View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Texas A&M Repository

PLANAR MULTICONTACT LOCOMOTION USING HYBRID ZERO DYNAMICS

A Thesis

by

JORDAN T. LACK

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,	Richard Malak
	John Hurtado
Department Head,	Andreas Polycarpou

December 2013

Major Subject: Mechanical Engineering

Copyright 2013 Jordan T. Lack

ABSTRACT

This thesis proposes a method for generating multi-contact, humanlike locomotion via a human-inspired optimization. The chief objective of this work is to offer an initial solution for obtaining multi-domain walking gaits containing domains with differing degrees of actuation. Motivated by the fact that locomotion inherently includes impacts, a hybrid systems approach is used. Through Lagrangian mechanics, a dynamic model of the system is derived that governs the continuous dynamics, while the dynamics during the impacts are modeled assuming perfectly plastic impacts in which the ground imparts an impulsive force on the impacting link.

Using the dynamic model of the planar bipedal robot Amber 2, a seven link biped, a human-inspired optimization is presented which leverages the concept of zero dynamics, allowing for a low dimensional representation of the full order dynamics. Within the optimization, constraints are constructed based on the interaction between the robot and the walking surface that ensure the optimized gait is physically realizable. Other constraints can be used to influence or "shape" the optimized walking gait such as kinematic and/or torque constraints. This optimized walking gait is then realized through the method of Input/Output Linearization. Finally, the utilization of online optimization in the form of a quadratic program increase the capabilities of simple Input/Output Linearization by introducing a notion of optimality as well as the ability to distribute torque as necessary to meet actuator requirements. Ultimately the combination of the flexability of the human-inspired optimization along with the controllers described result in not only multi-domain human-like walking, but even more importantly a tool for rapidly designing new walking gaits.

DEDICATION

To my parents Jim and Pam Lack, who have always pushed me to work hard, to be the best at everything I do, and to never quit.

ACKNOWLEDGEMENTS

I would first like to thank my advisor Dr. Aaron Ames for giving me the opportunity to be a part of AMBER lab and for his guidance. Working in the lab has allowed me to work with and learn from some truly brilliant individuals, and for this I am very grateful. I would also like to thank my wife Erin for supporting me every step of the way and for always being eager to listen to me talk about my research.

NOMENCLATURE

$\mathcal{HC}$	Hybrid Control System
Γ	Directed Graph
D	Domain describing the set of admissible configurations
U	Set of admissible controls
S	Set of guards
Δ	Set of reset maps
FG	Control system
V	Set of vertices in the directed graph Γ
E	Set of edges in the directed graph Γ
v	A specific vertex in the set of vertices
e	A specific edge in the set of edges
$\mathbb{R}$	Set of real numbers
u	vector of actuator torques
oa	Over actuation
fa	Full actuation
ua	Under actuation
ts	Toe Strike
hl	Heel lift
hs	Heel strike
Q	Generalized robot coordinates
Q_r	Robot body coordinates
sh	Stance heel
st	Stance toe

nsh	Nonstance heel
nst	Nonstance toe
C	Set of possible contacts
c_{oa}	Contact set for over actuation
c_{fa}	Contact set for full actuation
c_{ua}	Contact set for under actuation
n	Total number of degrees of freedom
n_{c_v}	Number of holonomic constraints in the domain v
m_r	Number of actuators on the robot
TQ	Tangent Bundle
F	Contact wrench of forces and moments
d_{fa}	Constant used to align time with zero at the start of the fully actuated domain
d_{ua}	Constant used to align time with zero at the start of the under actuated domain
$\mathcal{A}_v$	Unilateral constraints on the domain v
sor	Denoting a source domain
tar	Denoting a target domain
$\mathcal{R}_{oa\to fa}$	Relabeling matrix to map over actuated domain to fully actuated domain
$\mathcal{R}_{fa\to ua}$	Relabeling matrix to map fully actuated domain to under actuated domain
$\mathcal{R}_{ua\to oa}$	Relabeling matrix to map under actuated domain to over actuated domain

α	Matrix of controller parameters for the over actuated domain
β_{fa}	Matrix of controller parameters for the fully actuated domain
β_{ua}	Matrix of controller parameters for the under actuated domain
A	Decoupling matrix for Input-Output Linearization control law
y_{oa}	Vector of control outputs for over actuation
y_{fa}	Vector of control outputs for full actuation
y_{ua}	Vector of control outputs for under actuation
τ	Parameterization of time
ε	Controller gain dictating the rate of convergence
$\mathcal{Z}_{ua}$	A full zero dynamics surface for the under actuated domain
$\mathcal{PZ}_{oa}$	A partial zero dynamics surface for over actuation
$\mathcal{Z}_{fa}$	A full zero dynamics surface for full actuation
$\xi_{v,1}$	The position of the hip as a zero dynamics coordinate for a domain v
$\xi_{v,2}$	The velocity of the hip as a zero dynamics coordinate for a domain v
M	Inertia matrix
C	Coriolis matrix
G	Gravity vector
QP	Quadratic Program

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

The long standing, implicit goal of robotics has been to design and control robots to look and behave like humans. Great strides have been made in robotics research that have produced arms and hands capable of anthropomorphic motions with strength comparable to that of humans. In short, the development of robotic grasping and manipulation has greatly eclipsed robotic bipedal locomotion. One example of this is the Sandia Hand [37] produced by Sandia National Labs. The Sandia Hand possess incredible dexterity comparable to that of humans. In contrast, there are a relatively small number of bipedal robots capable of walking reliably, none of which have been capable of walking with anything other than slow, simple gaits. This is, however, not due to a lack of effort, as bipedal robots have been walking for quite some time. Initially it was shown that dynamic walking could be obtained without the use of control at all [26, 10]. The fact that passive approaches could be used to obtain walking is a testament to the fact that there is a great deal of passive energy stored in dynamic walking, and if harnessed correctly and paired with good control, dynamic walking can be achieved with minimal input.

