have been applied to the ZMP concept; the definition we prefer is that the "ZMP is defined as that point on the ground at which the net moment of the inertial force and the gravity force has no component along the horizontal axes" [41]. Informally, the ZMP can be used to determine whether a robot, standing on one or two feet, is balanced or falling. The ZMP is calculated via:

$$x_{zmp} = \frac{\sum M_y}{\sum F} \quad y_{zmp} = \frac{\sum M_x}{\sum F} \qquad (1.1)$$

where $\sum M_x$ and $\sum M_y$ represent the resultant moments acting on the support polygon and F is the reaction force. The ZMP is useful because it describes a state of equilibrium in the forces acting on the biped's support base; i.e., when a robot's ZMP lies within plane of the stance foot, the foot will not rotate. Therefore, as long as the ZMP does not leave the biped's support base during walking, the robot will not fall over.

Some researchers plan foot-step locations and trajectories of the ZMP – to keep it within the footsteps – over multiple steps and then compute inverse kinematics to determine corresponding joint angle trajectories, which are finally fed into stiff

3

position controllers to achieve the desired motion. Online modifications to ZMP trajectories via control of the biped's torso have also been proposed [26, 39]. Several notable robots employ ZMP methods, including Honda's Asimo [32]. Even the Aldebaran NAO robot, which we use in experimental implementation of the work in this thesis, is equipped with built-in walking algorithms that employ the ZMP concept! Other notable online ZMP approaches are given in additional references [24]. As the ZMP is such a prevalent control scheme, accurate sensing and estimation of the ZMP for robot hardware is an active area of research [9]. ZMP data from human walking experiments with "rigid shoes" is used to characterize flat footed walking [34] and a similar approach is applied to passivity based robots also walking with flat feet [10].

1.2.2 Hybrid Zero Dynamics (HZD)

Most closely related to the work in this thesis is the use of control to achieve Hybrid Zero Dynamics [43] in the hybrid system model of a robot. Hybrid system models encapsulate the continuous (foot swing) and discrete dynamics (foot strike) of rigid-body robot walking in a formal framework. The theory of Hybrid Zero Dynamics extends the concept of Zero Dynamics [19] in which continuous-time state feedback controllers create a *zero dynamics manifold* which is a low-dimensional representation of the full dynamics. Specifically, in Hybrid Zero dynamics, the zero dynamics manifold, $\mathcal{Z}$, must be invariant through impact, Δ, i.e.

$$\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z} \tag{1.2}$$

where $\mathcal{S}$ denotes the guard of a given hybrid system. Bipedal robotic walking controller design using HZD involves specification of an exponentially stable continuous-time controller and corresponding periodic orbit which satisfy (1.2). In the work of Grizzle et al. [14, 43] Bezier polynomials are used in the construction of state-

feedback control laws; Bezier polynomials are chosen as they provide analytic means for ensuring HZD. Work has also been done to control the evolution of the ZMP within a HZD-based controller [8]. In a similar light, in this thesis, we introduce the notion of "Partial Hybrid Zero Dynamics" and use constraints in an optimization problem to ensure that the evolution of the ZMP within our PHZD-based controller remains in the sole of the stance foot.

1.3   The Human-Inspired Method

This work leverages the authors' previous discovery [3, 4] that certain outputs of the human locomotion control system can be represented by the solution to an under-damped, second-order mass spring damper system; these outputs, thus, represent a low-dimensional system which encodes the fundamental behaviors of human walking[1]. When applied through feedback linearization control [35], these outputs yielded human-like walking for the NAO robot, despite the mismatch in mass and length distribution between the NAO and humans. Similar to this control approach is the use of other low-dimensional representations used to achieve robotic walking, such as the spring-loaded inverted pendulum [16], or SLIP model, and the construction of Hybrid Zero Dynamics [43] surfaces.

This thesis proposes a formal *human-inspired optimization* (HIO) which provably results in exponentially stable bipedal robotic walking *and* satisfies many of the physical constraints necessary to realize the walking experimentally. Specifically, the optimization minimizes an objective function which is the least-squares fit of the output functions of the robot to the human output data. Constraints are enforced which guarantee that the zero dynamics surface associated with the certain output functions is invariant through impact resulting in a *partial hybrid zero dynamics*

---

[1]Human walking data were obtained in a set of experiments conducted by Dr. Bajcsy at the University of California, Berkeley [40].

[3]. These constraints, together with a specific choice of (linear) output functions, allow for closed-form computation of the solution to the dynamics of the robot over the course of one step, i.e., the behavior of the robot can be determined *without integrating the dynamics of the system.* This allows for the computation of physical constraints required for experimental implementation, such as the ZMP and friction, to be added to the HIO as constraints and computed in a feasible time-frame (as opposed to the time required to integrate the full dynamics of the system, which in the case of the robot model considered in this paper is 20-dimensional). These formal results are verified both in simulation and in experiment with the NAO robot and NASA's prototype biped system.

1.4 Organization of This Thesis

In Section 2, we discuss how to construct a model of unsupported, 3D bipedal robotic walking using standard rigid-body assumptions: the method of Lagrange is applied to compute equations of motion of the robot and robot-ground interactions are model as instantaneous rigid impacts. The specific walking gait, or "behavior", we consider is flat-footed walking on level ground. At the end of Section 2, we show how all aspects of our model – continuous dynamics, impact equations and admissibility constraints on the walking gait – can be represented in a compact form via the construction of a hybrid system model. In Section 3, we discuss the main result of this thesis: how to create control laws for the modeled robot which yield stable periodic walking. Specifically, we describe the collection and examination of experimental human walking data, the construction of a parameterized Feedback Linearization control law based upon suggestions from these human data, and finally, the formulation of an optimization problem which determines values of the control parameters corresponding to stable robot walking controllers. The results from ap-

6

plication of our method to the Leg Testbed and the NAO robots are presented in Section 4. Here, simulations are performed using the hybrid system model of Section 2 together with the optimized walking controllers, obtained through the methods of Section 3. Furthermore, trajectories from simulation are used in walking experiments on the actual hardware; we show that the actual robot systems take several steps using controllers produced by our method. We give concluding remarks in Section 5.

2. ROBOT WALKING MODEL

The purpose of this section is to show how to construct a mathematical model of a biped robot; this model will be used to determine the relationship between control commands and observed motions. An accurate mathematical model of the robot is an essential component of successful torque control as the proposed control design method uses the modeled dynamics in the computation of torque commands.

As commonly done in the literature, in this thesis a robot is modeled as a system of rigid-bodies, termed "links", connected by "joints" which constrain the relative motion of two connected links to be rotation about a common joint axis. The inertial and length properties of each rigid link can be obtained through measurement, estimation (system identification), or CAD modeling. Standard methods are used to compute kinematics and dynamics of the rigid-body system; namely, the Euler-Lagrange equations are used to compute the dynamics of the robot model under holonomic constraints. These dynamics provide an estimation of the movement of the actual robot for a given set of torque commands. Interaction of the robot with the environment, through contact of the robot's feet with the walking surface, is modeled through impulsive impact events; these are assumed to be perfectly inelastic and to occur instantaneously.

The final product of this section is a hybrid system model of bipedal robotic walking which encapsulates phases of continuous evolution, as modeled by Lagrangian dynamics, and intermittent discrete impact events. This model provides an excellent representation of the relationship between controller commands and the resulting robot motion and does so in a manner that is amenable to the study of "stable robotic walking" via the methods of Poincaré.

2.1 Robot Model Properties

The robots of concern in this study are anthropomorphic bipeds. As such, we refer to components of the robot by the names of their human analogues. For example, the NAO and Leg Testbed robots have limb segments: feet, calves, thighs, hips and a torso; these segments are connected by joints: ankles, knees and hips. During walking experiments, a robot's limb segments undergo continuous deformation caused by forceful interaction with the environment; however, for modeling purposes this deformation is considered to be negligible; hence, the robot is a system of rigid-bodies. The Link-Segment Model Development found in [44] gives an excellent description of the properties of the robot models used in this thesis. Specifically, the robot model, as shown in Figure 2.1 , consists of:

- **Limb segments** - each limb segment, $\ell \in \{f, c, t, T\}$, corresponding to the foot, calf, thigh and torso, has a fixed mass, m_ℓ, located as a point mass at its center of mass (COM). The position of each COM is described by a line segment, r_ℓ, which starts at the distal joint of the limb segment.

  - The mass moment of inertia, I_ℓ, of each limb segment about its mass center is constant with respect to time.

  - The length of each segment, L_ℓ, remains constant with respect to time.

  Values of the mass, inertia, length and center of mass of the limb segments of a given robot are obtained through measurement, estimation, or CAD models.

- **Joints** - each joint is considered to be a frictionless hinge joint, driven by an ideal actuator, i.e. an ideal torque source with infinite bandwidth. Limits on the maximum torque capabilities of the actual robot actuators, which are

typically DC motors, are enforced on the robot model controller via constraints in the controller optimization.

2.1.1 Assumptions on the Robot Walking Gait

The phrase "bipedal walking" encompasses many qualitatively different behaviors, referred to as "walking gaits". A bipedal walking gait – human or robotic – is characterized by the (temporal or spatial) evolution of biped-environment contact points, i.e. locations on the biped's feet which are in contact with the walking surface. This thesis considers a single, specific type of bipedal walking gait consisting of alternating phases of single and double support. In the single-support phase considered here, the plane of one foot, termed the "stance foot", remains in contact with the ground while the other, "nonstance foot", remains entirely above the ground. Phases of double support occur instantaneously when the nonstance foot impacts the ground – these phases are modeled by impact mechanics as described in Section 2.3. The stance/nonstance naming convention is used throughout the thesis when identifying qualities and quantities corresponding to the leg in contact with the ground – the stance leg – and the opposite leg, termed the nonstance leg, which remains off the ground in single support. The primary objective of this work is to construct controllers which evoke this single-support / double-support flat-footed gait on a robot and in a symmetric and periodic manner.

2.1.2 Forward Kinematics: Coordinates and Convention

Forward kinematics maps [23], computed over generalized coordinates, are used to describe the position and orientation of any point on the robot with respect to a fixed reference frame. Here, the generalized coordinates are chosen to be the robot's configuration (joint) space together with the coordinates of a body-fixed (base) frame.
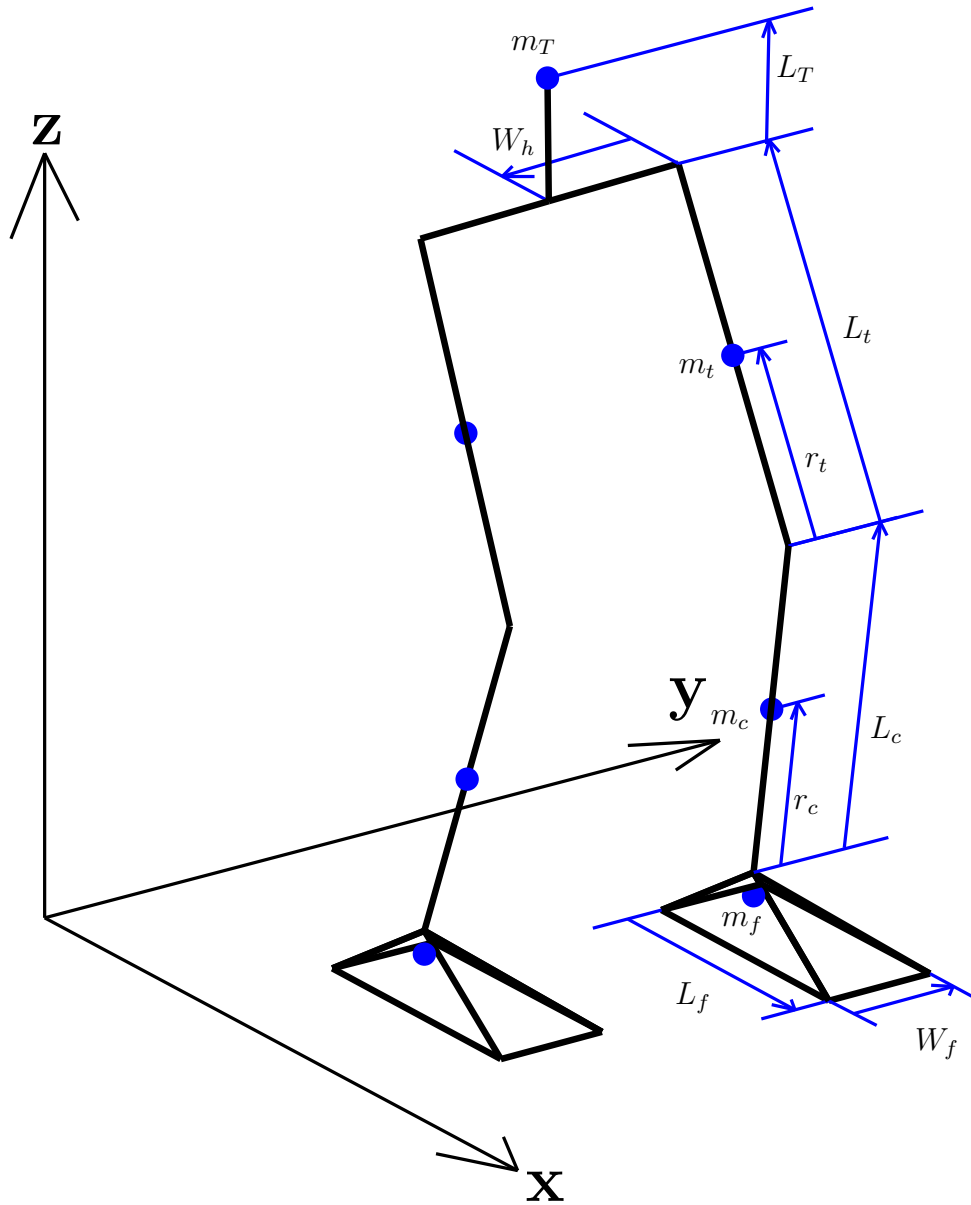
Figure 2.1: A depiction of the modeled rigid-body robot, comprised of links $\ell \in \{f, c, t, T\}$ corresponding to the foot, calf, thigh and torso, each with mass m_ℓ, inertia I_ℓ, length L_ℓ and center of mass r_ℓ. The width of a hip is given by W_h and the width of the foot is W_f.

The configuration space, $\mathcal{Q}^R \in \mathbb{R}^{10}$, of the system is given in coordinates by:

$$q = (\varphi_{sa}, \theta_{sa}, \theta_{sk}, \theta_{sh}, \varphi_{sh}, \varphi_{nsh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa}, \varphi_{nsa})^T, \qquad (2.1)$$

where, as illustrated in Figure 1.1a, φ_{sa}, φ_{sh}, φ_{nsh}, and φ_{nsa} are the stance ankle, stance hip, nonstance hip and nonstance ankle roll angles, respectively, and θ_{sa}, θ_{sk}, θ_{sh}, θ_{nsh}, θ_{nsk}, and θ_{nsa} are the stance ankle, stance knee, stance hip, nonstance hip, nonstance knee and nonstance ankle pitch angles, respectively. Note that the configuration is the 3D version of the commonly employed seven-link biped model [17, 6]. To complete the generalized coordinates, as in [13], let R_0 be a fixed inertial frame and let R_b be a reference frame attached to the projection of the ankle joint onto the stance foot. Let $p_b \in \mathbb{R}^3$ be the Cartesian position of R_b with respect to R_0 and let $\phi_b \in SO(3)$ be the orientation. Then $q_e = (p_b, \phi_b, q) \in \mathcal{Q}_e = \mathbb{R}^3 \times SO(3) \times \mathcal{Q}^R$ is set of generalized coordinates for the robot model.

The position and orientation of a Cartesian reference frame, a, with respect to another Cartesian reference frame, b, are denoted $p_b^a(q)$ and $R_b^a(q)$, where $p_b^a(q)$ is a Euclidean vector and $R_b^a(q)$ is a rotation matrix. These relations can be grouped into a single entity, termed a *homogeneous transformation* [37]. Homogeneous transformations are used to establish position and orientation of reference frames, fixed to key locations on the robot, with respect to the world frame: this is useful for determining when the nonstance foot hits the ground and for visualization purposes (Figure 2.1 and all simulation figures in this thesis use homogeneous transformations to compute line segments of the robot model). Furthermore, homogeneous transformations are also used in solving inverse kinematics problems, such as those posed in the optimization of Section 3.3 .

2.2 Robot Model Dynamics and Constraints

The dynamics of the robot model are the relationship between torque input and observed link segment motion as described by *equations of motion* which can be formulated via several methods, e.g. Newton-Euler or Hamiltonian's method. Here the Euler-Lagrange equations (see standard references [23]) are used. The Lagrangian, $\mathcal{L}_e : TQ_e \to \mathbb{R}$, of the unconstrained robot model is calculated via:

$$\mathcal{L}_e(q_e, \dot{q}_e) := T_e(q_e, \dot{q}_e) - V_e(q_e) \tag{2.2}$$

where $T_e(q_e, \dot{q}_e)$ and $V_e(q_e)$ are the kinetic and potential energy of the robot, respectively. The Lagrangian is used to calculate equations of motion:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_e}{\partial \dot{q}_e}\right) - \frac{\partial \mathcal{L}_e}{\partial q_e} = B_e(q_e)u, \tag{2.3}$$

which are commonly reorganized into the standard equations of motion for a frictionless robot manipulator:

$$D_e(q_e)\ddot{q}_e + H_e(q_e, \dot{q}_e) = B_e(q_e)u. \tag{2.4}$$

with $D_e(q_e) \in \mathbb{R}^{16 \times 16}$ a generalized inertia matrix, $H_e(q_e, \dot{q}_e) \in \mathbb{R}^{16}$ a vector of Coriolis and gravity terms and $B_e(q_e) \in \mathbb{R}^{16 \times 10}$ a torque distribution map. These equations of motion describe the dynamics of the robot when it is not in contact with any other objects, including the ground, thus the robot is simply "floating". In the next section, we describe how to incorporate the ground-reaction forces – experienced at the interface between the ground and the sole of the robot's stance foot – into the dynamical model.