In the field of bipedal locomotion control, a large number of novel approached have been developed. One of the more popular approaches has been to design controllers based on the zero-moment point(ZMP) [22, 57]. The ZMP approach has been successfull in producing surprisingly robust locomotion on a number of humanoid platforms including Honda's Assimo [25], the entire series of HRP humanoids from Kawada Industries(see HRP-2 in [55]), as well as countless other humanoid robots(see [21] for another example). Zero-moment point control is based upon precisely planned gaits centered around balancing the forces and moments at the feet in order to keep the

feet flat.

Another control strategy similar to ZMP based control is capture point control [24, 31]. Capture point control is based upon the assumption that there is a region on the stepping surface in which, if the swing foot is placed, the robot can successfully stop. Capture point methods have certain inherent characteristics which make them especially powerful, that is the ability to directly calculate whether or not it is feasible for the robot to slow down and stop, or if the robot must prepare for a fall. As with ZMP based control, capture point has been successfully realized on multiple platforms including the humanoid M2V2 [32]. ZMP and capture point control are part of a large family of quasi-static approaches to bipedal locomotion, of which the major downfall is that the resulting walking tends to be somewhat inefficient.

Other more mathematically formal methods that have been developed more recently include geometric reduction [16, 46], control symmetries [49], and hybrid zero dynamics [4, 3, 16]. These control approaches are designed to allow for formal notions of stability for locomotion. Geometric reduction and control symmetries have yet to be realized experimentally; however, hybrid zero dynamics based control has seen great success on multiple platforms such as RABBIT [9], MABEL [52], ATRIAS [20], and Amber 1 [36], the point footed predecessor to the robot discussed in this thesis. Extraordinarily, MABEL has also achieved human-like running using control based on hybrid zero dynamics [52]. RABBIT, MABEL, and Amber 1 are all 2-D planar bipeds with point feet; therefore, they are only able to take the notion of being human-like so far. Controllers designed based on hybrid zero dynamics have been shown to work on footed robots as well, such as the 2-D footed biped Amber 2 which is the robot model used to produce the results in this thesis and can be seen in Figure 1.1. Human-inspired hybrid zero dynamics based controllers have also been shown to give dynamic walking in 3 dimensions as well, as demonstrated on the NAO

humanoid robot [11, 4].

Nearly all of the locomotion approaches discussed thus far, despite their success, fail to address one very important aspect of human walking, and that is the fact that, when humans walk, there are often multiple points in contact with the ground at a given time. Furthermore during locomotion the contact points are constantly changing. This multi-contact problem along with having to model the impacts with the environment are just two of the many issues that make physically realizable human-like walking extremely difficult to achieve both in simulation or experimentally [44, 45, 46]. A small amount of multi-contact theory has been done in [40] and [61]; however, these only address multi-contact points for full body control during static behaviors which is very different than controlling a biped through multi-contact behaviors while maintaining the dynamic stability inherent to human locomotion. In [25], multiple contacts are dealt with for bipedal balancing, but not for dynamic walking which is of course very different than balancing.

Not only does the human gait consist of multi-contact phases (formally referred to as over actuation), but there are also phases of both full actuation and under actuation. Full actuation and under actuation are two cases that are encountered much more often in controls and thus have been given far more attention than the control of over actuated systems. Under actuation has been studied in the context of bipedal walking for some time in cases such as point footed walking in [17, 41, 59, 1] to name just a few. The phase of footed human-like walking that is under actuated is the portion of the gait in which the stance toe is the only point in contact with the ground and the stance heel is lifting. In [8], regulation of the ZMP point is used to achieve the under actuated heel lift domain. Unfortunately, over actuated systems have not been studied as extensively as have systems with full and under actuation. An over actuated system is one which has more actuators than degrees of