2.2.1 Foot-Ground Contact Constraints

The flat stance foot assumption described in Section 2.1.1 is modeled as a holonomic constraint on the generalized coordinates [13]:

$$\eta(q_e) = \begin{bmatrix} p_b(q_e) \\ \phi_b(q_e) \end{bmatrix} = \text{constant}, \qquad (2.5)$$

which simply means that the stance foot position, p_b, and orientation, ϕ_b, must remain fixed in the world frame. Taking the first and second time-derivative of both sides of (2.5) yields:

$$J(q_e)\dot{q}_e = 0 \qquad (2.6)$$

$$\dot{J}(q_e, \dot{q}_e)\dot{q}_e + J(q_e)\ddot{q}_e = 0 \qquad (2.7)$$

where $J = \frac{\partial \eta(q_e)}{\partial q_e}$ is commonly referred to as the Jacobian of $\eta(q_e)$ with respect to q_e. When enforced, these constraints ensure that the velocity of the sole of the stance foot is constant, and thus, can be used to regulate a flat stance foot on the robot. Additionally, the enforcement of these holonomic constraints reduces the degrees of freedom of the system to 10.

2.2.2 Lagrangian Dynamics with Constraints

The Euler-Lagrange equations of motion for constrained robot manipulators can be written in the form [23]:

$$D_e(q_e)\ddot{q}_e + H_e(q_e, \dot{q}_e) = B_e(q_e)u + J^T(q_e)F_{st}, \qquad (2.8)$$

14

where F_{st} is a vector of *Lagrange Multipliers*. For our specific choice of $J(q_e)$, i.e. the Jacobian of the position and orientation of the sole of the stance foot, the elements in F_{st} are the resulting reaction forces and moments acting on the robot's foot at the position described by the projection of the ankle onto the sole. To obtain an expression for F_{st}, first rewrite (2.8)

$$\ddot{q}_e = D_e^{-1}(q_e)(-H_e(q_e, \dot{q}_e) + B_e(q_e)u + J^T(q_e)F_{st}), \tag{2.9}$$

and then substitute (2.9) into (2.7) to obtain

$$\dot{J}(q_e, \dot{q}_e)\dot{q}_e + J(q_e)D_e^{-1}(q_e)(-H_e(q_e, \dot{q}_e) + B_e(q_e)u + J^T(q_e)F_{st}) = 0 \tag{2.10}$$

Finally, rewrite (2.10)

$$F_{st} = \left[J(q_e)D_e^{-1}(q_e)J^T(q_e) \right]^{-1} \left[J(q_e)D_e^{-1}(q_e)(H_e(q_e, \dot{q}_e) - B_e(q_e)u) - \dot{J}(q_e, \dot{q}_e)\dot{q}_e \right], \tag{2.11}$$

which is an expression for the reaction forces and wrenches describing the influence of the ground on the robot, at the interface between the sole of the foot and the ground; (2.11) is used to verify that the reaction forces induced by our control law u satisfy ZMP conditions. These forces and moments, described by (2.11), are substituted into (2.8) to yield the constrained equations of motion for the biped:

$$D(q)\ddot{q} + H(q, \dot{q}) = B(q)u \tag{2.12}$$

with $D(q) \in \mathbb{R}^{10 \times 10}$ a constrained generalized inertia matrix, $H(q, \dot{q}) \in \mathbb{R}^{10}$ and $B(q) \in \mathbb{R}^{10 \times 10}$.

This section presents the computation of the constraints which must be satisfied to maintain planar contact between the stance foot and the ground.

Stance Foot Constraints. The element-wise Lagrange multipliers, F_{st}, in (2.11), are labeled

$$F_{st} = (F_{st}^{fx}, F_{st}^{fy}, F_{st}^{fz}, F_{st}^{mx}, F_{st}^{my}, F_{st}^{mz}) \tag{2.13}$$

where the first three components are the forces and the last three components are the moments acting on the stance foot. To prevent rotation about an edge, the following constraints on the ground reaction moment must hold [7]:

$$-\frac{w_f}{2}F_{st}^{fz} < F_{st}^{mx} < \frac{w_f}{2}F_{st}^{fz} \tag{2.14}$$

$$-l_h F_{st}^{fz} < F_{st}^{my} < l_t F_{st}^{fz}, \tag{2.15}$$

where w_f is the width of the foot, l_t is the length of the toe and l_h is the length of the heel. When these inequalities are satisfied, the ZMP (Zero Moment Point) of the biped is located within the plane of the stance foot [20, 41].

Nonstance Foot Constraints. To satisfy the flat foot modeling assumption, the nonstance foot must make planar contact with the ground, i.e., the foot must land flat. Therefore, define two kinematic outputs computed over the modeled-robot's state: $\psi_x^R(q)$, the absolute roll angle of the nonstance foot, and $\psi_y^R(q)$, the absolute pitch angle of the nonstance foot. These constraints are enforced via control, as discussed in Section 3.2; where the nonstance foot is driven to be parallel to the ground throughout the entire gait; as a result, $\psi_x^R(q(t)) \equiv \psi_y^R(q(t)) \equiv 0$ for all t.

2.3 Impact Mechanics

An impact – in the gait of consideration – occurs when the nonstance foot hits the ground. Based on the instantaneous double-support phase assumption, the nonstance foot immediately becomes the stance foot after impact and furthermore, the stance foot is assumed to always be in planar contact with the ground. Therefore, the nonstance foot must make planar contact with the ground during impact. The mechanics of robot-ground impacts are modeled using the rigid-body impact method of [18, 38, 42], closely following the formulation in [42]. Specifically, at an impact the Euler-Lagrange equations of motion are:

$$D_e(q_e)\ddot{q}_e + H_e(q_e, \dot{q}_e) = B_e(q_e)u + \delta F_{ext}, \tag{2.16}$$

where δF_{ext} is a vector of external (impact) forces incurred by collision of the non-stance foot and the walking surface. Integrating this equation over an infinitesimally small time, yields

$$D_e(q_e^+)\dot{q}_e^+ - D_e(q_e^-)\dot{q}_e^- = F_{ext} \tag{2.17}$$

where $F_{ext} := \int_{t^-}^{t^+} \delta F_{ext}(\tau)d\tau$ is the result of integrating the impulsive contact force of the impact duration, $\dot{q}_e^-$ is the velocity just before the impact and $\dot{q}_e^+$ is the velocity just after the impact.

Attach a body-fixed frame, R_{nst} to the projection of the nonstance ankle on the nonstance foot, and let $p_{nst}(q_e)$ and $\phi_{nst}(q_e)$ be the position and orientation of R_{nst}

with respect to the world frame. At impact,

$$
\begin{bmatrix} p_{nst}^z(q_e) \\ \phi_{nst}(q_e) \end{bmatrix} = 0,
\tag{2.18}
$$

where $p_{nst}^z(q_e)$ describes the vertical position of the foot with respect to the floor. It follows from the principle of virtual work that

$$
F_{ext} = J_{nst}^T(q_e)F_{nst}
\tag{2.19}
$$

where $J_{nst}(q_e) = \frac{\partial}{\partial q_e}p_{nst}(q_e)$ and F_{nst} is a a vector of reaction forces and moments which the ground imparts on the nonstance foot. The combined set of equations yields

$$
\begin{bmatrix} D_e(q_e^-) & -J_{nst}^T(q_e) \\ J_{nst}^T(q_e) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_e^+ \\ F_{nst} \end{bmatrix} = \begin{bmatrix} D_e(q_e^-)\dot{q}_e^- \\ 0 \end{bmatrix}.
\tag{2.20}
$$

the post impact velocity and corresponding impact forces can then be obtained by solving

$$
\begin{bmatrix} \dot{q}_e^+ \\ F_{nst} \end{bmatrix} = \begin{bmatrix} D_e(q_e^-) & -J_{nst}^T(q_e) \\ J_{nst}^T(q_e) & 0 \end{bmatrix}^{-1} \begin{bmatrix} D_e(q_e^-)\dot{q}_e^- \\ 0 \end{bmatrix},
\tag{2.21}
$$

To simplify notation, the change in velocity due to impact is written:

$$
\dot{q}_e^+ = \Delta_{\dot{q}}(q_e^-)\dot{q}_e^-.
\tag{2.22}
$$

2.4 Hybrid System Model

A hybrid control system model encapsulates all components of the robot model in this section. Here, the robot is modeled as a *simple hybrid control system*:

$$\mathscr{HC}^R = (\mathcal{D}^R, U^R, S^R, \Delta^R, f^R, g^R). \tag{2.23}$$

The individual elements of this hybrid system are obtained through traditional methods, as discussed in the following paragraphs.

The domain, $\mathcal{D}^R$, specifies the allowable configuration of the system while the guard, S^R, indicates the edge of the domain, i.e. the configuration in which the robot transitions out of the given domain. The domain is specified by the list of assumptions made on the biped's gait in Section 2.1.1, in particular, the single-support domain is the set of all joint angles and velocities such that the nonstance foot is above the ground. Mathematically, this is described by a unilateral constraint function, $h^R(q)$, the height of the nonstance foot above the walking surface. ZMP constraints on the stance foot are treated as implicit admissibility conditions; only gaits in which ZMP are satisfied (via proper control design) are considered. The guard event occurs when the nonstance foot strikes the ground, i.e. the instantaneous double support phase of the assumed walking gait. The domain and guard are given by:

$$\mathcal{D}^R = \left\{ (q, \dot{q}) \in T\mathcal{Q}^R : h^R(q) \geq 0 \right\}.$$
$$S^R = \left\{ (q, \dot{q}) \in T\mathcal{Q}^R : h^R(q) = 0 \text{ and } dh^R(q)\dot{q} < 0 \right\},$$

where $dh^R(q)$ is the Jacobian of h^R at q.

In a simple hybrid system, the reset map, Δ^R, is a map from the guard to the

domain, i.e.

$$\Delta^R : S^R \to \mathcal{D}^R, \qquad \Delta^R(q, \dot{q}) = \begin{bmatrix} \Delta_q q \\ \Delta_{\dot{q}}(q)\dot{q} \end{bmatrix}, \tag{2.24}$$

where Δ_q is the relabeling matrix which switches the stance and non-stance leg at impact (by appropriately changing the angles). Here, $\Delta_{\dot{q}}$ determines the change in velocity due to impact (see [18], [13] and [3]).

The affine control system, (f^R, g^R), is a set of first order ordinary differential equations (ODEs) obtained from the equations of motion under holonomic constraints (2.12) and which can be written:

$$\dot{x} = f^R(x) + g^R(x)u \tag{2.25}$$

where

$$f^R(q, \dot{q}) = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)H(q, \dot{q}) \end{bmatrix}, \quad g^R(q) = \begin{bmatrix} \mathbf{0} \\ D^{-1}(q)B(q) \end{bmatrix},$$

with $U^R \subset \mathbb{R}^{10}$ a set of constraints on the admissible joint torques, i.e. actuator torque limitations that are obtained from the robot hardware specification sheets. The control system describes the continuous-time evolution of the robot model and is paramount in the construction of control laws for walking, as discussed in the next section.

3. WALKING CONTROLLER CONSTRUCTION

The Human-Inspired Optimization (HIO) is a method for *rapidly* generating walking controllers for a robot model; the HIO provides a torque control law, u, and fixed point, x^*, corresponding to a stable periodic orbit in the state-space of the hybrid system model of the robot (2.23). The method leverages insight obtained from human walking data; specifically that human walking data suggest bipedal human walking can be represented by a set of kinematic output "behaviors" which follow a very simple function. These simple behaviors found in human walking data motivate the design of parameterized control objectives or *outputs* which are imposed on the robot model through Feedback Linearization [35]. Values of the control parameters are obtained through the solution of a constrained nonlinear optimization problem which minimizes the least-squares error of robot and human output functions while satisfying physical constraints, e.g. actuator torque limits, and ensuring that the resulting hybrid system has a stable periodic orbit. That is, after solving an optimization problem, we obtain parameters for control laws, which we apply to the robot control system and achieve periodic walking which, through minimizing the cost to human data, looks remarkably human-like. This section describes the Human-Inspired Optimization approach for obtaining single-domain walking in a 10 degree of freedom (DOF), 3D robot model.

3.1 Insights from Human Walking Data

Human walking can be represented, with high correlation, as a set of simple behaviors; this revelation by Dr. Ames in the summer of 2011 was the genesis of the Human-Inspired Optimization method [3]. The main idea behind this method is that the simple behaviors humans display in walking can be used as suggestions for the construction of robot behaviors (outputs) for robot walking controllers. Specific sets of these outputs are sought which seem to encode certain fundamental kinematics behaviors present in human walking. This section describes the processing and inspection of human walking data for the purpose of furthering our understanding of bipedal locomotion.

3.1.1 Data Collection

Human walking data were obtained in a set of experiments conducted by Dr. Bajcsy at the University of California, Berkeley [40]. In these experiments, test subjects walked forward along a straight line and on a level floor. The goal of these experiments was to track the evolution of the spatial positions of specific points on the human body during walking on flat ground. For each trial in the experiments, LED sensors were fixed to a test subject in key locations, such as the joints, along the lower body. As the test subject walked forward, the spatial XYZ position (with respect to a fixed reference frame) of each LED sensor was measured using the Phase Space [27] system – which measures the position of LED sensors with a 1 millimeter accuracy at a sampling rate of 480 Hz. The relevant data used from these experiments is the mean data from 11 trials per subject for 9 subjects. Figure 3.1a is a representation of the LED sensor layout on a given test subject - with RED circles indicating LED sensor placement at the joints of the lower body.

22

(a) LED Placement

(b) Linearized Hip Position Data vs Time
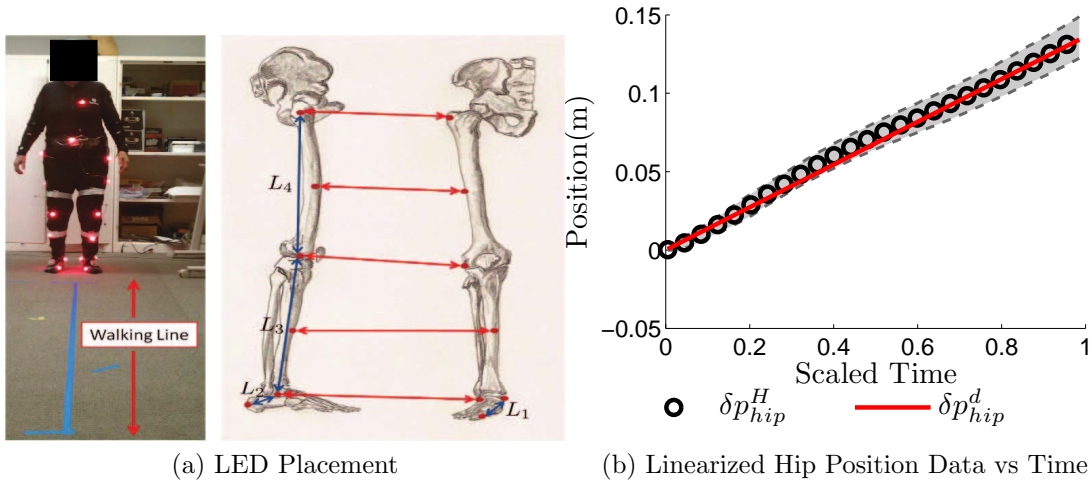
Figure 3.1: Visualization of the LED sensor placement used on test subjects during the walking experiments (left) and a fit of (3.2) to the mean data for the forward position of the hip vs time (right).

3.1.2 Data Analysis

Consider the data corresponding to a single trial in the experiments. Let $T \in \mathbb{R}$ denote the total time elapsed during the trial, and $K \in \mathbb{N}$ denote the total number of samples collected. Let $t[k]$ be the time associated with a discrete sampling event k during the trial, with $k \in \{1, 2, ..., K\}$ and where $t[1] = 0$ and $t[k] < t[k+1]$. The corresponding position of an LED, ℓ, at a sampling event time $t[k]$ is given by $p_\ell(t[k]) = (p_\ell^x(t[k]), p_\ell^y(t[k]), p_\ell^z(t[k]))$, with $\ell \in \{sh, st, sa, sk, ship, nship, nsk, nsa, nst, nsh\}$. These LED position data are used to compute various kinematic quantities, or *outputs*, of the human lower body at each sampled point in time. For example, the "nonstance slope output", $m_{nsl}(t[k])$ is calculated via

$$m_{nsl}(t[k]) := \frac{p_{nship}^x(t[k]) - p_{nsa}^x(t[k])}{p_{nship}^z(t[k]) - p_{nsa}^z(t[k])}. \tag{3.1}$$

23

The nonstance slope represents a virtual line segment from the test subject's nonstance hip to the nonstance ankle. Examination of human walking data over a single-step interval indicates that this output seems to perform a vital role in walking – an insight that will be further expounded upon in the following sections.

Another observation from the walking data is that all test subjects in the aforementioned experiments displayed a linear relationship in "forward" position[1] of the hip data with respect to time. Specifically, all test subject's x-position of the hip data can be fit with high correlation via the following function:

$$p_{hip}^{x}(t[k]) = v_{hip}^{x}t[k] + p_{hip}^{x}(t[1]), \tag{3.2}$$

with $v_{hip}^{x} \in R$ a positive constant. This data is shown in Figure 3.1b.

The nonstance slope, forward position of the hip and other kinematic maps on the data are termed "human walking outputs" as they represent the kinematic outputs of the human locomotion control system. The motivation of studying these outputs is that they provide a representation of the mechanics of human walking which abstracts away the complexity of the human locomotor system. The goal is to identify key outputs associated with successful (human) bipedal locomotion, which will ultimately provide suggestions on control design for successful (robot) bipedal locomotion.

3.1.3 Simple Behaviors Observed in Human Walking

After further investigation, Dr. Ames found that the data for several human walking outputs could all be represented by a single candidate function termed the

---

[1]Note that x is the "forward" direction for the choice of coordinate system in this thesis, see Figure 2.1.

canonical human walking function:

$$y_H(t, \alpha) = e^{-\alpha_4 t}(\alpha_1 \cos(\alpha_2 t) + \alpha_3 \sin(\alpha_2 t)) + \alpha_5, \qquad (3.3)$$

which represents the solution to an under-damped mass-spring-damper system. This becomes apparent by noting that $\alpha_1 = c_0$, $\alpha_2 = \omega_d$, $\alpha_3 = c_1$, $\alpha_4 = \zeta \omega_n$ and $\alpha_5 = \hat{g}$, where ζ is the damping ratio, ω_n is the natural frequency, $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ is the damped natural frequency, c_0 and c_1 are determined by the initial conditions of the system, and $\hat{g}$ is a gravity-related constant.

The human walking outputs relevant to this thesis are given in Table 3.1. The linearized position of the hip and nonstance slope, δp_{hip} and δm_{nsl}, are given by:

$$\delta p_{\text{hip}}(\theta) = L_c(-\theta_{sa}) + L_t(-\theta_{sa} - \theta_{sk}) \qquad (3.4)$$

$$\delta m_{nsl}(\theta) = -\theta_{sa} - \theta_{sk} - \theta_{sh} - \theta_{nsh} + \frac{L_c}{L_c + L_t}\theta_{nsk} \qquad (3.5)$$

where L_c and L_t are the lengths of the calf and thigh, respectively. The degree to which the human canonical walking function (3.3) can represent the human walking data is determined by a least squares fit. Represent the mean human output

Table 3.1: Human walking outputs.

Human Output	Description	Canonical Function
δp_{hip}	the linearized x-position of the hip	$\delta p_{\text{hip}}^d(t, v) = v_{hip}t$
δm_{nsl}	the linearized slope of the non-stance leg	$\delta m_{nsl}^d(t, \alpha_{nsl}) = y_H(t, \alpha_{nsl})$
θ_{sk}	the angle of the stance knee	$\theta_{sk}^d(t, \alpha_{sk}) = y_H(t, \alpha_{sk})$
θ_{nsk}	the angle of the nonstance knee	$\theta_{nsk}^d(t, \alpha_{nsk}) = y_H(t, \alpha_{nsk})$
θ_{tor}	the absolute sagittal angle of the torso	$\theta_{tor}^d(t, \alpha_{tor}) = y_H(t, \alpha_{tor})$
φ_{sa}	the coronal angle of the stance ankle	$\varphi_{sa}^d(t, \alpha_{sa}) = y_H(t, \alpha_{sa})$
φ_{sh}	the coronal angle of the stance hip	$\varphi_{sh}^d(t, \alpha_{sh}) = y_H(t, \alpha_{sh})$
φ_{nsh}	the coronal angle of the nonstance hip	$\varphi_{nsh}^d(t, \alpha_{nsh}) = y_H(t, \alpha_{nsh})$
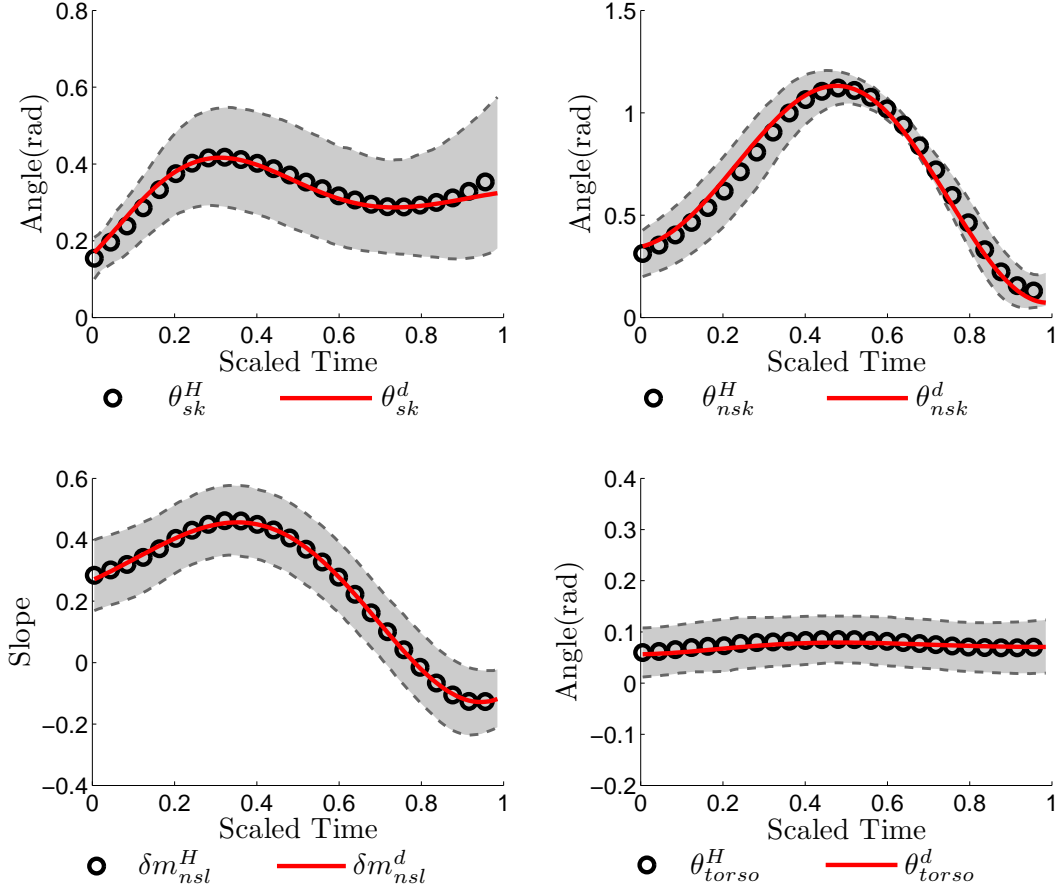
Figure 3.2: "Walking Outputs", as computed on experimental human locomotion data, are shown versus (normalized) time for one continuous phase of walking. The corresponding least-squares fits using the canonical walking function are displayed in red.

data by $y_i^H[k]$ and the canonical walking functions by $y_i^d(t, \alpha_i)$ for $i \in$ Output $=$ $\{nsl, sk, nsk, tor, sa, sh, nsh\}$; for example, $y_{sa}^H[k] = \varphi_{sa}^H[k]$ and $y_{sa}^d(t, \alpha_{sa}) = \varphi_{sa}^d(t, \alpha_{sa})$. Define the following human-data-based cost function:

$$\text{Cost}_{\text{HD}}(\alpha) = \sum_{k=1}^{K} \sum_{i \in \text{Output}} \left(y_i^H[k] - y_i^d(t^H[k], \alpha_i) \right)^2 \tag{3.6}$$

which is simply the sum of squared residuals. To determine the parameters for the

26

human walking functions, we need only solve the optimization problem:

$$\alpha^* = \underset{\alpha \in \mathbb{R}^{36}}{\operatorname{argmin}} \ \mathrm{Cost}_{\mathrm{HD}}(\alpha) \qquad (3.7)$$

which yields the least squares fit of the mean human output data with the canonical walking functions. The parameters given by solving this optimization problem are stated in Table 3.2. The correlations, as given in the same table, show that the fitted walking functions very closely model the human output data, i.e., the chosen human walking functions appear to be, in fact, canonical. Indeed, the coefficients of correlation are all very high, ranging from 0.8767 to 0.9997. These output functions are used in the development of robotic walking controllers as discussed in the next section.

Table 3.2: Parameter values of the canonical walking function obtained via a least squares fit to the experimental human data.

$y_1^d = v_{\mathrm{hip}}t, \quad y_2^d = y_H(t, \alpha)$ given in (3.3)							
f.	v_{hip}	a_1	a_2	a_3	a_4	a_5	Corr.
δp_{hip}	0.2288	—	—	—	—	—	0.9984
δm_{nsl}	—	-0.0065	8.9157	0.1162	-2.2638	0.2750	0.9997
θ_{sk}	—	-0.1600	12.4473	0.0980	3.6061	0.3240	0.9751
θ_{nsk}	—	-0.3322	-10.2168	-0.1109	-0.9345	0.6772	0.9948
θ_{tor}	—	-0.0166	10.4416	-0.0033	3.2976	0.0729	0.8767
φ_{sa}	—	0.0543	-0.0000	0.0137	4.0740	-0.2686	0.9952
φ_{sh}	—	0.0543	-0.0000	0.0137	4.0740	-0.2686	0.9952
φ_{nsh}	—	0.0170	1.1655	-0.0174	-5.9435	-0.2615	0.9896

3.2   Controller Design

Motivated by the desire to obtain human-like, bipedal robotic locomotion, the goal is to construct a controller which drives outputs of the robot to the "human walking outputs" discussed in the previous section. This goal is effected formally through a control law $u : T\mathcal{Q}^R \to U^R$ which guarantees that $y^a(q(t)) \to y^d(t)$ exponentially as $t \to \infty$, where $y^a : \mathcal{Q}^R \to \mathbb{R}^{10}$ is a vector of kinematics maps on the robot representing the human outputs and $y^d : \mathbb{R} \to \mathbb{R}^{10}$ is a vector of canonical human functions. As the dynamics of the robot model are highly nonlinear, a natural choice of control method for this system is Input/Output Linearization [35].

The affine control system for the hybrid system model of our robot, (2.25), can be written as a Multi-Input Multi-Output (MIMO) system of the form, which

$$\dot{x} = f^R(x) + g^R(x)u \tag{3.8}$$

$$y = y^a(x) - y^d(x) \tag{3.9}$$

with $x \in \mathbb{R}^{2n}$, $y \in \mathbb{R}^m$ and $u \in \mathbb{R}^m$ (recall that $n=m$ are the number of actuated degrees of freedom in our model). This system is called "square" as the number of inputs equals the number of outputs. As discussed above, the objective is to design a $u(x)$ that ensures $y \to 0$ exponentially fast, resulting in convergence of actual robot outputs , $y^a(x)$, to corresponding desired values, $y^d(x)$. The following section describes how the actual and desired output functions are designed from the conclusions drawn on human locomotion data. Parameters for the desired output functions will be determined via optimization, as discussed later in this section.

3.2.1 Actual and Desired Robot Outputs

The construction of this control law uses the human walking functions considered in Section 3.1.3. With the goal of controlling the velocity of the robot, define the relative degree 1 actual output as the velocity of the hip and define the the desired velocity of the hip:

$$y_1^a(q, \dot{q}) = \delta \dot{p}_{\text{hip}}^R(q, \dot{q}) = d\delta p_{\text{hip}}^R(q)\dot{q}, \qquad y_1^d = v_{\text{hip}}. \tag{3.10}$$

Furthermore, define the linear (relative degree 2) actual outputs of the robot to be the output functions considered in Section 3.1.3 and the desired outputs to be the corresponding outputs of the human as represented by the walking functions:

$$y_{2,L}^a(q) = \begin{bmatrix} \delta m_{nsl}^R(q) \\ \theta_{sk} \\ \theta_{nsk} \\ \theta_{tor}^R(q) \\ \varphi_{sa} \\ \varphi_{sh} \\ \varphi_{nsh} \end{bmatrix}, \quad y_{2,L}^d(t, \alpha) = \begin{bmatrix} y_H(t, \alpha_{nsl}) \\ y_H(t, \alpha_{sk}) \\ y_H(t, \alpha_{nsk}) \\ y_H(t, \alpha_{tor}) \\ y_H(t, \alpha_{sa}) \\ y_H(t, \alpha_{sh}) \\ y_H(t, \alpha_{nsh}) \end{bmatrix}, \tag{3.11}$$

where $y_{2,L}^a : \mathcal{Q}^R \to \mathbb{R}^7$ are the actual linear outputs of the robot, and $y_{2,L}^d : \mathbb{R} \times \mathbb{R}^{35} \to \mathbb{R}^7$ are the desired functions for these linear outputs. The actual linear outputs of the robot, $y_{2,L}^a(q)$, are carefully chosen *to only depend on the first eight angles of the system.* That is, define $q_{1:8}$ by:

$$q_{1:8} = \{\varphi_{sa}, \theta_{sa}, \theta_{sk}, \theta_{sh}, \varphi_{sh}, \varphi_{nsh}, \theta_{nsh}, \theta_{nsk}\} \subset q, \tag{3.12}$$

29

which are all angles *except* the nonstance ankle angles. Due to the linear form of the outputs considered, they can be written as:

$$y_{2,L}^a(q) = H\, q_{1:8} \tag{3.13}$$

for $H \in \mathbb{R}^{7\times 8}$ with full row rank (where, for example, the top row of H is obtained by taking the Jacobian of (3.5)). This linear form is essential in the construction of the optimization problem of Section 3.3 and is only applicable to the first eight angles of the system. To complete the set of controller outputs, the final nonlinear relative degree two outputs from Section 2.2.3 are needed:

$$y_{2,N}^a(q) = \begin{bmatrix} \psi_x^R \\ \psi_y^R \end{bmatrix}, \quad y_{2,N}^d(t, \alpha) = \begin{bmatrix} y_H(t, \alpha_{\psi x}) \\ y_H(t, \alpha_{\psi y}) \end{bmatrix}. \tag{3.14}$$

Note that $y_{2,N}^a(q)$ *depend on all ten angles of the system.* Grouping the linear and nonlinear relative degree two outputs results in:

$$y_2^a(q) = \begin{bmatrix} y_{2,L}^a(q) \\ y_{2,N}^a(q) \end{bmatrix}, \quad y_2^d(t, \alpha) = \begin{bmatrix} y_{2,L}^d(t, \alpha) \\ y_{2,N}^d(t, \alpha) \end{bmatrix}. \tag{3.15}$$

The parameters of all of the outputs can be combined to yield a single vector $\alpha \in \mathbb{R}^{46}$.

3.2.2 Parameterization of Time

The goal is for the outputs of the robot to agree with the outputs of the human, motivating the final form of the outputs to be used in feedback linearization:

$$y_1(q, \dot{q}, \alpha) = y_1^a(q, \dot{q}) - v_{hip}, \tag{3.16}$$

$$y_2(q, \alpha) = y_2^a(q) - y_2^d(\tau(q), \alpha), \tag{3.17}$$

30

where

$$\tau(q) = \frac{\delta p_{\text{hip}}^R(q) - \delta p_{\text{hip}}^R(q^+)}{v_{\text{hip}}}, \tag{3.18}$$

is a state-based parameterization of time with $\delta p_{\text{hip}}^R(q^+)$ the linearized position of the hip of the robot at the beginning of a step. This parameterization is important as it allows for control over walking speed through the parameter v_{hip}.

3.2.3   Torque Controller via Feedback Linearization

Feedback linearization is used to obtain a linear input-output relationship in the dynamics of the outputs. Specifically, the first time-derivative of the relative degree one outputs yields a linear relationship between $\dot{y}_1$ and u:

$$\dot{y}_1 = \frac{\partial y_1}{x} f^R(x) + \frac{\partial y_1}{x} g^R(x)u \tag{3.19}$$

$$:= L_{f^R} y_1(x) + L_{g^R} y_1(x)u \tag{3.20}$$

where $L_{f^R} y_1(x) = \frac{\partial y_1}{x} f^R(x)$ and $L_{g^R} y_1(x) = \frac{\partial y_1}{x} g^R(x)$ are called the *Lie derivatives* of y_1 with respect to f^R and g^R, respectively. Note that as y_1 is a function of the joint velocities, $L_{g^R} y_1(x) \neq 0$, and hence, it is an output of vector relative degree one [35].

The relative degree two outputs are differentiated twice to yield a linear relationship between $\ddot{y}_2$ and u, with the first derivative calculated via:

$$\dot{y}_2 = \frac{\partial y_2}{x} f^R(x) + \frac{\partial y_2}{x} g^R(x)u \tag{3.21}$$

$$:= L_{f^R} y_2(x) + \underbrace{L_{g^R} y_2(x)u}_{0}, \tag{3.22}$$

31

and the second derivative given by

$$\ddot{y}_2 = \frac{\partial L_{f^R} y_2}{x} f^R(x) + \frac{\partial L_{f^R} y_2}{x} g^R(x) u \tag{3.23}$$

$$:= L_{f^R}^2 y_2(x) + L_{g^R} L_{f^R} y_2(x) u. \tag{3.24}$$

These outputs can be used to define a *human-inspired controller*:

$$u_{\alpha,\varepsilon}(q, \dot{q}) = \tag{3.25}$$

$$-\begin{bmatrix} L_{g^R} y_1(q) \\ L_{g^R} L_{f^R} y_2(q) \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 \\ L_{f^R}^2 y_2(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} L_{f^R} y_1(q, \dot{q}) \\ 2\varepsilon L_{f^R} y_2(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} \varepsilon y_1(q, \dot{q}) \\ \varepsilon^2 y_2(q) \end{bmatrix} \right),$$

with control gain ε. The pre-multiplied matrix is known as the decoupling matrix and is non-singular because of the careful choice of output functions, i.e., as discussed in Section 3.1.3. This control law results in the following dynamics on the outputs:

$$\dot{y}_1 = -\varepsilon y_1 \tag{3.26}$$

$$\ddot{y}_2 = -2\varepsilon \dot{y}_2 - \varepsilon^2 y_2$$

and it follows that for a control gain $\varepsilon > 0$, the control law $u_{\alpha,\varepsilon} : T\mathcal{Q}^R \times \mathbb{R}^{46} \times \mathbb{R}^+ \to U^R$ renders the output exponentially stable [35]. That is, the human-inspired output $y \to 0$ exponentially at a rate of ε as $t \to \infty$; in other words, the outputs of the robot will converge to the canonical human walking functions exponentially quickly.