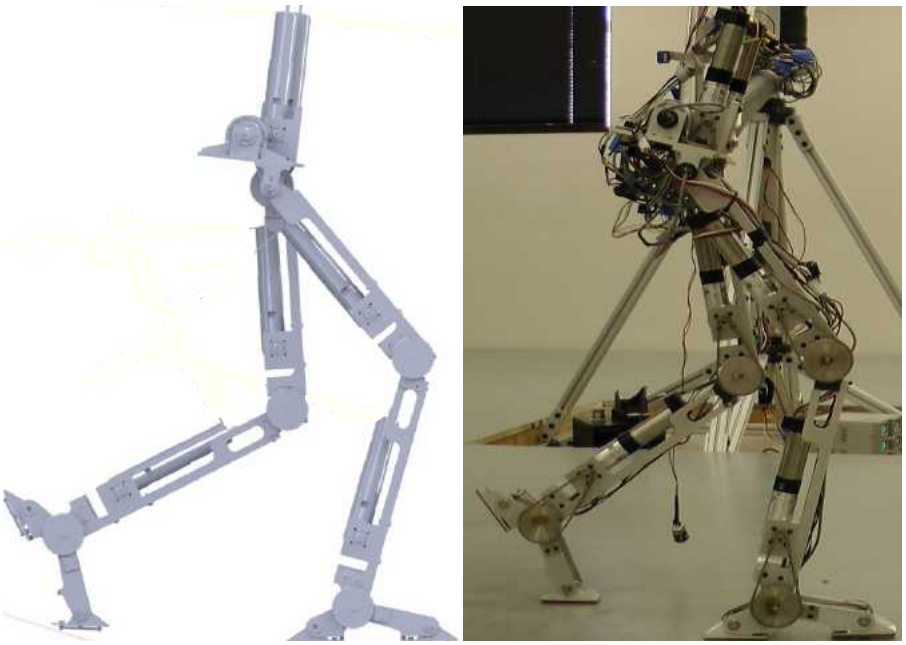

Figure 1.1: Amber 2 is a footed, planar robot designed in AMBER Lab at Texas A&M University. A SolidWorks render (left) is shown as well as the actual robot (right).

freedom. The difficulties that arise in the control of these systems are that there are potentially an infinite number of ways to apply actuation that will achieve a given control objective. In this case, the goal becomes how the control objective can be met in an optimal manner.

When designing controllers for human-like locomotion, the control strategy is often to first disect the human gait into phases and design the controllers based on these phases. These phases are generally based on the impacts and contact conditions, however, bipedal robotic locomotion has been achieved by using a number of different phase breakdowns. The most simple domain breakdown which has been studied extensively is the single phase, compass gait biped [48, 12, 15]. As the complexity of the model is increased, often so is the number of phases, referred to as domains for the purposes of this thesis. The point footed model with knees and feet has been considered as both a single domain or as two domains in which

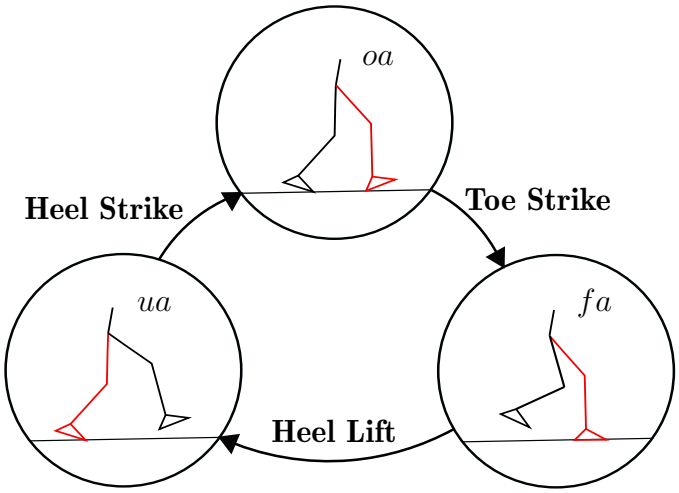

Figure 1.2: Hybrid domain structure for 3-Domain multi-contact walking.

the swing phase is divided into pre and post knee lock [45]. Adding feet allows for an even more complex and humanlike domain breakdown by having as many as four [44] or even five [45] domains. The domain breakdown considered for this thesis is shown in Figure 1.2 and consists of three domains including a multi-contact phase(over actuation), a single support phase with a flat stance foot(full actuation), and a single support phase in which the robot is rotating about the stance toe while the nonstance leg is swinging(under actuation).

The remainder of this thesis will present a method for achieving multi-phase, planar bipedal locomotion including phases of under actuation, full actuation, and over actuation, as well as a simple method for navigating through over actuated domains with minimal actuator requirements. Additional considerations that must be considered when designing controllers for bipedal locomotion are ground contact forces, usually referred to as lagrange multipliers in bipedal locomotion [56]. When designing controllers for locomotion, one must take care in ensuring that the forces, enforced as holonomic constraints, are indeed physically realizable. For example in

considering the 2D planar robot with feet, during the fully actuated phase, one must make sure that the stance foot does not roll on either the heel or toe prematurely [16].

This thesis leverages a hybrid systems approach to bipedal walking as well as the principle of hybrid zero dynamics. The control design approach is inspired by data gathered in human walking experiments. In order to achieve 3-domain human-like walking, a human inspired optimization will be used to design surfaces that have encoded in them the desired behavior of the robot. Using these surfaces and the control algorithm Input/Output(IO) Linearization [39], the human-inspired optimization will be shown to be a an exceptional tool for gait design due to the ease with which gaits can be altered while remaining physically realizable. Also shown will be a method for overcoming various issues inherent to over actuated control.

The remainder of this thesis will be structured as follows. Chapter 2 will introduce hybrid systems and discuss how the hybrid systems approach is useful in modeling multi-domain systems with both continuous and discrete dynamics. Chapter 3 discusses the dynamic model of Amber 2 as well as the method used for modeling impacts. Chapter 4 details how the use of motion capture data obtained from human walking experiments is used in human-inspired control. In Chapter 5, a human-inspired optimization is introduced which utilizes the data from the walking experiments discussed in Chapter 4. Chapter 6 shows how online optimization via quadratic programs can be used to introduce a notion of optimality to the human inspired controllers, and in Chapter 7, simulation results from a walking gait obtained from the human inspired optimization will be presented as well as a short discussion of future work.

2. HYBRID SYSTEMS

Due to the fact that walking robots are systems with impacts, a hybrid systems approach will be used; however, other methods are available for modeling such systems such as differential inclusions [47]. Hybrid systems are defined as systems with both continuous and discrete dynamics; therefore, it is natural to use a hybrid systems approach when modeling walking robots. In addition it is a well studied area even in the context of bipedal robots [60, 59] and has been shown to yield successful experimental results on both planar and 3 dimensional robots [36, 4].

The formal model of a bipedal robot with a multi-contact multi-domain walking gait follows the general development given in [44], In particular, consider a *hybrid control system model* given by a tuple:

$$\mathcal{HC} = (\Gamma, D, U, S, \Delta, FG), \tag{2.1}$$

where

- $\Gamma = (V, E)$ is a directed graph, where V is the set of vertices and $E \subset V \times V$ is the set of edges; an edge $e \in E$ can be written as $e = (i \to j)$, and the source of e is $sor(e) = i$ and the target of e is $tar(e) = j$.
- $D = \{D_v\}_{v \in V}$ is a set of domains, where $D_v \subseteq \mathbb{R}^{n_v} \times \mathbb{R}^{m_v}$ is a smooth submanifold of $\mathbb{R}^{n_v} \times \mathbb{R}^{m_v}$ (with $\mathbb{R}^{m_v}$ representing conrol inputs),
- $U = \{U_v\}_{v \in V}$, where $U_v \subset \mathbb{R}^{m_v}$ is a set of admissible controls,
- $S = \{S_e\}_{e \in E}$ is a set of guards, where $S_e \subseteq D_{sor(e)}$,

- $\Delta = \{\Delta_e\}_{e \in E}$ is a set of reset maps, where $\Delta_e : \mathbb{R}^{n_{sor(e)}} \to \mathbb{R}^{n_{tar(e)}}$ is a smooth map,

- $FG = \{(f_v, g_v)\}_{v \in V}$, where (f_v, g_v) is a control system on D_v, i.e. , $\dot{x} = f_v(x) + g_v u$ for $x \in D_v$ and $u \in U_v$.

The remainder of this section will be devoted to defining the specific elements of this hybrid system in the context of the multi-domain walking gait of interest.

Graph Structure. For the multi-contact walking gait of interest, the graph Γ of the hybrid control system $\mathcal{HC}$ is pictured in Figure 1.2. In particular, the discrete structure of the walking gait implies that Γ is a directed cycle, with vertices and edges given by:

$$\begin{aligned} V &= \{oa, fa, ua\} \\ E &= \{ts = (oa \to fa), hl = (fa \to ua), hs = (ua \to oa)\}. \end{aligned} \tag{2.2}$$

where in this case the vertices (also referred to as domains) are labeled by the type of actuation each of these domains possess (as will be discussed later), i.e., the vertices oa, fa and ua correspond to over, full and under actuation, respectively.

Coordinates, Constraints and Actuation Types. The basic terminology related to coordinates, constraints and actuation types that will be necessary to define the hybrid control system (2.1) modeling a planar bipedal robot exhibiting a multi-domain walking gait will now be introduced. In particular, due to the multi-domain structure of the hybrid system model, this involves considering the generalized coordinates of the robot: $Q = \mathbb{R}^2 \times SO(2) \times Q_r$, where Q_r is the (relative) configuration space of the robot characterized by the relative angles of the system. It is assumed that a subset of Q_r are chosen so that there is a well-defined coordinate system,

i.e., so that Q_r is embeddable in $\mathbb{R}^{n_r}$, $Q_r \subset \mathbb{R}^{n_r}$, with coordinates expressed as $q_r \in Q_r$. The generalized coordinate space $Q \subset \mathbb{R}^n$, can be expressed in coordinates as $q = (p^T, \varphi_0, q_r)^T$, where $p = (p_x, p_z)^T$ is a position and $\varphi_0 \in SO(2)$ is an angle expressing the position and orientation of a reference frame, R_b, attached to the body of the robot relative to a world frame R_0.

Contact Conditions: For each vertex of the graph Γ, there are associated contact points interacting with the physical world that dictate multi-contact conditions in the robot. This is represented by a set of contact points $C = \{sh, st, nsh, nst\}$, indicating which points on the robot can, or are, interacting with the world. Since the robot model is that of a planar robot locomoting, one need only consider the contact points associated with the feet, i.e., sh is the stance-heel, st is the stance-toe, nsh is the non-stance heel and nst is the non-stance toe. Associated with the indexing set of contact points are two types of constraints: *unilateral* and *holonomic*. Unilateral constraints, denoted by h_v for $v \in V$, are a vector-valued function that dictates the admissible configurations of the system on each domain by codifying which contact points are not on the ground and must stay above the ground as well as points that are in contact with the ground and must continue pressing into the ground. Conversely, holonomic constraints, denoted by η_v for $v \in V$, is a vector-valued function encoding which contact points are in contact with the ground and, therefore, must be held constant.

Example 1 *To provide a specific example, for the domain structure considered in this thesis (see Figure 1.2) there are the following constraints for each domain:*

- *For $v = fa \in V$, $h_{fa}(q)$ consists of the vertical reaction force at the stance heel, and $\eta_{fa}(q) \in \mathbb{R}^3$ consists of the x, z position of the stance toe together with the z position of the stance heel.*

- *For $v = ua \in V$, $h_{ua}(q)$ consists of the z position of the non-stance heel, while $\eta_{ua}(q) \in \mathbb{R}^2$ consists of the x, z position of the stance toe.*

- *For $v = oa \in V$, $h_{oa}(q)$ consists of the z position of the stance toe, while $\eta_{oa}(q) \in \mathbb{R}^4$ consists of the x, z position the stance heel and the x, z position of the stance toe.*

Actuation Type: With notions of coordinates and constraints in hand, explicit definitions of what is meant by full, over, and under actuation are needed. First, let m_r denote the number of actuators of the robot, and n_{c_v} denote the number of holonomic constraints in a given domain. For a domain, $v \in V$, the type of actuation is

- *Fully-actuated* if $m_r = n - n_{c_v}$,

- *Under-actuated* if $m_r < n - n_{c_v}$,

- *Over-actuated* if $m_r > n - n_{c_v}$.