For the hybrid control system $\mathscr{HC}^R$, the human-inspired control law is applied to obtain the hybrid system

$$\mathscr{H}_{\alpha,\varepsilon}^R = (\mathcal{D}^R, S^R, \Delta^R, f_{\alpha,\varepsilon}^R) \tag{3.27}$$

32

with

$$f_{\alpha,\varepsilon}^R(q, \dot{q}) = f^R(q, \dot{q}) + g^R(q, \dot{q})u_{\alpha,\varepsilon}(q, \dot{q}). \tag{3.28}$$

The end result of the modeling process is a hybrid system $\mathscr{H}_{\alpha,\varepsilon}^R$ that depends on the parameters of the human inspired control α and ε.

3.2.3.1 Zero Dynamics

For the *continuous* dynamics of the hybrid system $\mathscr{H}_{(\alpha,\varepsilon)}^R$, the controller renders the *full zero dynamics surface*

$$\mathbf{FZ}_\alpha = \left\{ (q, \dot{q}) \in T\mathcal{Q}^R : y_1(q, \alpha) = 0, \; y_2(q, \dot{q}, \alpha) = \mathbf{0}_9, \; L_{f^R}y_2(q, \dot{q}, \alpha) = \mathbf{0}_9 \right\} \tag{3.29}$$

exponentially stable (where $\mathbf{0}_n$ is a vector of n zeros). However, enforcing invariance of this surface through the discrete impacts would be too strong of a requirement since it would force the system to evolve on a 1-dimensional manifold. Therefore, hybrid invariance is enforced only for the relative degree 2 outputs. The corresponding surface is referred to as the *partial zero dynamics surface*

$$\mathbf{PZ}_\alpha = \left\{ (q, \dot{q}) \in T\mathcal{Q}^R : y_2(q, \alpha) = \mathbf{0}_9, \; L_{f^R}y_2(q, \dot{q}, \alpha) = \mathbf{0}_9 \right\}. \tag{3.30}$$

Since the only output that is not included in the partial zero dynamics surface is the forward velocity of the hip, enforcing partial hybrid zero dynamics, rather than full hybrid zero dynamics, means that the velocity of the hip is allowed to drift at impact; relaxation of the forward velocity output helps mitigate the impulsive effects induced on the control system through foot-ground impact.

3.3 Human Inspired Walking Controller Optimization

This section presents the main result of this paper: an optimization problem which yields parameters for the human-inspired controller, $u_{\alpha,\varepsilon}$, that minimize a human data-based cost function [5] while simultaneously yielding robotic walking in simulation and satisfying physical constraints of the actual robot hardware. A novel method is presented for computing these constraints in closed form (rather than explicitly integrating the dynamics) through the interplay between full and partial hybrid zero dynamics.

3.3.1 Objective Function

The objective, or cost, function of the proposed optimization is the human-data based cost function, (3.6), used in Section 3.1.3 to correlate the canonical human walking function with experimental walking data. Recall that this cost function is the sum of least squares fits of desired outputs, described by (3.3), to the corresponding mean human data. Stated again for reference, the human-data based cost (3.6) is computed via

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

The goal is to minimize this function; as $\text{Cost}_{\text{HD}}(\alpha) \to 0$, the desired robot behavior becomes increasingly similar to human behavior – on the level of walking outputs. Differences in morphology between robot and human locomotor systems – and the rigorous notion of walking considered here – make it very unlikely that an exact fit, $\text{Cost}_{\text{HD}}(\alpha) = 0$, would result in robot walking. That is to say, we expect that to satisfy constraints on robotic walking, as discussed in the following section, the optimized robot outputs should slightly deviate from the human output data.

3.3.2   Equality and Inequality Constraints

Several constraints must be enforced on the optimization problem to ensure that the optimized controller parameters yield stable walking in simulation *and* satisfy the physical modeling limitations of the actual robot hardware, e.g. balanced ground reaction forces and motor performance specifications. These constraints, however, require computation of the robot's state over the course of one step. This could be done through numeric integration, but this would be computationally expensive. Therefore, a method of computing the robot's state, $(q^s, \dot{q}^s)$, as a function of t and α is first presented.

3.3.2.1   Partial Hybrid Zero Dynamics Constraints

Following from [3, 4], to compute the constraints needed to ensure *partial hybrid zero dynamics*, the outputs and guard functions are used to explicitly solve for the configuration of the system $\vartheta(\alpha) \in \mathcal{Q}^R$ on the guard $(h^R(\vartheta(\alpha)) = 0)$ in terms of the parameters α. In particular, let

$$\vartheta(\alpha) = q \quad \text{s.t.} \quad y_2(\Delta_q q) = \mathbf{0}_9 \ \text{ and } \ h^R(q) = 0 \tag{3.31}$$

where Δ_q is the relabeling matrix (2.24). Note that multiple solutions to $\vartheta(\alpha)$ exist because $y_{2,N}^d(\Delta_q q)$ and $h^R(q)$ are nonlinear functions of all joint angles; however, restrictions are placed on $\vartheta(\alpha)$ such that only one solution corresponds to a valid configuration. Using $\vartheta(\alpha)$ allows for the explicit solution of a point $(\vartheta(\alpha), \dot{\vartheta}(\alpha)) \in \mathbf{FZ}_\alpha \cap S^R$. In particular, let

$$Y(q) = \begin{bmatrix} d\delta p_{\text{hip}}^R(q) \\ dy_2(q) \end{bmatrix}. \tag{3.32}$$

It follows from the definition of y_1 and y_2 that

$$\begin{bmatrix} y_1(q,\dot{q}) \\ L_{f^R} y_2(q,\dot{q}) \end{bmatrix} = Y(q)\dot{q} - \begin{bmatrix} v_{\text{hip}} \\ \mathbf{0}_9 \end{bmatrix}. \tag{3.33}$$

Therefore, define

$$\dot{\vartheta}(\alpha) = Y^{-1}(\vartheta(\alpha)) \begin{bmatrix} v_{\text{hip}} \\ \mathbf{0}_9 \end{bmatrix}, \tag{3.34}$$

where Y is invertible because of the choice of outputs. Finally, the constraints needed for partial hybrid zero dynamics can be written:

$$y_2(\vartheta(\alpha)) = \mathbf{0}_9, \tag{3.35}$$

$$dy_2(\Delta_q \vartheta(\alpha))\Delta_{\dot{q}}(\vartheta(\alpha))\dot{\vartheta}(\alpha) = \mathbf{0}_9, \tag{3.36}$$

$$dh^R(\vartheta(\alpha))\dot{\vartheta}(\alpha) < 0. \tag{3.37}$$

3.3.2.2 Computing solutions: $q^s(t,\alpha)$ and $\dot{q}^s(t,\alpha)$ on a PHZD surface

This section utilizes the fact that the human outputs were specifically chosen to be linear in order to explicitly construct the *partial hybrid zero dynamics*. Because of the specific choice of $y_{2,L}^a$, the following representation of the partial zero dynamic coordinates is employed:

$$\xi_1 = \delta p_{\text{hip}}^R(q) =: c\, q_{1:8}, \tag{3.38}$$

$$\xi_2 = y_1^a(q,\dot{q}) = \delta \dot{p}_{\text{hip}}^R(q,\dot{q}) =: c\, \dot{q}_{1:8}.$$

where $c \in \mathbb{R}^{1 \times 8}$ is obtained from (3.4) and recalling that $q_{1:8} \subset q$ is a vector of all angles of the system except the nonstance ankle angles. By (3.26), the partial zero dynamics is therefore given by:

$$\dot{\xi}_1 = \xi_2, \tag{3.39}$$

$$\dot{\xi}_2 = -\varepsilon(\xi_2 - v_{\text{hip}}). \tag{3.40}$$

Moreover, since ξ_1 is just the linearized position of the hip, which is used to parameterize time, the desired relative degree two outputs can be written $y_{2,L}^d(\tau(q), \alpha) = y_{2,L}^d(\xi_1, \alpha)$. Integrating (3.40) yields a closed form expression for the actual velocity of the hip at some time t after impact:

$$\xi_2(t) = \int_{0^+}^t -\varepsilon(\xi_2(\tau) - v_{\text{hip}})d\tau = v_{\text{hip}} + (\xi_2(0^+) - v_{\text{hip}})e^{-\varepsilon t} \tag{3.41}$$

and integrating once more gives the actual position of the hip:

$$\xi_1(t) = \int_{0^+}^t v_{\text{hip}} + (\xi_2(0^+) - v_{\text{hip}})e^{-\varepsilon \tau}d\tau \tag{3.42}$$

$$= \xi_1(0^+) + v_{\text{hip}}t + \frac{1}{\varepsilon}(\xi_2(0^+) - v_{\text{hip}})(1 - e^{-\varepsilon t}) \tag{3.43}$$

Using the solution to the inverse kinematics problem, (3.31), an exact expression of the evolution of the position and velocity of the hip are given by

$$\xi_1^s(t) \quad := \quad \xi_1(\Delta_q \vartheta(\alpha)) + v_{\text{hip}}t + \frac{1}{\varepsilon}(\xi_2(\Delta_q \vartheta(\alpha), \Delta_{\dot{q}}(\vartheta(\alpha))\dot{\vartheta}(\alpha)) - v_{\text{hip}})(1 - e^{-\varepsilon t}),$$

$$\xi_2^s(t) \quad := \quad v_{\text{hip}} + (\xi_2(\Delta_q \vartheta(\alpha)) - v_{\text{hip}})e^{-\varepsilon t}. \tag{3.44}$$

These expressions can be used in the partial zero dynamics surface to obtain an exact solution for the full-order system by picking the coordinates

$$\eta_1 = y_{2,L}^a(q) = H\,q_{1:8}, \tag{3.45}$$

$$\eta_2 = L_{f^R}y_{2,L}^a(q,\dot{q}) = H\,\dot{q}_{1:8},$$

with $H \in \mathbb{R}^{7\times 8}$ as in (3.13), and defining

$$\Phi(\xi_1,\alpha) = \begin{bmatrix} c \\ H \end{bmatrix}_{8\times 8}^{-1} \begin{pmatrix} \xi_1 \\ y_{2,L}^d(\xi_1,\alpha) \end{pmatrix}, \tag{3.46}$$

$$\Psi(\xi_1,\alpha) = \begin{bmatrix} c \\ H \end{bmatrix}_{8\times 8}^{-1} \begin{pmatrix} 1 \\ \frac{\partial y_{2,L}^d(\xi_1,\alpha)}{\partial \xi_1} \end{pmatrix}. \tag{3.47}$$

These yield solutions of the first eight angles and corresponding velocities of the system:

$$q_{1:8}^s(t,\alpha) = \Phi(\xi_1^s(t),\alpha), \tag{3.48}$$

$$\dot{q}_{1:8}^s(t,\alpha) = \Psi(\xi_1^s(t),\alpha)\xi_2^s(t). \tag{3.49}$$

Denote the remaining two angles of the system by $q_{9:10} := \{\theta_{nsa}, \varphi_{nsa}\} \subset q$, such that $(q_{1:8} \cup q_{9:10}) = q$. Solutions of the final four states of the system $q_{9:10}^s$ and $\dot{q}_{9:10}^s$ are computed under the assumption that the nonstance foot remains flat throughout the gait (achieved via control). Specifically, let $R_0^{nsf}(q)$ be a rotation matrix describing the orientation of the nonstance foot in the world frame. Note that $R_0^{nsf}(q)$ can be represented as the product of intermediate rotation matrices, $R_0^{nsa}(q_{1:8})$ which describes the orientation of a frame attached to the nonstance ankle with respect

to the world frame, and $R_{nsa}^{nsf}(q_{9:10})$ which describes the orientation of the nonstance foot frame with respect to the nonstance ankle frame:

$$R_0^{nsf}(q) = R_0^{nsa}(q_{1:8})R_{nsa}^{nsf}(q_{9:10}), \tag{3.50}$$

where $R_0^{nsa}(q_{1:8}^s)$ is known via the solution to (3.48) for the first eight angles of the system. To solve for the last two angles of the system, the following equation will be used:

$$R_{nsa}^{nsf}(q_{9:10}) = [R_0^{nsa}(q_{1:8})]^{-1} R_0^{nsf}(q), \tag{3.51}$$

and this will be accomplished by leveraging standard properties of rotation matrices to extract an equation for θ_{sa} and an equation for φ_{sa}. Specifically, $R_0^{nsf}(q)$ – and any rotation matrix – can be also represented by a set of three Euler angles [36], $\theta_X, \theta_Y, \theta_Z$:

$$R_0^{nsf}(q) = R(\theta_X)R(\theta_Y)R(\theta_Z), \tag{3.52}$$

where $\theta_X, \theta_Y, \theta_Z$ describe rotations about the world frame X, Y, Z axes, respectively. Due to the flat nonstance foot assumption, $\theta_X = \theta_Y = 0$, and thus, $R_0^{nsf}(q)$ can be described by a matrix whose elements are functions of θ_Z:

$$R_0^{nsf}(q) = \begin{bmatrix} \cos(\theta_Z) & -\sin(\theta_Z) & 0 \\ \sin(\theta_Z) & \cos(\theta_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.53}$$

Additionally, we know – by definition of the robot joint axes in Section 2.1.2 – that $R_{nsa}^{nsf}(q_{9:10})$ is a rotation matrix whose elements are given by:

$$R_{nsa}^{nsf}(q_{9:10}) = \begin{bmatrix} \cos(\theta_{sa}) & \sin(\theta_{sa})\sin(\varphi_{sa}) & \sin(\theta_{sa})\cos(\varphi_{sa}) \\ 0 & \cos(\varphi_{sa}) & -\sin(\varphi_{sa}) \\ -\sin(\theta_{sa}) & \cos(\theta_{sa})\sin(\varphi_{sa}) & \cos(\theta_{sa})\cos(\varphi_{sa}) \end{bmatrix} \quad (3.54)$$

With these properties, we are now ready to solve for θ_{sa} and φ_{sa}. Denote the element in the ith row of the jth column in $[R_0^{nsa}(q_{1:8})]^{-1}$ by r_{ij}, and use (3.53) so that (3.51) can be rewritten:

$$R_{nsa}^{nsf}(q_{9:10}) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} \cos(\theta_Z) & -\sin(\theta_Z) & 0 \\ \sin(\theta_Z) & \cos(\theta_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.55)$$

where the value of each r_{ij} is known through $q_{1:8}^s$. Equating the third columns of (3.54) and (3.55) yields

$$\begin{bmatrix} \sin(\theta_{sa})\cos(\varphi_{sa}) \\ -\sin(\varphi_{sa}) \\ \cos(\theta_{sa})\cos(\varphi_{sa}) \end{bmatrix} = \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix}, \quad (3.56)$$

from which we obtain equations for the final two angles of the system, θ_{sa} and φ_{sa}:

$$\theta_{sa}(t, \alpha) = \tan^{-1}\left(\frac{r_{13}}{r_{33}}\right) \quad (3.57)$$

$$\varphi_{sa}(t, \alpha) = \tan^{-1}\left(-\frac{r_{23}\cos(\theta_{sa})}{r_{33}}\right). \quad (3.58)$$

It follows that $(q^s, \dot{q}^s) \in \mathbf{PZ}_\alpha$.

3.3.2.3   Model Constraints

To ensure that the stance foot remains flat as the human-inspired controller is applied to the robot, the ground reaction inequalities from Section 2.2.3 are computed over the course of one step through the approximation to the solution. Specifically, the elements of F_{st} are all functions of the torque provided to the system. Therefore, equations (2.14)-(2.15) can be rearranged and stated in terms of inequalities of the form $C_i(u) < 0$ for $i \in \{1, \dots, 4\}$. Moreover, using the solution to the robot states, $(q^s(t, \alpha), \dot{q}^s(t, \alpha))$, the torque is computed at each time, t, over the course of a step:

$$u^s_{\alpha,\varepsilon}(t) := u_{\alpha,\varepsilon}(q^s(t, \alpha), \dot{q}^s(t, \alpha)). \tag{3.59}$$

Therefore, the ZMP constraints on the stance foot can be stated as the constraint:

$$\max_{i \in \{1, \dots, 4\}} \max_{t \in [0, \tau(\vartheta(\alpha))]} C_i(u^s_{\alpha,\varepsilon}(t)) < 0, \tag{3.60}$$

where $\tau(\vartheta(\alpha))$ is the duration of a step.

The nonstance foot is controlled to be parallel to the ground through the following constraints on α:

$$\alpha_{\psi x} = \mathbf{0}_{1 \times 5}, \qquad \alpha_{\psi y} = \mathbf{0}_{1 \times 5}. \tag{3.61}$$

These constraints on $\alpha_{\psi x}$ and $\alpha_{\psi y}$ reduce the size of the optimization search space to $\mathbb{R}^{36}$. Through input/output linearization and this choice of parameters, the orientation of the nonstance foot is exponentially driven to be parallel with respect to the ground plane.

3.3.3 Optimization Problem Statement

The goal of *human-inspired PHZD optimization* is to find parameters α^* which solve the following constrained optimization problem:

$$\alpha^* = \underset{\alpha \in \mathbb{R}^{46}}{\operatorname{argmin}} \ \mathrm{Cost}_{\mathrm{HD}}(\alpha) \tag{3.62}$$

$$\text{s.t.} \quad (3.35), (3.36), (3.37), (3.60), (3.61)$$

with $\mathrm{Cost}_{\mathrm{HD}}$ the cost given in (3.6). The main result of this paper is established by combining the constructions and results of this section with Theorem 2 of [4]. It particular, it establishes that solving this optimization problem results in a exponentially stable periodic orbit for $\mathscr{H}^R_{\alpha^*,\varepsilon}$.