Similarly, V_{fa} is defined to be the set of full-actuated domains, V_{ua} to be the set of under-actuated domains, and V_{oa} to be the set of over-actuated domains.

Example 2 *For a floating base model (See Chapter 3) of AMBER 2, $n = 9$ and $m_r = 6$. For $v = oa$, $n_{c_{oa}} = 4$ thus $n - n_{c_{oa}} = 9 - 4 = 5 < 6$ and the robot is over-actuated. For $v = fa$, $n_{c_{fa}} = 3$ thus $n - n_{c_{fa}} = 9 - 3 = 6$; therefore, the robot is said to be fully-actuated. For $v = ua$, $n_{c_{ua}} = 2$ thus $n - n_{c_{ua}} = 9 - 2 = 7 > 6$; therefore, the robot is under-actuated. Note that it will be useful to refer to this example when reading the control section where controllers will be developed specifically based upon the type of actuation.*

3. ROBOT MODELING

Modeling bipedal robots is similar to modeling any kinematic chain in that they are simply a series of rigid bodies connected by a series of prismatic and/or rotational degrees of freedom. Two popular techniques for modeling these systems include The Denavit-Hartenberg [42] method and the method of Exponential Twists [28]. Both approaches are much less laborious than the standard Newtonian method, and are equally viable. For high degree of freedom systems, numerical techniques such as *Spatial Vector Algebra* [13] become the favored approach due to the slow computational speed of the closed form dynamics.

The remainder of this chapter will discuss all aspects of the bipedal robot model by first deriving in detail the process of using Lagrangian mechanics to obtain the dynamic model for a planar biped with feet. It will then be shown how to calculate and apply both unilateral and holonomic constraints as well as verify that the constraints are satisfied. The discrete phase dynamics will then be derived, followed by an explanation of the impact map and relabeling matrix.

3.1 Floating Base Model

The bipedal robot model of importance for this paper is a seven link, nine degree of freedom kinematic chain. Some choose to model bipeds as being pinned to the ground at one contact point; however, in order to simplify the overall modeling process and to strive for maximum generality, the biped for this paper will be modeled as floating with nine degrees of freedom (DOF) in which certain contact constraints will be shown later to reduce the mechanical degrees of freedom. In order to do this, it is first necessary to establish a coordinate frame that is fixed to a point on the robot, denoted R_b. This body fixed frame is referenced with respect to a fixed world

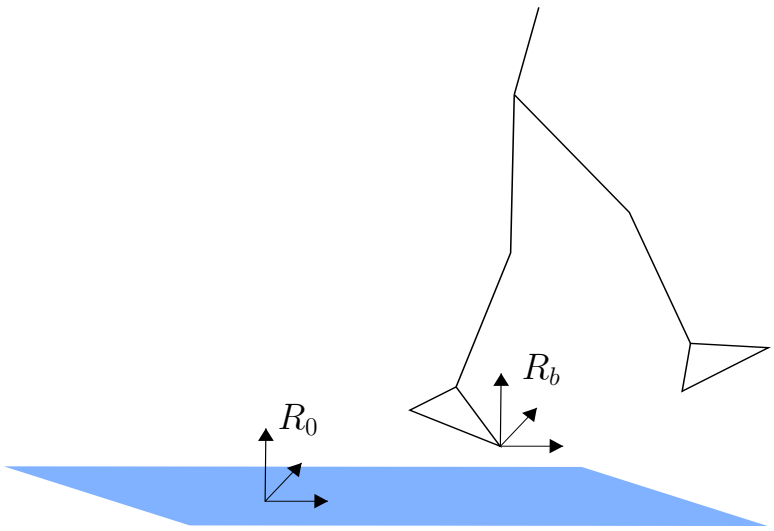

Figure 3.1: In a floating base model, the robot is modeled as a floating chain of links with a body fixed frame R_b referenced relative to a fixed world frame R_0.

frame, R_0. The two orthonormal axes of R_0 denoted as $(\hat{x}, \hat{z})$ are oriented such that $\hat{z}$ is aligned with gravity and pointing upward. The frames R_0 and R_b are shown in Figure 3.1 for clarification. In order to express the position and orientation of R_b with respect to R_0, $p_b = \{p_x, p_z\} \in \mathbb{R}^2$ is defined as the Cartesian position of the body fixed frame, and $\varphi_0 \in SO(2)$ being defined as the orientation. Letting $q_r \in Q_r$ be a vector of body coordinates, it follows that $q = \{p_b, \varphi_0, q_r\} \in Q$ represent a set of generalized coordinates for the robot. For the robot model here, that of AMBER 2, the configuration space Q_r is given by $q_r = \{q_{sa}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk}, q_{nsa}\}$. These coordinates are illustrated in Figure 3.2. Having established a set of generalized coordinates for the robot, a Lagrangian formulation of the dynamics can now be derived. The robot Lagrangian $L(q, \dot{q}) : TQ \to \mathbb{R}$ is of the form,