Theorem 1 *Let α^* be parameters solving (3.62). If $\tau(\vartheta(\alpha)) > 0$ then there exists a constant $\bar{\varepsilon} > 0$ such that for all $\varepsilon > \bar{\varepsilon}$ the hybrid system $\mathscr{H}^R_{\alpha^*,\varepsilon}$ has an exponentially stable periodic orbit. Moreover, for $(q^*_\varepsilon, \dot{q}^*_\varepsilon)$ the fixed point of this orbit, and T_ε the period, the following properties are satisfied:*

$$\lim_{\varepsilon \to \infty} T_\varepsilon = \tau(\vartheta(\alpha^*)) \tag{3.63}$$

$$\lim_{\varepsilon \to \infty} (q^*_\varepsilon, \dot{q}^*_\varepsilon) = (\vartheta(\alpha^*), \dot{\vartheta}(\alpha^*)) \tag{3.64}$$

This theorem was proved by Dr. Ames in [4]; and as a result of the theorem, if we can solve the optimization problem, (3.62), then we will have obtained a set of parameters corresponding to bipedal robotic walking controllers along a stable periodic orbit. In the next section, we describe the application of this optimization method for the two walking robots of this study, and we show results from implementation of the resulting control laws in simulation and on hardware.

4. WALKING CONTROLLER IMPLEMENTATION

This section describes the results from implementation of the Human-Inspired Walking control design method in simulation and hardware demonstrations with two robot systems: the Leg Testbed robot at NASA Johnson Space Center (JSC) and the NAO robot in the AMBER Lab at Texas A&M. For each robot, we pose and solve the Human-Inspired Optimization to obtain controller parameters corresponding to stable, periodic walking in rigid-body dynamics simulation – results from these simulations are provided in the form of figures showing periodic orbits, states and torques, and kinematics of the walking. Additionally, we present the results from successful implementation of the control method on the two robot hardware systems.
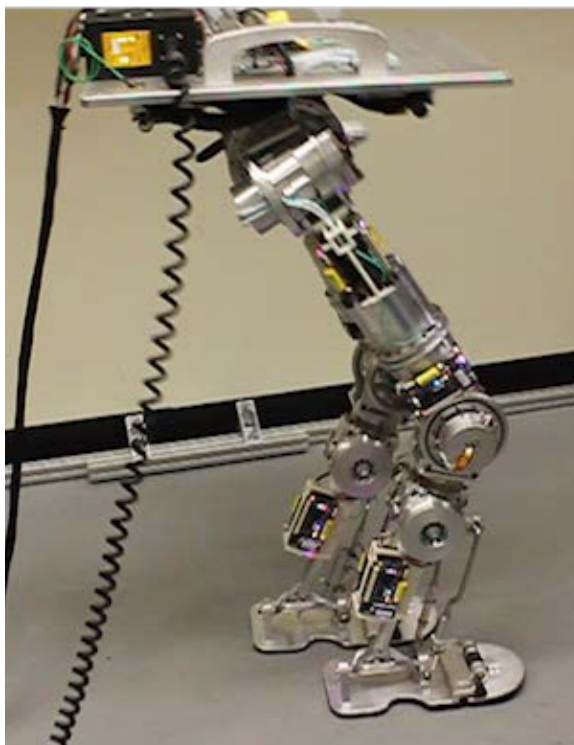
Figure 4.1: The Leg Testbed prototype biped robot.

4.1   JSC Leg Testbed Robot

As part of NASA's Graduate Student Research Fellowship program, we are fortunate to have the opportunity to implement our robot walking control method on a prototype biped robot which was built by engineers at NASA's Johnson Space Center and is referred to as the "Leg Testbed" robot. This prototype is a meter tall and weighs approximately 70 kilograms. The Leg Testbed, shown in Figure 4.1, uses technology similar to that of Robonaut 2 [2] – the first humanoid robot in space. Specifically, all joints on the Leg Testbed – except the ankles – are driven by series elastic actuators and the motor control system consists of twelve physically distinct motor drivers – one for each motor.

The rigid-body modeling process, as described in Section 2, is used to obtain a hybrid system model of the Leg Testbed (in the form of MATLAB code); inertial and length properties of the Leg Testbed, as used to compute the kinematics and the Lagrangian, are obtained through classified ProE CAD models. To simplify the mathematical model, the series elastic actuators are assumed to be perfect torque sources. Furthermore, the yaw joints are assumed to be fixed. Thus, the Leg Testbed is treated as a rigid, 10 degrees of freedom (DOF) system with the same ordering of

Table 4.1: Optimized control parameter values for the Leg Testbed robot model.

f.	$vhip$	a_1	a_2	a_3	a_4	a_5
$phipL$	0.2073	*	*	*	*	*
$mnsL$	*	0.0334	6.8777	0.0283	-1.3716	-0.0739
qsk	*	-0.0052	16.2863	0.0518	6.6773	0.7624
$qnsk$	*	-0.2667	-10.1798	-0.0088	-0.0566	1.0290
$thetatorso$	*	0.0004	7.2555	-0.0076	4.8858	0.0081
$qsar$	*	0.1443	-2.3936	0.1930	0.6970	0.0859
$qshr$	*	-0.0790	11.8453	-0.0173	1.3527	0.1108
$qnshr$	*	-0.1331	10.9139	-0.0191	0.7723	0.0527

joint axes as those described in the model of Section 2. The Leg Testbed dynamics are computed using Spatial Vector Algebra (SVA) [12]; we adapted the publicly available SVA MATLAB code [11] to compute the specific dynamics of the Leg Testbed.

4.1.1 Optimization

Using the modeled dynamics for the Leg Testbed, the optimization of Section 3.3 is developed and solved in MATLAB code which invokes the `fmincon` nonlinear constrained optimization routine. The original best fit to the human data – given in Table 3.2 – is used as an initial guess of the parameter matrix α_0. Constraints are added to the human-inspired optimization to ensure that the resulting gait is physically realizable by the Leg Testbed hardware. Specifically, the maximum torque is constrained to be less than 300 Newton-meters, joint velocities are constrained to be less than five radians per second and the feet are not allowed to self collide. The control parameters obtained by solving the optimization for the Leg Testbed are given in Table 4.1.

4.1.2 Simulation

MATLAB's `ode45` numerical integration function is used to simulate the hybrid control system, $\mathscr{H}_{\alpha^*,\varepsilon}^{L}$, modeling the Leg Testbed (hence the superscript L); in this simulation, the robot starts on the point on the guard, $(\vartheta(\alpha^*), \dot{\vartheta}(\alpha^*))$, and is controlled via the human-inspired control law, $u_{\alpha^*,\varepsilon}$ with parameters α^* obtained through optimization and $\varepsilon = 10$ as the control gain. Selected frames from one step of the simulated walking are shown in Figure 4.2; this figure portrays the aesthetics of the optimized walking gait, where it can be seen that the stride-length of this gait is relatively short. The conservative stride-length is intended to increase the probability of success when the controller is translated to implementation on the hardware.

The resulting periodic orbits for the pitch angles and roll angles of the system

45
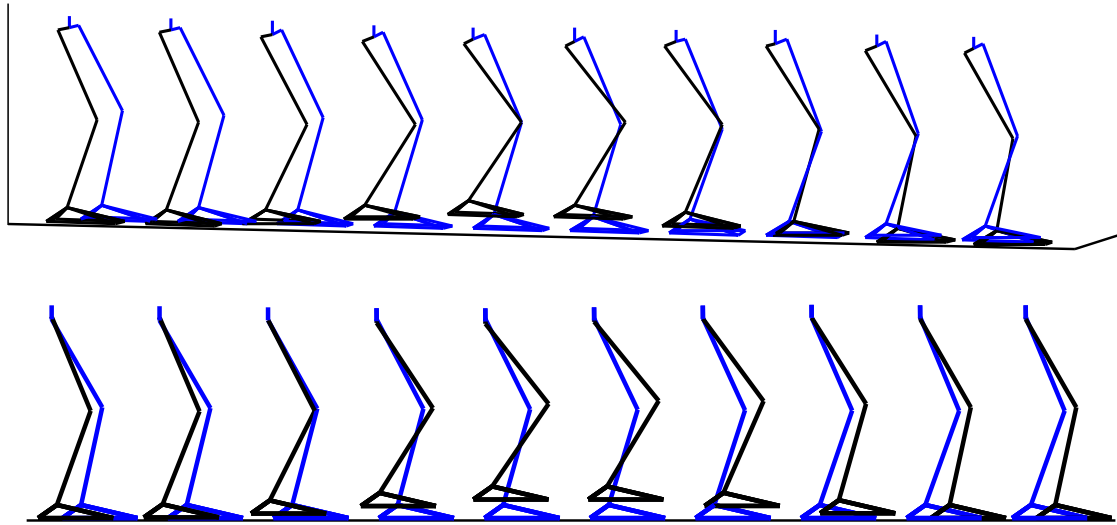
Figure 4.2: Snapshots from the Leg Testbed rigid body dynamics and human-inspired control walking simulation; here, we show a three-dimensional view (top) and a saggital plane view (bottom) for the simulated walking behavior.

are given in Figure 4.3a and Figure 4.3b. As seen in these figures, the optimization successfully limited the maximum joint velocity to be under three radians per second while still achieving closed periodic orbits. Similarly, through the use of constraints on the optimized control parameters, the maximum joint torque in simulation of this walking gait is less than 150 Newton-meters – which is well within the limitations of the Leg Testbed's motors. The simulated joint torques are shown in Figure 4.3c and Figure 4.3d. Perhaps the most remarkable aspect of the simulation of the control method is that the robot achieves these velocity and torque constraints while satisfying ZMP conditions, as see in Figure 4.3e and Figure 4.3f. Indeed, in the solving of the Human Inspired Optimization for the Leg Testbed, the most difficult constraints to satisfy is often the ZMP constraints; however, as they are satisfied for the gait of study, the resulting hardware implementation is much more robust.

(a) Periodic Orbits (Pitch)

(b) Periodic Orbits (Roll)

(c) Torques (stance)

(d) Torques (nonstance)

(e) ZMP Constraints (Pitch)

(f) ZMP Constraints (Roll)

Figure 4.3: This figure presents results from simulation of the Leg Testbed robot model. The top row shows periodic orbits for the simulated behavior of the Leg Testbed, showing pitch angles (a) and roll angles (b). The middle row show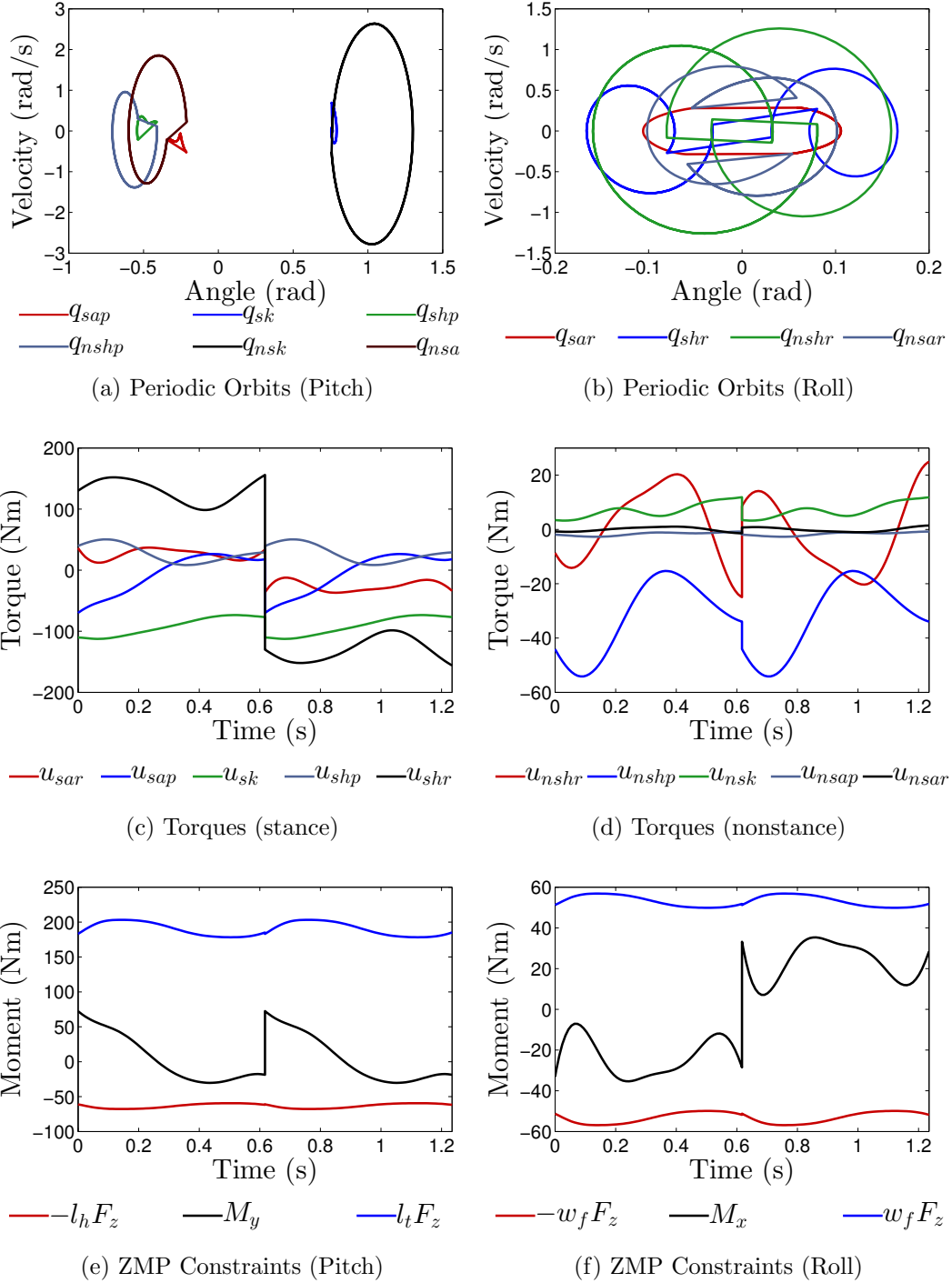s stance torques (c) and nonstance torques (d) from simulation. The bottom row shows that the simulated gait satisfies pitch (e) and roll (f) ZMP constraints.

4.1.3   Starting from Rest

The optimization problem yields control parameters corresponding to a periodic walking gait; however, to realize this walking on the physical hardware system, we need a control method for starting to walk from rest, i.e. a state with zero joint velocities. This problem motivated the development of "Motion Transitions", in which we use control to effect an additional partial hybrid zero dynamics surface which connects the at-rest-state to the periodic walking gait. In particular, we construct an additional control law for the first step of walking with the Leg Testbed; the primary goal of this controller is to transition the robot from a resting state to continuous walking – without falling over. Most of the ground work for "Motion Transitions" was established to solve the problem of starting from rest with NASA's Leg Testbed system; this work was later expanded upon, yielding novel methods for walking speed control [29], and multi-contact walking [21].

4.1.3.1   Extended Canonical Walking Function

It was found in [30] that to describe more complex walking motions, such as going up and down stairs, the canonical walking function must be augmented to account for the role that the environment plays on this system. Specifically, the *extended canonical walking function (ECWF)* is given by the time solution to a linear mass-spring-damper system subject to sinusoidal excitation:

$$
\begin{aligned}
y_H^e(t, \alpha_i^e) =\, & e^{-\alpha_{i,4}^e t}\big(\alpha_{i,1}^e \cos(\alpha_{i,2}^e t) + \alpha_{i,3}^e \sin(\alpha_{i,2}^e t)\big) \\
& + \alpha_{i,5}^e \cos(\alpha_{i,6}^e t) + \kappa(\alpha) \sin(\alpha_{i,6}^e t) + \alpha_{i,7}^e,
\end{aligned} \tag{4.1}
$$

where $\kappa(\alpha_i^e) = (2\alpha_{i,4}^e\alpha_{i,5}^e\alpha_{i,6}^e)/((\alpha_{i,4}^e)^2 + (\alpha_{i,2}^e)^2 - (\alpha_{i,6}^e)^2)$ and $i \in$ Outputs. Note that due to the linearity of the parameters $\alpha_{i,1}^e$, $\alpha_{i,3}^e$, $\alpha_{i,5}^e$ and $\alpha_{i,7}^e$ in (4.1), we can write

$$y_H^e(t, \alpha_i^e) = Y_H^e(t, \alpha_{i,2}^e, \alpha_{i,4}^e, \alpha_{i,6}^e) \begin{bmatrix} \alpha_{i,1}^e \\ \alpha_{i,3}^e \\ \alpha_{i,5}^e \\ \alpha_{i,7}^e \end{bmatrix} \tag{4.2}$$

where $Y_H^e(t, \alpha_i^e) \in \mathbb{R}^{1 \times 4}$ only depends on the parameters $\alpha_{i,2}^e$, $\alpha_{i,4}^e$, $\alpha_{i,6}^e$. The CWF can naturally be written as a special case of the ECWF by, given parameters $\alpha_i \in \mathbb{R}^5$ for the CWF (3.3), defining $\iota_e(\alpha_i) := (\alpha_i, 0, 0)$. Through this embedding, we can therefore consider the same human-inspired controller that was considered for the CWF by replacing the CWF with the ECWF in (3.17). Similarly, we can consider the PHZD surface for the ECWF which we denote by: $\mathbf{PZ}_{\alpha^e}$. Finally, we note that since ξ_1 is just the linearized position of the hip, which is used to parameterize time, we can write the parameterized ECWF as $y_H^e(\xi_1, \xi_1^0, v_{\text{hip}}, \alpha_i^e) := y_H^e(\frac{\xi_1 - \xi_1^0}{v_{\text{hip}}}, \alpha_i^e)$, which is now viewed as a function of ξ_1.