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q) \tag{3.1}$$

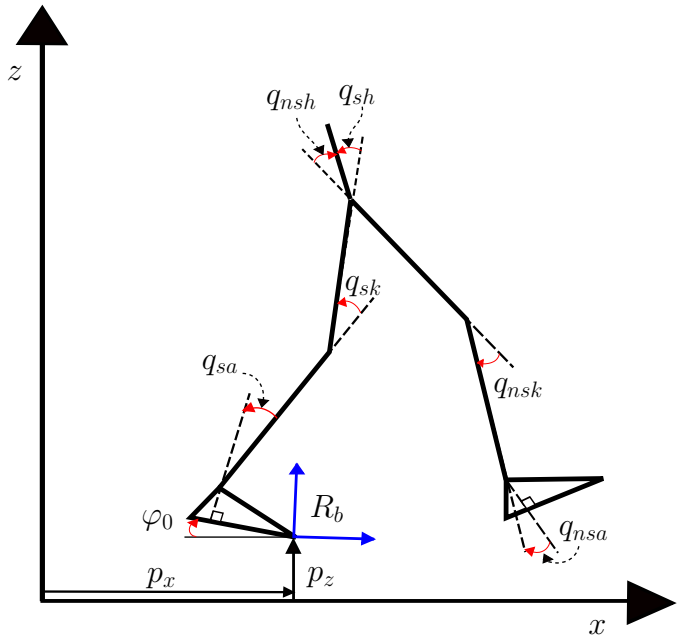

Figure 3.2: The generalized coordinates for the 2 dimensional bipedal robot.

where $T(q, \dot{q})$ is the kinetic energy and $V(q)$ is the potential energy [51, 42, 28]. The kinetic energy, $T(q, \dot{q})$, is given by,

$$T(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} \tag{3.2}$$

where $M(q)$ is the inertia matrix. The equations of motion are then given by substituting (3.1) into Lagrange's equations [28],

$$\frac{d}{dt}\frac{\delta L}{\delta \dot{q}} - \frac{\delta L}{\delta q} = Bu \tag{3.3}$$

where B is the torque distribution matrix which has a rank equal to that of the number of actuated joints, and u is a vector of combined actuator torques. Equation

(3.3) leads to the standard form for the equations of motion for a robot,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu \tag{3.4}$$

where $C(q,\dot{q})$ is the coriolis matrix, and $G(q)$ is a vector describing the effects of gravity.

Without the presence of ground contacts, the dynamics of the system are described by (3.4); however, in the presence of constraints introduced by contacts with the environment, the dynamical model needs to be updated to include these constraints. The following section will introduce the notion of holonomic constraints as well as show how to formally include them into the dynamical model.

3.2 Constraints

During times of single and double support, the ground acts as a constraint on the allowable configurations of the system, leading to such constraints to be also referred to as kinematic constraints. These types of constraints are known formally as *holonomic constraints* [28], which actually constrain the kinematic degrees of freedom of the system. The dynamical model for each contact set can be obtained by imposing these holonomic constraints to the unpinned dynamical model described by (3.4).

3.2.1 Holonomic Constraints

In locomotion, humans use the interaction with ground to both accelerate and decelerate. For a bipedal robot and a given domain $v \in V$, this contact is defined to occur at a small number of points termed *contact points* denoted by $c_v = \{c_1, c_2, \ldots, c_k\}$ where each point c_i is associated with a point on the robot in contact with the environment. A holonomic constraint as briefly introduced in the previous chapter is

then defined as η_v for $c_v \in C$ where for the contact to be maintained the following must be true,

$$\eta_v(q, \dot{q}) = constant, \tag{3.5}$$

with $q \in Q$. For ground contact constraints as seen in bipedal walking, the constraint is expressed as

$$\eta_v(q, \dot{q}) = J_v(q)\dot{q} = constant \tag{3.6}$$

where $J_v(q)$ is the jacobian of the constraint p_{oa}(see Example 3).

In order to impose these constraints on the dynamical model, (3.4) is modified to contain the contact wrench according to the principle of virtual work as demonstrated in [50, 28],

$$M(q, \dot{q})\ddot{q} + H(q, \dot{q}) = Bu + J_v^T(q)F_v(q, \dot{q}, u), \tag{3.7}$$

where for simplicity, $H = C(q, \dot{q})\dot{q} + G(q)$ from (3.4). $F_v(q, \dot{q}, u)$ in (3.7) is known as a Lagrange multiplier and $F_v(q, \dot{q}, u)$ is a multidimensional vector of forces and moments, also called a wrench. In order to fully define the model, the holonomic constraint is differentiated twice which allows the wrench to be solved [16, 44],

$$\dot{J}_v(q, \dot{q})\dot{q} + J_v(q)\ddot{q} = 0. \tag{3.8}$$

Combining (3.8) and (3.7), the contact wrench $F(q)$ can be solved yielding,

$$F_v(q,\dot{q},u) = -(J_v(q)M^{-1}(q)J_v(q)^T)^{-1}(\dot{J}_v(q,\dot{q})\dot{q}+ \\ J_v(q)M^{-1}(q)(Bu - H(q,\dot{q}))), \tag{3.9}$$

where the wrench results in a vector with three forces and three moments for 3-dimensional bipeds and two forces and one moment for 2-dimuensional bipeds. For the 2-dimensional system being considered here the resulting wrench is,

$$F_{v,i}(q,\dot{q},u) = (F^x, F^z, \mathcal{M}^y)^T \qquad i \in c_v \tag{3.10}$$

It is important to note that $J_v(q)$ *must have full row rank*. The following example will help to better clarify the specification of the holonomic constraint(s) for a given domain.