4.1.3.2 Motion Transitions

The advantage to the ECWF is that any two PHZD surfaces can be connected with the ECWF to ensure that partial hybrid zero dynamics is maintained, i.e., the ECWF can "glue" together any two PHZD surfaces; this is not possible with the CWF as there are not enough parameters present. To see this, let $(q(0), \dot{q}(0))$ be an at-rest state on an "at-rest PHZD surface", and let α^l be the parameters of the CWF associated with continuous walking as obtained through optimization. Associated with these parameters are the position of the hip at the beginning and end of a step: $\xi_1^{0,l} = \delta p_{\text{hip}}(\Delta_q \vartheta(\alpha^l))$ and $\xi_1^{f,l} = \delta p_{\text{hip}}(\vartheta(\alpha^l))$ and the position of the

hip in the at-rest state, $\xi_1^{0,l-1}$. To construct a surface connecting the at-rest state to continuous walking, consider actual outputs associated with the at-rest state and the ECWF at the end of a step associated with α^l:

$$y_i^0 = y_H^e(\xi_1^{0,l-1}, \xi_1^{0,l-1}, v_{\text{hip}}^*, \alpha_i^e)) \tag{4.3}$$

$$\dot{y}_i^0 = \frac{d}{d\xi_1} y_H^e(\xi_1, \xi_1^{0,l-1}, v_{\text{hip}}^*, \alpha_i^e) \Big|_{\xi_1 = \xi_1^{0,l-1}} \tag{4.4}$$

$$y_i^f = y_H^e(\xi_1^{f,*}, \xi_1^{0,*}, v_{\text{hip}}^*, \iota_e(\alpha_i^*)) \tag{4.5}$$

$$\dot{y}_i^f = \frac{d}{d\xi_1} y_H^e(\xi_1, \xi_1^{0,*}, v_{\text{hip}}^*, \iota_e(\alpha_i^*)) \Big|_{\xi_1 = \xi_1^{f,*}} \tag{4.6}$$

for $i \in$ Outputs. The goal is to find a parameters, α_i^e, for the ECWF such that $\iota_e(\alpha_i^*)$ can be replaced by α_i^e in (4.5)-(4.6). To achieve the goal of determining the parameters α_i^e, we utilize (4.2) to form the following matrix:

$$\mathbb{Y} = \begin{bmatrix} Y_H^e(\xi_1^{0,l-1}, \xi_1^{0,l-1}, v_{\text{hip}}^l, \alpha_{i,2}^e, \alpha_{i,4}^e, \alpha_{i,6}^e) \\ \frac{d}{d\xi_1} Y_H^e(\xi_1, \xi_1^{0,l-1}, v_{\text{hip}}^l, \alpha_{i,2}^e, \alpha_{i,4}^e, \alpha_{i,6}^e) \Big|_{\xi_1 = \xi_1^{0,l-1}} \\ Y_H^e(\xi_1^{f,l}, \xi_1^{0,l}, v_{\text{hip}}^l, \alpha_{i,2}^e, \alpha_{i,4}^e, \alpha_{i,6}^e) \\ \frac{d}{d\xi_1} Y_H^e(\xi_1, \xi_1^{0,l}, v_{\text{hip}}^l, \alpha_{i,2}^e, \alpha_{i,4}^e, \alpha_{i,6}^e) \Big|_{\xi_1 = \xi_1^{f,l}} \end{bmatrix}$$

It is easy to verify that picking $\alpha_{i,2}^e = \alpha_{i,2}^l$, $\alpha_{i,4}^e = \alpha_{i,4}^l$ and $\alpha_{i,6}^e > 0$ results in $\mathbb{Y}$ being nonsingular. Therefore, the final four parameters of α_i^e can be determined by picking:

$$\begin{bmatrix} \alpha_{i,1}^e \\ \alpha_{i,3}^e \\ \alpha_{i,5}^e \\ \alpha_{i,7}^e \end{bmatrix} = \mathbb{Y}^{-1} \begin{bmatrix} y_i^0 \\ \dot{y}_i^0 \\ y_i^f \\ \dot{y}_i^f \end{bmatrix}$$

The end result of solving for α^e in this manner is that when started from the

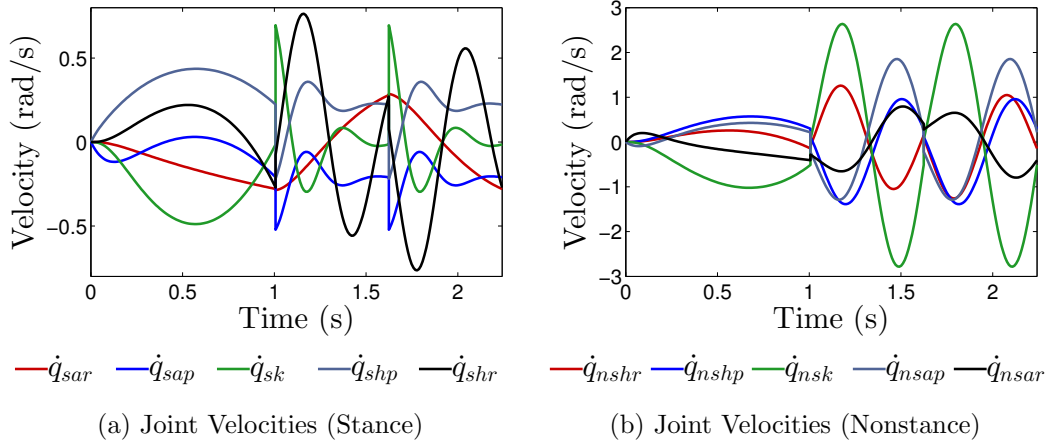(a) Joint Velocities (Stance)   (b) Joint Velocities (Nonstance)

Figure 4.4: This figure presents results from simulation of the Leg Testbed robot model starting from rest, showing joint velocities for the stance leg (left) and the nonstance leg (right) over the course of five steps. The controller applied in this simulation uses a Motion Transition to successfully move the robot from an at-rest state (zero velocity) to the periodic walking orbit obtained through optimization.

specified initial rest-state, $(q(0), \dot{q}(0))$, the controller will drive the robot along the PHZD surface for the Motion Transition, $\mathbf{PZ}_{\alpha^e}$, which intersects the continuous walking gait PHZD surface, $\mathbf{PZ}_{\alpha^l}$, at the guard (nonstance foot strike). In other words, we will have connected the initial state,$(q(0), \dot{q}(0))$, and $\mathbf{PZ}_{\alpha^l}$ through $\mathbf{PZ}_{\alpha^e}$, which then allows us to test (on the robot hardware) the continuous walking gaits that result from solving the primary optimization problem discussed in this thesis. Figure 4.4a and Figure 4.4b show results from simulation of a Motion Transition for the Leg Testbed.

4.1.4 Experimental Implementation

Using the Motion Transition to control the robot from a rest state to our optimized walking gait, we successfully achieved multiple steps on the Leg Testbed platform. Snapshots from an experimental walking gait are shown in Figure 4.5. To implement the proposed control method for walking on the Leg Testbed hardware,
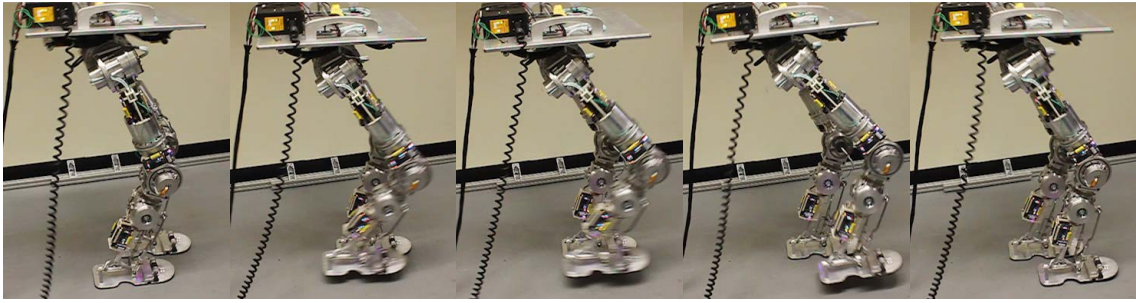
Figure 4.5: Snapshots from a Leg Testbed walking experiment.

we interfaced with the preexisting Leg Testbed control code for sending position commands to the Leg Testbed's motor drivers. While the Leg Testbed is capable of a variety of control modes – position, torque, voltage and impedance control – for the scope of this thesis, and based on the constrained time-line for implementation over the course of two 10-week internships, we selected to use only the position control functionality. As used in the Summer 2012 walking experiments at NASA JSC, this control scheme consisted primarily of a slow (200 Hz) position command loop termed the "high-level control" – running on the (off-board) operator computer – used to communicate desired positions to the twelve motor controllers on the Leg Testbed, which themselves run fast "low-level" (10000 Hz) PD controllers on the error in measured and commanded position. For the walking gaits considered, the 10 kHz position controllers showed excellent performance, tracking our desired trajectories with a maximum error of 0.01 radians, a point to which we will return in the following sections.

4.1.4.1   ROS C++ Position Control Scheme and LUA Operator Interface

The high-level control framework uses a combination of the Robot Operating System (ROS) [31] and Open Robot Control Software (OROCOS) [25] to generate and communicate commands, which are sent to the motor drivers using a custom protocol

termed "robonet." We interfaced with this system via a C++ implementation of a ROS node which publishes joint position commands at 200 Hz. Specifically, we used the angle reconstruction method from the optimization to compute desired position commands, corresponding to the optimized periodic orbit with parameters α^* and fixed point $(\vartheta(\alpha^*), \dot{\vartheta}(\alpha^*))$, to be published on rostopics (which ultimately communicate motor position commands) at intervals of 0.005 seconds. For operating and performing experiments on testbeds, one method the Robonaut software architecture used was a LUA script which invokes ROS and OROCOS nodes. We interfaced with this system through the creation of a set of LUA functions to execute our C++ ROS walking node. The LUA script was also used for initialization, calibration, tuning, and debugging the 10 kHz turbo driver position controllers.

4.1.4.2 Implementation

After calibrating the joints and tuning the position control gains, we implemented the walking gait obtained from the human-inspired control method for a total of seven steps (limited by lab space). The initial rest configuration was determined through heuristics, in which the robot's COM was placed over the left foot via modification of the robot's roll angles. To begin walking, we implemented a Motion Transition – which connects the initial rest state to the continuous walking gait from optimization – through joint trajectory tracking. After the first step, the control scheme switched to the continuous walking gait which resulted from optimization.

The experiment was repeatable; in the few unsuccessful trials, our method failed due to lack of robustness in the very first step of the gait (this is a major point of consideration for future work). Commanded (PosCom) and measured (APS1 and APS2) motor positions from one experiment are shown versus time in Figure 4.7 and Figure 4.8; observe that the Leg Testbed motor controller performance was

exceptional – the maximum error in position tracking during these experiments was less than 0.02 radians. The success of these walking experiments with the Leg Testbed hardware shows that the proposed bipedal robotic walking control method can be used to generate physically realizable walking gaits. To the best of our knowledge, these experiments were the first in which the Leg Testbed robot successfully walked forward multiple steps. Snapshots of the walking experiments – taken from different viewpoints – are shown in Figure 4.6.
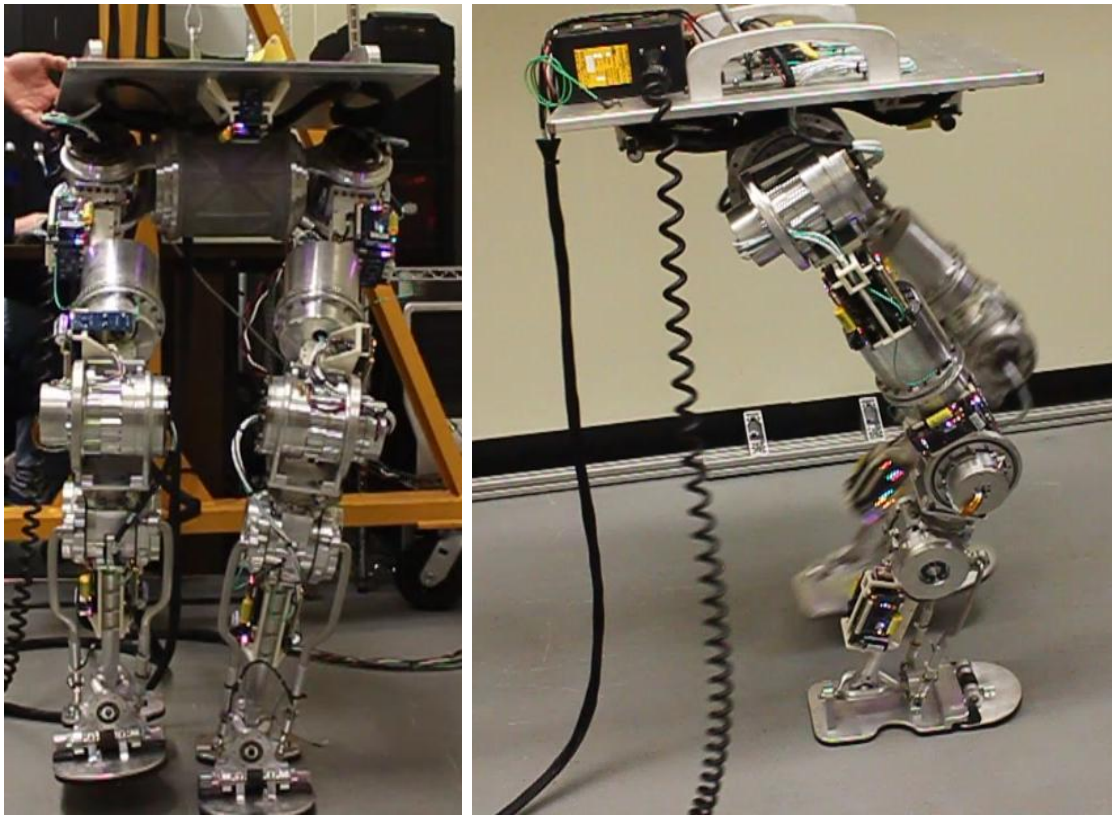
Figure 4.6: Snapshots from two Leg Testbed walking experiments.

Figure 4.7: Commanded (PosCom) and measured (APS1) positions during one Leg Testbed walking experiment are given, showing the first three joints – hip pitch (J1), yaw (J2) and roll(J3) – on the left (LL) and right (RL) legs, and also showing the measurements of the output of the springs (APS2).
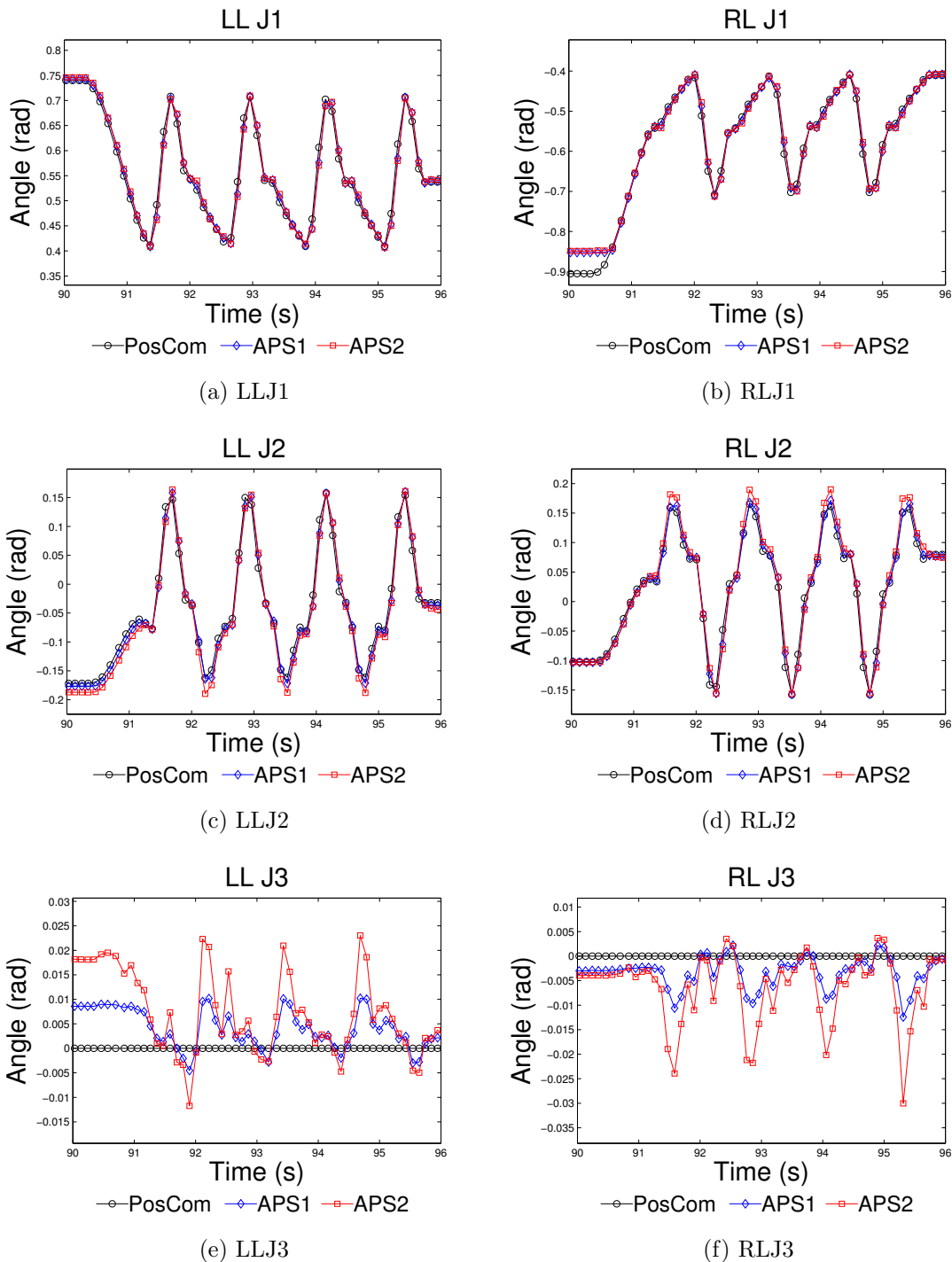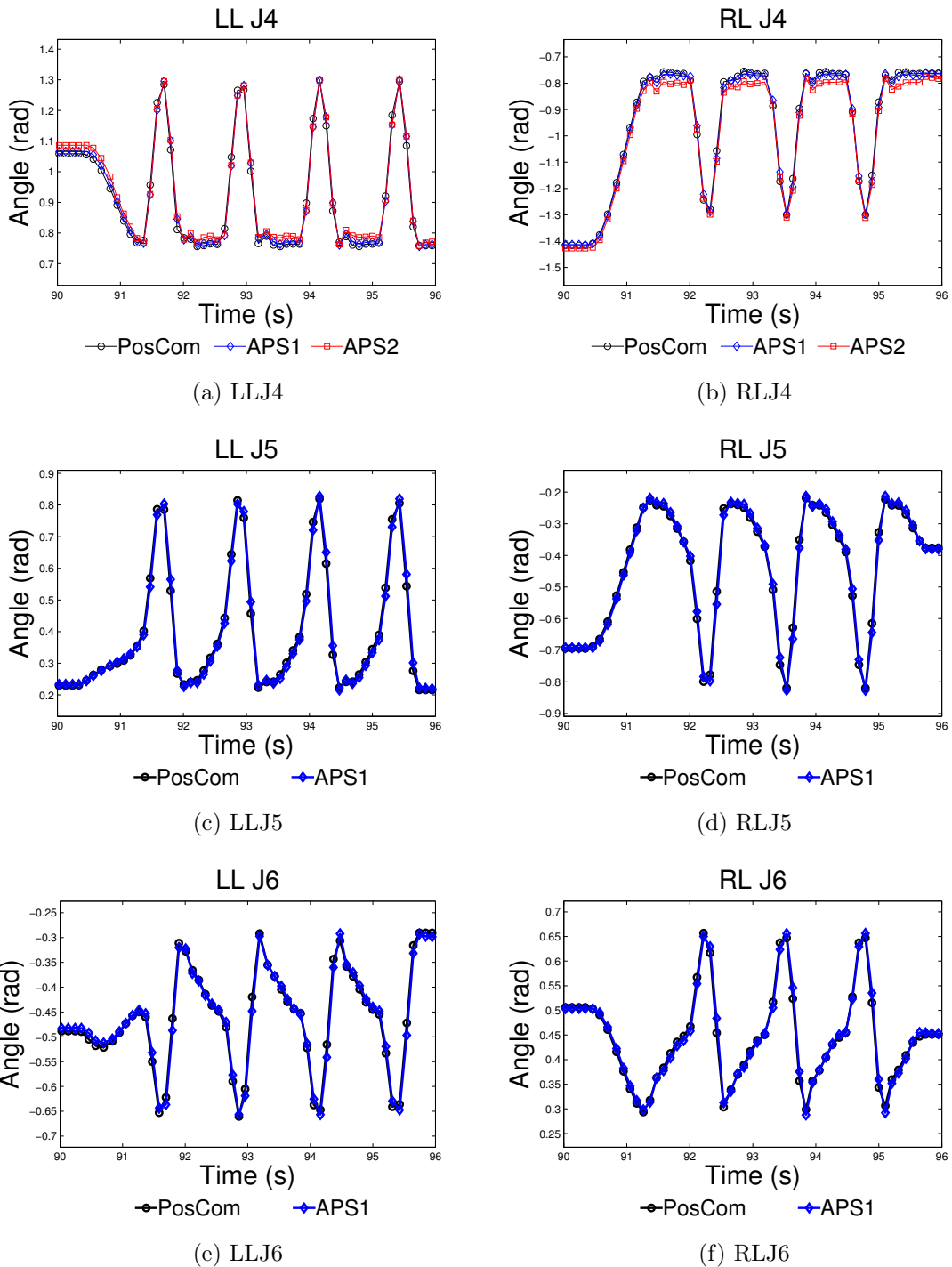
Figure 4.8: Commanded (PosCom) and measured (APS1) positions during one Leg Testbed walking experiment are given, showing the last three joints – knee(J4) and both ankle motors (J5 and J6) – on the left (LL) and right (J2) legs. Note that the ankle joints are not driven by series elastic actuators, and therefore, do not have APS2s.
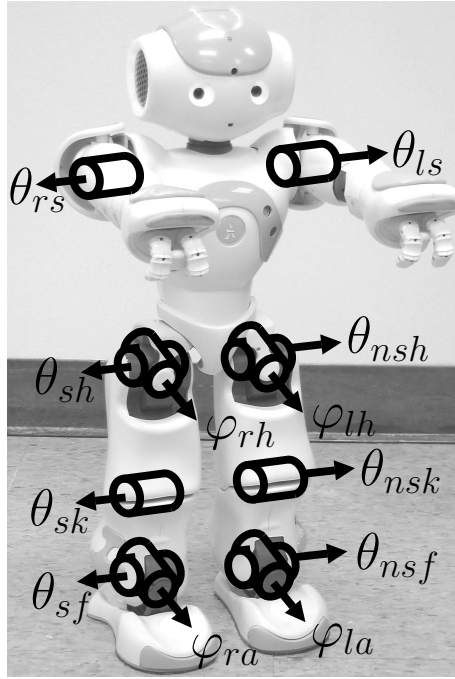
Figure 4.9: NAO humanoid robot

4.2 NAO Robot

We also implemented the proposed method in the design of walking controllers for the NAO [1] robot – a 573 millimeter tall humanoid biped which weighs 5.2 kilograms. NAO's lower body consists of 10 actuated degrees of freedom, with joint axes which map one-to-one to those in the robot model (see Figure 4.9). For modeling purposes, the 15 joints in NAO's upper body are treated as fixed, such that the entire upper body is treated as one rigid link with constant mass and inertia properties computed from the composite masses and inertia of the limbs in the upper body. The mass and length properties of NAO are obtained from the specifications sheet given on Aldebaran's website [1] – these properties are used in the development of Mathematica (ported to MATLAB .m functions) and MATLAB code which computes kinematics and dynamics as described in Section 2 and the Feedback lin-

earization controller proposed in Section 3. The dynamics were first calculated using the Mathematica modules available at [22]; we later switched to a custom MATLAB numeric implementation of the dynamics, expanding on suggestions and code from Jessy Grizzle and Ryan Sinnet [13], to improve execution speed.

4.2.1 Optimization

Using the modeled dynamics for NAO, the optimization of Section 3.3 is developed and solved in MATLAB code which invokes the `fmincon` nonlinear constrained optimization routine. The original best fit to the human data – given in 3.2 – is used as an initial guess of the parameter matrix α_0. Constraints are added to ensure that the resulting gait is physically realizable by the NAO hardware: joint velocities must be less than 3.5 radians per second, knee angles must be greater than zero (cannot hyper-extend), and to avoid self collision, the feet are not allowed to come closer than 1 cm to each other (laterally). The optimized parameters, α^*, are given in Table 4.2.

4.2.2 Simulation

After determining optimal control parameters for NAO walking, MATLAB's `ode45` numerical integration function is used to simulate the hybrid control sys-

Table 4.2: Optimized control parameter values for the NAO robot model.

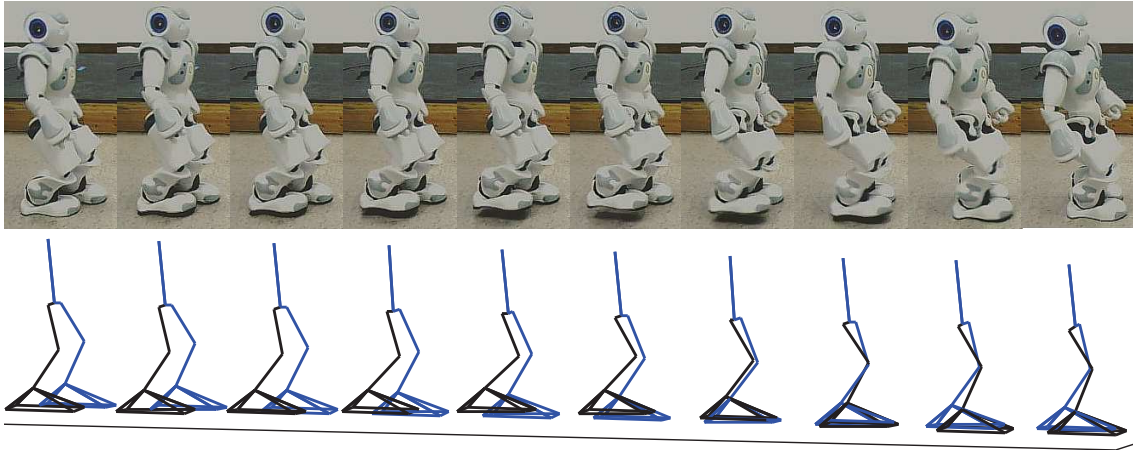| f.	$vhip$	a_1	a_2	a_3	a_4	a_5
$phipL$	0.1432	*	*	*	*	*
$mnsL$	*	0.0594	10.7146	0.0388	-3.0263	0.1410
qsk	*	-0.0002	16.9486	-0.0001	-15.8735	0.9581
$qnsk$	*	-0.2180	-15.8541	0.1290	1.0987	1.0777
$thetatorso$	*	-0.0001	4.6137	0.0000	-4.6512	0.1000
$qsar$	*	0.0373	-0.0027	-4.4946	0.2393	-0.0516
$qshr$	*	-0.1441	-3.9159	0.1176	0.10440	0.1475
$qnshr$	*	-0.0905	18.2141	-0.0272	0.6090	0.0663

Figure 4.10: Comparison of snapshots from the actual (top) and simulated (bottom) walking with the NAO robot using the proposed control method.

tem, $\mathscr{H}^N_{\alpha^*,\varepsilon}$, modeling the NAO robot (hence the superscript N); in this simulation, the robot starts on the point on the guard, $(\vartheta(\alpha^*), \dot\vartheta(\alpha^*))$, and is controlled via the human-inspired control law, $u_{\alpha^*,\varepsilon}$ with parameters α^* obtained through optimization and $\varepsilon = 10$ as the control gain. Selected frames from one step of the simulated walking are shown in Figure 4.10. The resulting periodic orbits for the pitch angles and roll angles of the system are given in Figure 4.11a and Figure 4.11b respectively. Notice that the nonstance knee joint reaches 3.5 radians per second, which is the upper bound on allowable joint velocities enforced via a constraint in the human-inspired optimization for NAO; velocities of the other joints do not exceed three radians per second. Furthermore, as shown in Figure 4.11e and Figure 4.11f, the torque commands generated by our human-inspired control law are all less than three Newton-meters – well within NAO's limits.

4.2.3 Experimental Implementation

The human inspired control approach is implemented experimentally on the actual NAO robot via open-loop trajectory tracking. Specifically, NAO's built-in PID

59

controller is used to track the $q(t)$ trajectories from a simulation in which the robot takes 10 steps. These trajectories were communicated to NAO's on-board computer via a custom set of C++ functions which were written by fellow AMBER lab researchers, Eric Cousineau, Wenlong Ma, Ayongga Hereid and Ryan Sinnet. Using this experimental configuration, the NAO robot successfully completed 10 steps repeatably over several trials. The relative degree 2, actual control outputs, y_2^a, are computed from both simulation and experiment data for one trial and compared against one another in Figure 4.12 and Figure 4.13. Note that the experimental outputs agree closely with the simulated outputs (with minor discrepancies in θ_{sk}, θ_{tor} and φ_{sa}, which are a result of the open-loop controller). Snapshots of the experimental walking are given with the simulated gait on the previous page; where again, the experiment and simulation match very closely. A video of the walking achieved on NAO is available online [28]. Our walking results with NAO were well received during presentation at the 2013 IEEE International Conference on Robotics and Automation (ICRA) in Germany [29].

(a) Periodic Orbits (Pitch)

(b) Periodic Orbits (Roll)

(c) Outputs (Sagittal)

(d) Outputs (Coronal and Foot)

(e) Torques (Stance)

(f) Torques (Nonstance)

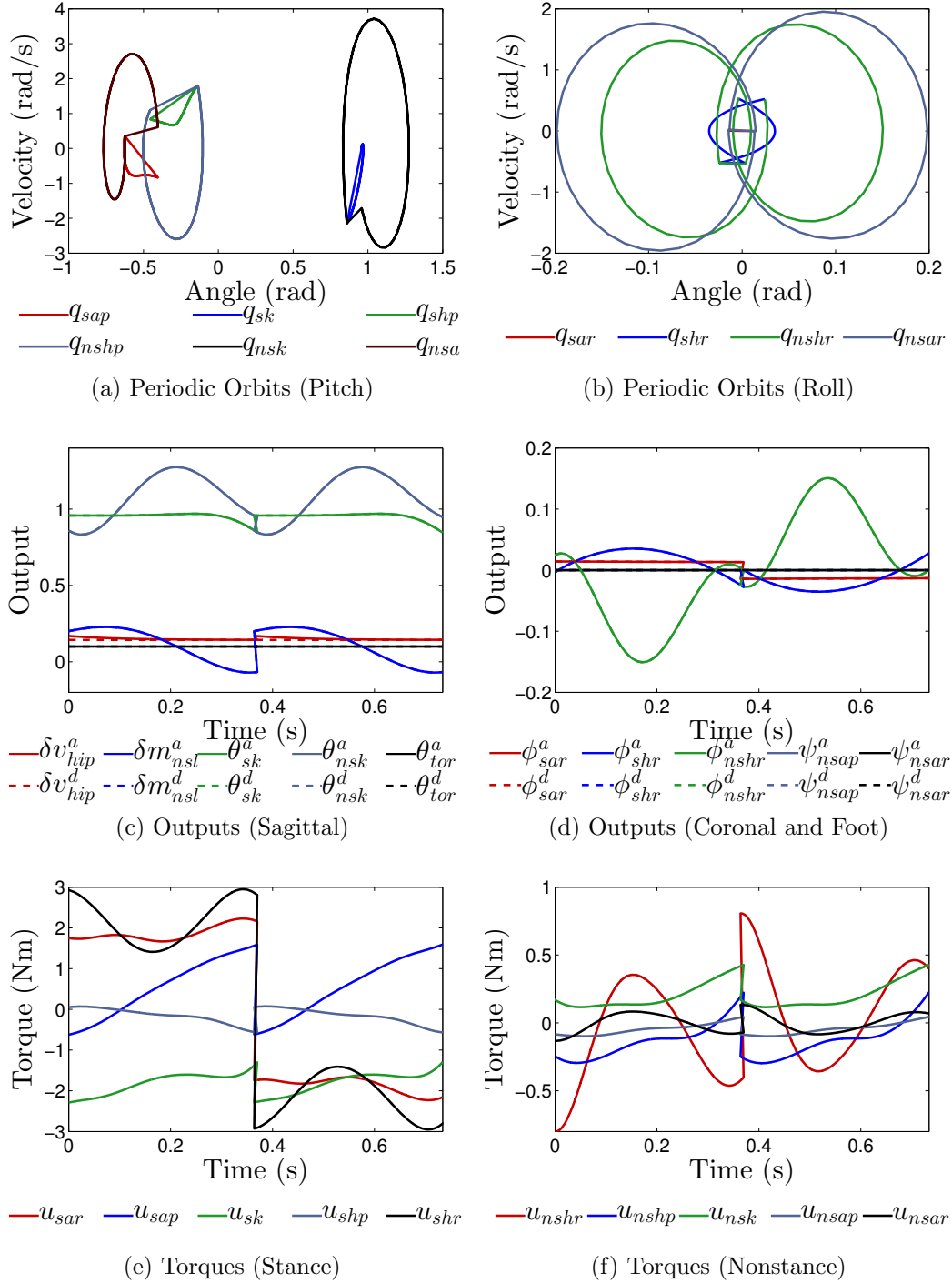Figure 4.11: This figure presents results from simulation of the NAO robot model. The top row shows periodic orbits for the simulated behavior of NAO, showing pitch angles (a) and roll angles (b). The middle row shows optimized desired and corresponding simulated (actual) walking sagittal outputs (c) and coronal and nonstance foot outputs (d). The bottom row shows stance torques (e) and nonstance torques (f) from simulation.