Example 3 *Consider the domain of over actuation(See Figure 1.2) where the stance heel and nonstance toe are in contact with the ground and the robots feet are rotating about each contact point. For this domain,* $c_{oa} = \{sh, nst\}$. *Consider the frames* R_{sh} *and* R_{nst} *located at both the stance heel and nonstance toe with each frames Cartesion position with respect to the world frame denoted by* $p_{sh} = (x_{sh}, z_{sh})^T$ *and* $p_{nst} = (x_{nst}, z_{nst})^T$. *Each is then grouped together yielding* $p_{oa} = (p_{sh}^T, p_{nst}^T)^T$. *This is then differentiated yielding* $J_{oa} = \partial p_{oa}/\partial q$ *which completes the constrained dynamic model for the double support domain in* (3.7)

For convenience, (3.7) can be written as a control system of the form $\dot{x} = f_v(x) +$

$g_v(x)u$ where $x = (q, \dot{q})^T$ and,

$$f_v(q, \dot{q}) = \begin{bmatrix} \dot{q} \\ M^{-1}(q)((J_v^T(q)\Xi(q)J_v(q)M^{-1}(q) \\ -I)H(q, \dot{q}) - J_v^T(q)\Xi(q)\dot{J}_v(q, \dot{q})\dot{q}) \end{bmatrix} \tag{3.11}$$

$$g_v(q) = \begin{bmatrix} \mathbf{0} \\ M^{-1}(q)(I - J_v^T(q)\Xi(q)J_v(q)M^{-1}(q))B \end{bmatrix}, \tag{3.12}$$

where $\Xi(q) = (J_v(q)M(q)^{-1}J_v^T(q))^{-1}$ for simplicity

3.2.2 Unilateral Constraints

Another type of constraint, known as a unilateral constraint, will be used to signify the boundary of certain domains as well as force based constraints that further restrict the set of admissible configurations. Given a particular domain, i.e. a $v \in V$ and a contact set c_v, constraints are used to prevent the robot from pulling against the ground by ensuring $F_{v,i}^z(q, \dot{q}, u) \geq 0$ where $F_{v,i}(q, \dot{q}, u)$ for $i \in c_v$ is a contact wrench. Coupling the force based constraints with the constraint on the domain, $h_v \geq 0$, which for domains in which an impact occurs denotes the height of impacting points and denotes the vertical reaction force of the stance heel for the domain of full actuation (i.e. the heel is lifting), the unilateral constraints are given by,

$$\mathcal{A}_v(q, \dot{q}, u) = \begin{bmatrix} F_{v,i}^z(q, \dot{q}, u) \\ h_v(q) \end{bmatrix} \geq 0. \tag{3.13}$$

3.3 Modeling Impacts

In bipedal robotics as well as other areas of robotics, contact with the environment is unavoidable. In certain areas of robotics it may be possible to ignore such

impacts if the impacting velocity is relatively low thus having a small effect on the overall dynamics of the system. With locomotion, however, the impacts can have a stabilizing effect since they remove energy from the system.

One of the more popular methods for modeling impacts is to assume elastic impacts [58, 33]. Though this may be a somewhat simple approach, it is dependent on guessing coefficients desrcribing the the elasticity of the walking surface. A second method for modeling impacts is to model them as inelastic collisions, which is the approach that will be used in this paper and has been used extensively [3, 43] and is derived in detail in [59].

In order to model impacts as plastic collisions, the force due to the impact is assumed to be an impulse which acts over an *infinitesimal* time interval. This impulse, denoted δf, will result in a discrete jump in the velocities of the system; however, the angles of the system will be continuous through the impact. It is important to note that this method requires prior knowledge of the contact points post impact. Beginning with the constrained dynamical in (3.7) model and adding in an impulse wrench yields,

$$M(q)\ddot{q} + H(q,\dot{q}) = Bu + J_v^T(q)\delta f_v, \tag{3.14}$$

where J_v is derived in the same way as in Example 3. Using the assumption that the robots actuators do not produce impulsive torques, (3.14) simplifies to,

$$M(q)\ddot{q} = J_v^T(q)\delta f. \tag{3.15}$$

Integrating (3.15) yields the following equation in terms of the reaction wrench, F_{imp},

the pre impact joint angles, and the pre and post impact joint velocities,

$$M(q)(\dot{q}^+ - \dot{q}^-) = J_v^T(q)F_{imp}, \tag{3.16}$$

where q^- and q^+ are the pre and post impact joint velocities, respectively, and F_{imp} is obtained by integrating the impulse over the duration of the impact and assuming no rebound or sliding with,

$$F_{imp} = \int_{t^-}^{t+} \delta f(\tau) d\tau. \tag{3.17}$$

It is clear that in (3.16) both $\dot{q}^+$ and F_{imp} are unknown. Another constraint is needed in order to make the equation solvable. To do this, certain assumptions about the impact must be assumed even before the impact occurs. The constraint equation required to complete the model again comes from the fact that the points impacting the ground are to have zero post impact velocity, i.e.

$$J_v(q)\dot{q}^+ = 0 \tag{3.18}$$

where J_v is the jacobian of the constraints to be applied *post impact*. To apply this constraint, some information about the impact is assumed in order to know exactly what J_v needs to be. This can be clarified by the following example.

Following Example 3 and using (3.16), both the post impact velocities and the impulsive forces imparted on the robot by the contact surface can be obtained.