61

(a) Non-Stance Slope

(b) Stance Knee

(c) Non-Stance Knee

(d) Torso Angle
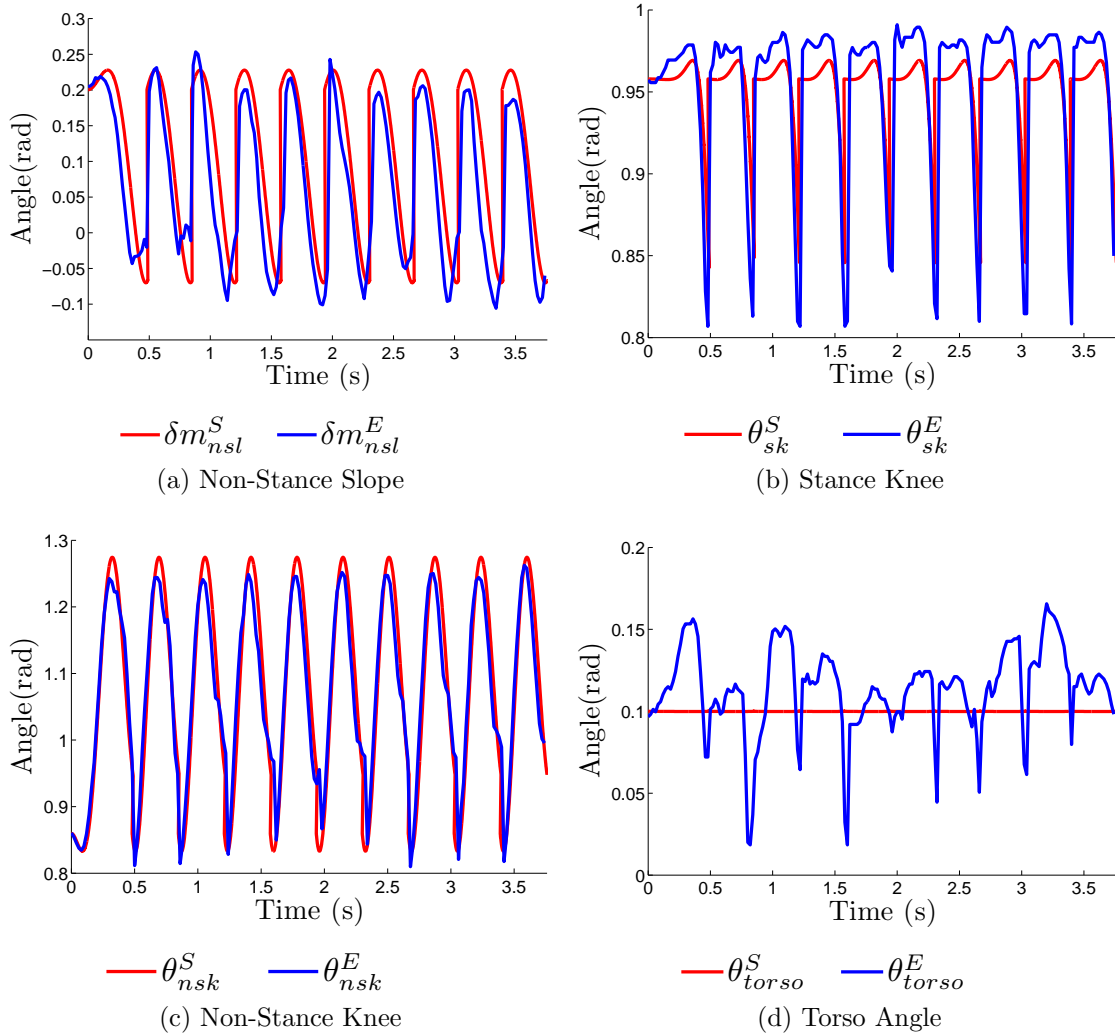
Figure 4.12: Simulated and experimental values for the relative degree two actual sagittal outputs, y_2^a, are shown over the course of 10 steps in NAO walking with the proposed control method.
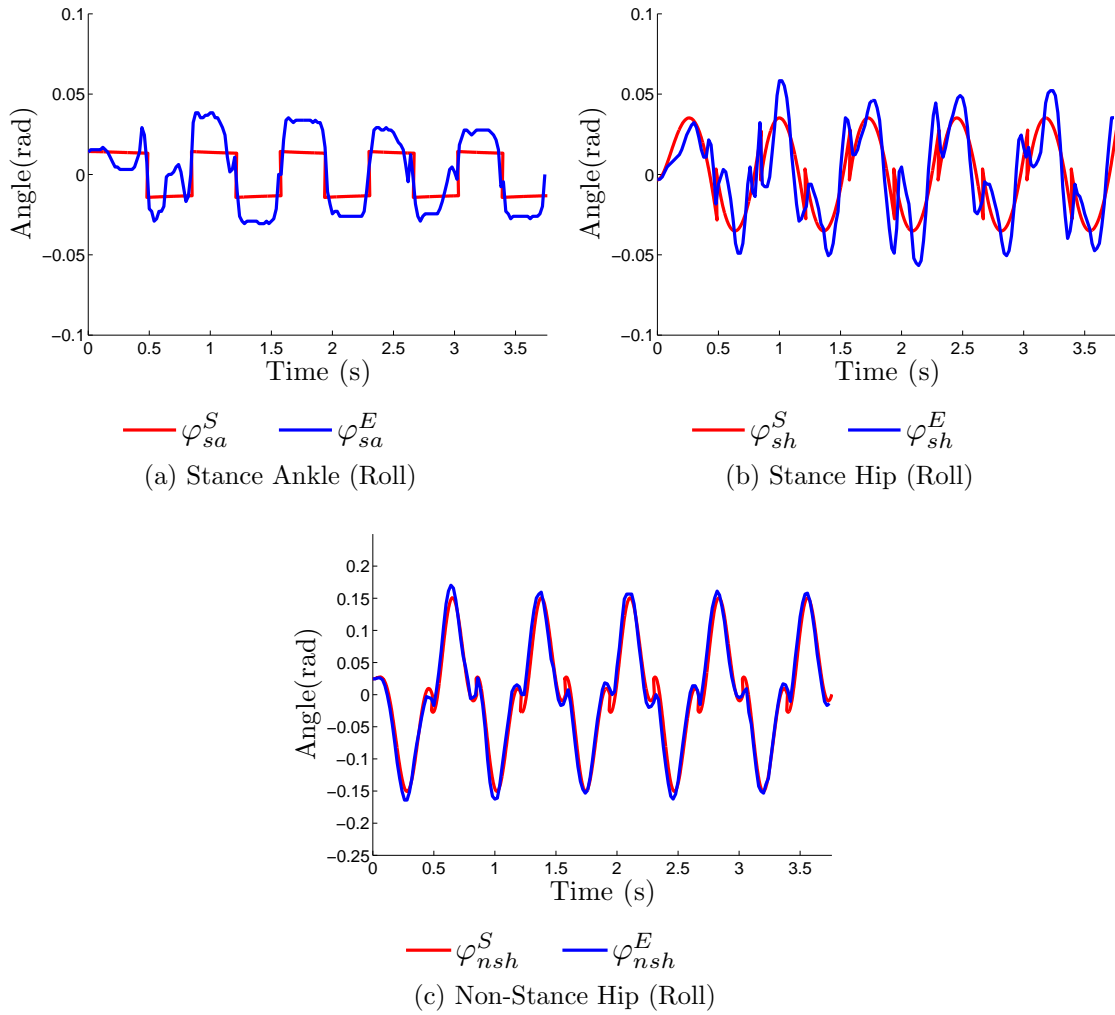
Figure 4.13: Simulated and experimental values for the relative degree two actual roll outputs, y_2^a, are shown over the course of 10 steps in NAO walking with the proposed control method.

5. CONCLUSION

In this thesis, we present a method for the construction of control laws corresponding to stable, periodic bipedal robotic walking in a hybrid system model of an actual robot hardware platform. The hybrid system model estimates the relationship between control command and resulting motion in the real robot; we show how to construct this model through the computation of the Euler-Lagrange equations of motion to predict the behavior of the system during leg swing and through the computation of impact equations to predict the behavior of the robot during foot-ground collision. Motivated to achieve stable walking with robot hardware, we equate stable walking with stable, periodic orbits in the hybrid system model of the robot and seek to construct control laws corresponding to stable, periodic orbits in our hybrid control system.

Our control approach begins with the analysis of experimental human walking data. Through examination of the data, we find that all nine test subjects display universal behaviors while walking forward on a level surface. Specifically, we observe that certain kinematic functions, or outputs, on the walking data can be represented, with high correlation, by the time solution to under-damped, second order system – termed the *canonical walking function*. We leverage this insight to frame parameterized control laws for our hybrid system model of the robot. Using feedback linearization, these torque controllers drive the actual robot outputs to desired values of these outputs as determined by parameterized canonical walking functions. The main contribution of this thesis is an optimization problem which determines control parameters which result in stable, periodic orbits in the hybrid system model of the robot while simultaneously satisfying constraints on the system, e.g. velocity

and torque limits, as dictated by the limitations of the robot hardware.

The control parameter optimization expands upon the Human-Inspired Control method for achieving walking in a planar, point-foot model of the robot. In both optimizations, realization of a *partial hybrid zero dynamics* is encoded via closed-form computation of equality constraints. Novel to this thesis is extension of the method to unsupported, three-dimensional walking with feet and the addition of constraints necessary to achieve walking with actual robot hardware. These constraints (ZMP, torque bounds, etc.) are encoded via inequality constraints, which are computed in closed form via calculation of the robot's state in closed form over the course of the step.

The method was implemented on two robot hardware platforms: the NAO robot walking in the AMBER Lab at Texas A&M and the Leg Testbed robot walking in the Robonaut Lab at NASA JSC. Results from simulation of the hybrid system model for each system show the closed, periodic orbits achieved by the proposed control law and corresponding torques and outputs. As part of larger motivations for this work, we also implemented the method on the actual robot hardware systems via motor position control on trajectories from the stable periodic orbits. Both robots took several steps in multiple walking experiments; in other words, the proposed method resulted in repeatable experimental walking with two robots! Future work will entail expanding the method to include other locomotion behaviors, e.g. turning, heel-toe walking, standing and crouching.

REFERENCES

[1] ALDEBARAN ROBOTICS. NAO. `http://www.aldebaran-robotics.com`. Accessed 01-Oct-2013.

[2] AMBROSE, R. O., ALDRIDGE, H., ASKEW, R. S., BURRIDGE, R. R., BLUETHMANN, W., DIFTLER, M., LOVCHIK, C., MAGRUDER, D., AND REHNMARK, F. Robonaut: NASA's space humanoid. *IEEE Intelligent Systems and their Applications 15*, 4 (2000), 57–63.

[3] AMES, A. D. First steps toward automatically generating bipedal robotic walking from human data. In *Robotic Motion and Control 2011* (London, 2012), vol. 422 of *Lecture Notes in Control and Information Sciences*, Springer, pp. 89–116.

[4] AMES, A. D., COUSINEAU, E. A., AND POWELL, M. J. Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control* (Beijing, China, Apr. 2012), pp. 135–144.

[5] AMES, A. D., VASUDEVAN, R., AND BAJCSY, R. Human-data based cost of bipedal robotic walking. In *Proceedings of the 14th ACM International Conference on Hybrid systems: Computation and Control* (Chicago, USA, Apr. 2011), pp. 153–162.

[6] BRAUN, D. J., MITCHELL, J. E., AND GOLDFARB, M. Actuated dynamic walking in a seven-link biped robot. *IEEE Transactions on Robotics 17*, 1 (Feb. 2012), 147–56.

[7] CHEVALLEREAU, C., BESSONNET, G., ABBA, G., AND AOUSTIN, Y. *Bipedal Robots: Modeling, Design and Walking Synthesis.* Wiley-ISTE, New York, 2009.

[8] CHEVALLEREAU, C., DJOUDI, D., AND GRIZZLE, J. W. Stable bipedal walking with foot rotation through direct regulation of the zero moment point. *IEEE Transactions on Robotics 25*, 2 (Apr. 2008), 390–401.

[9] ERBATUR, K., OKAZAKI, A., OBIYA, K., TAKAHASHI, T., AND KAWAMURA, A. A study on the zero moment point measurement for biped walking robots. In *Proceedings of the 7th International Workshop on Advanced Motion Control* (Maribor, Slovenia, July 2002), pp. 431–436.

[10] FATEHI, N., AKBARIMAJD, A., AND ASADPOUR, M. ZMP analysis for dynamic walking of a passivity-based biped robot with flat feet. In *Proceedings of the 7th International Workshop on Control Automation and Systems (ICCAS)* (Gyeonggi-do, South Korea, Oct. 2010), pp. 1419–1423.

[11] FEATHERSTONE, R. Spatial vector and rigid-body dynamics software. `http://royfeatherstone.org/spatial/v2/index.html`. Accessed 01-Oct-2013.

[12] FEATHERSTONE, R. *Rigid Body Dynamics Algorithms.* Springer, New York, 2008.

[13] GRIZZLE, J. W., CHEVALLEREAU, C., AMES, A. D., AND SINNET, R. W. 3D bipedal robotic walking: models, feedback control, and open problems. In *Proceedings of the 8th International Federation of Automatic Control Symposium on Nonlinear Control Systems* (Bologna, Italy, Sept. 2010), pp. 505–532.

[14] GRIZZLE, J. W., CHEVALLEREAU, C., AND SHIH, C. HZD-based control of a five-link underactuated 3D bipedal robot. In *IEEE Conf. on Decision and Control* (Cancún, Mexico, Dec. 2008), pp. 5206–5213.

[15] Hitt, J. K. DARPA warrior web. `http://www.darpa.mil/Our_Work/DSO/`
`Programs/Warrior_Web.aspx`. Accessed 01-Oct-2013.

[16] Holmes, P., Full, R. J., Koditschek, D. E., and Guckenheimer, J.
The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM
Review 48*, 2 (Feb. 2006), 207–304.

[17] Huang, Y., Wang, Q., Chen, B., Xie, G., and Wang, L. Modeling and
gait selection of passivity-based seven-link bipeds with dynamic series of walking
phases. *Robotica 30*, 1 (Jan. 2012), 39–51.

[18] Hürmüzlü, Y., and Marghitu, D. B. Rigid body collisions of planar kine-
matic chains with multiple contact points. *International Journal of Robotics
Research 13*, 1 (1994), 82–92.

[19] Isidori, A. *Nonlinear Control Systems.* Springer, London, 1995.

[20] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K.,
Yokoi, K., and Hirukawa, H. Biped walking pattern generator allowing
auxiliary ZMP control. In *IEEE/RSJ International Conference on Intelligent
Robots and Systems* (Beijing, China, Oct. 2010), pp. 2993–2999.

[21] Lack, J., Powell, M. J., and Ames, A. D. Planar multi-contact bipedal
walking using hybrid zero dynamics and quadratic programs. In *Proceedings of
the 2014 IEEE International Conference on Robotics and Automation (ICRA)*
(Hong Kong, China, June 2014).

[22] Murray, R. M., Li, Z., and Sastry, S. S. Robotlinks Mathematica pack-
age. `http://www.cds.caltech.edu/~murray/mlswiki/index.php/Software`.
Accessed 01-Oct-2013.

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

[24] Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura, Algarve, Portugal, Oct. 2002), pp. 2684–2689.

[25] Open Robot Control Software. `http://www.orocos.org`. Accessed 01-Oct-2013.

[26] Park, J. H., and Cho, H.-C. An online trajectory modifier for the base link of biped robots to enhance locomotion stability. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* (San Francisco, USA, Apr. 2000), pp. 3353–3358.

[27] PhaseSpace Inc. Phasespace motion capture. `http://www.phasespace.com`. Accessed 01-Oct-2013.

[28] Powell, M., Hereid, A., and Ames, A. 3D walking with NAO through human-inspired optimization.
`http://www.youtube.com/watch?v=ql2XfwuP2uI`. Accessed 01-Oct-2013.

[29] Powell, M. J., Hereid, A., and Ames, A. D. Speed regulation in 3D robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation* (Karlsruhe, Germany, May 2013), pp. 4803–4810.

[30] Powell, M. J., Zhao, H., and Ames, A. D. Motion primitives for human-inspired bipedal robotic locomotion: walking and stair climbing. In *Proceedings*

of the 2012 IEEE International Conference on Robotics and Automation (St. Paul, USA, May 2012), pp. 543–549.

[31] ROBOT OPERATING SOFTWARE. `http://wiki.ros.org`. Accessed 01-Oct-2013.

[32] SAKAGAMI, Y., WATANABE, R., AOYAMA, C., MATSUNAGA, S., HIGAKI, N., AND FUJIMURA, K. The intelligent ASIMO: system overview and integration. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura, Algarve, Portugal, Oct. 2002), pp. 2478–2483.

[33] SANKAI, Y. Hal: Hybrid assistive limb based on cybernics. In *Robotics Research*, M. Kaneko and Y. Nakamura, Eds., vol. 66 of *Springer Tracts in Advanced Robotics*. Springer, Berlin, 2011, pp. 25–34.

[34] SARDAIN, P., AND BESSONNET, G. Gait analysis of a human walker wearing robot feet as shoes. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation* (Seoul, Korea, May 2001), pp. 2285–2292.

[35] SASTRY, S. S. *Nonlinear Systems: Analysis, Stability and Control.* Springer, New York, 1999.

[36] SPONG, M. W. Passivity based control of the compass gait biped. In *Proceedings of the 1999 International Federation of Automatic Control World Congress* (Beijing, China, July 1999).

[37] SPONG, M. W., AND VIDYASAGAR, M. *Robot Dynamics and Control.* Wiley, Hoboken, 1989.

[38] STRONGE, W. J. *Impact Mechanics.* Cambridge University Press, Cambridge, 2000.

[39] TAKANISHI, A., OK LIM, H., TSUDA, M., AND KATO, I. Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface. In *Proceedings of the 1990 IEEE/RSJ International Workshop on Intelligent Robots and Systems* (Ibaraki, Japan, July 1990), pp. 323–330.

[40] VASUDEVAN, R., AMES, A. D., AND BAJCSY, R. Human based cost from persistent homology for bipedal walking. In *Proceedings of the 2011 International Federation of Automatic Control World Congress* (Milano, Italy, Aug. 2011), pp. 3292–3297.

[41] VUKOBRATOVIĆ, M., AND BOROVAC, B. Zero-moment point–thirty-five years of its life. *International Journal of Humanoid Robotics 1*, 1 (2005), 157–173.

[42] WESTERVELT, E. R., GRIZZLE, J. W., CHEVALLEREAU, C., CHOI, J. H., AND MORRIS, B. *Feedback Control of Dynamic Bipedal Robot Locomotion.* CRC Press, Boca Raton, 2007.

[43] WESTERVELT, E. R., GRIZZLE, J. W., AND KODITSCHEK, D. E. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control 48*, 1 (2003), 42–56.

[44] WINTER, D. A. *Biomechanics and Motor Control of Human Movement.* Wiley, Hoboken, 2009.