Rearranging the two equations yields,

$$\begin{bmatrix} \dot{q}^+ \\ F_{imp} \end{bmatrix} = \begin{bmatrix} M(q) & -J_v(q) \\ J_v(q) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} M(q)\dot{q}^- \\ \mathbf{0} \end{bmatrix} \tag{3.19}$$

Once the post impact velocities have been calculated, it is also necessary to verify that post-impact state of the system is consistent with the holonomic constraints being applied and the post impact state belongs to the domain of admissibility.

3.4 Reset Map

When modeling bipedal robots, it is common to consider the robot as having a stance and nonstance leg rather than a right and left leg. At a prespecified point in the gait(tyically at an impact), the angles are relabeled and the stance leg becomes the nonstance leg and vice versa. Consider a source domain D_{sor} and target domain D_{tar} as well as a pretransition state x^- that is on the guard of D_{sor}, i.e. $x^- \in (S_{sor} \cap D_{sor})$. The state $x^- = (q^-, \dot{q}^-)^T$ is mapped to a state on the target domain $x^+ \in D_{tar}$ by,

$$x^+ = \Delta_{sor \to tar}(x^-) \tag{3.20}$$

where $\Delta_{sor \to tar}$ represents the application of the impact map and the relabeling matrix. Specifically, $\Delta_{sor \to tar}(x^-)$ is,

$$\Delta_{sor \to tar}(q^-, \dot{q}^-) = \begin{bmatrix} \mathcal{R}_{sor \to tar} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_{sor \to tar} \end{bmatrix} \begin{bmatrix} q^- \\ P(q^-, \dot{q}^-), \end{bmatrix} \tag{3.21}$$

where q^- and $\dot{q}^-$ are vectors of the pre impact positions and velocities and $P(q^-, \dot{q}^-)$ is a vector of the post impact velocities and are computed from (3.19) and isolated

from F_{imp} using the Schur Complement [62],

$$P(q^-, \dot{q}^-) = \Big(I - M^{-1}(q^-)J_{tar}^T(q^-)(J_{tar}(q^-)M^{-1}(q^-)J_{tar}^T(q^-))^{-1}J_{tar}(q^-)\Big)\dot{q}^- \quad (3.22)$$

The relabeling matrix $\mathcal{R}_{sor\to tar}$ is used to swap the stance and nonstance legs. For the walking discussed in this thesis, the legs are swapped at heel strike, thus the relabeling matrix for $sor = ua$ and $tar = oa$ is,

$$\mathcal{R}_{ua\to oa} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.23)$$

while the relabeling matrices $\mathcal{R}_{oa\to fa}$ and $\mathcal{R}_{fa\to ua}$ are simply the identity matrix because no leg swapping occurs during those domain transitions.

3.5 Hybrid System Construction

In this section a hybrid system is constructed using a Lagrangian, a directed cycle, a hybrid model, and a domain breakdown. Recall in Section 3.1 the Lagrangian model was developed for the robot, and along with the holonomic constraints from Section 3.2.1, a control system, FG, of the form $\dot{x} = f_v(x) + g_v(x)u$ is constructed in which $f_v(x)$ and $g_v(x)$ are given in (3.11). With a control system and the contstraints

described in (3.13), the domain of admissibility is expressed as,

$$D_v = \{(q, \dot{q}, u) \in TQ \times \mathbb{R}^{m_v} : \mathcal{A}_v(q, \dot{q}, u) \geq 0\} \tag{3.24}$$

The guard, S_v is simply the boundary of a domain with the vector field pointing outside of the domain. For a particular edge, $e \in E$,

$$S_v = \{(q, \dot{q}) \in D_v : h_v = 0 \text{ and } \dot{h}_v < 0\} \tag{3.25}$$

The impact map, $\Delta_{sor \to tar}(q, \dot{q}) : TQ \to TQ$, derived in detail in Section 3.3 relies on information about the post impact contact conditions and ultimately maps a preimpact state on a source domain to a post impact state on a target domain as shown in (3.20). Each compenent defined here is an element in the tuple describing the hybrid control system for the 3-domain walking considered and when put together the hybrid system takes takes the form defined in Equation (2.1). It is important to understand that for the 3-domain walking discussed in this thesis, the impact map is only applied at heel strike and toe strke. At the transition from full actuation to under actuation there is no impact; therefore, the impact map is not applied at this transition.

4. HUMAN INSPIRED CONTROL

In this section, a notion of human inspired control will be introduced, leveraging knowledge obtained via the analysis of human walking data obtained from motion capture experiments. It will be shown how the human data directly inspires human outputs that can be used in control to achieve human-like locomotion. Controllers utilizing Input-Output Linearization will also be introduced that will allow robot outputs to be driven to referrence trajectories inspired by the human walking experiments, thus resulting in human-like motion.

4.1 Human Walking Experiments

Human locomotion has been studied extensively in the literature (see [34, 54, 18, 6, 19, 35]) and is a natural place to look when designing controllers for bipedal robots. For the purposes of this thesis, a series of motion capture experiments were conducted [5] in which 19 LED sensors were fixed to the body of each test subject in specific locations on the lower and upper body as each subject walked a straight line as in Figure 4.1. Each subject performed 11 trials and there were a total of 9 subjects consisting of 3 females and 6 males. The age of the subjects ranged from 17 to 77, heights from 161-189 centimeters, and weights from 47.6-90.7 kg. Each subject was instructed to walk normally at a comfortable pace. The experiments were conducted using the Phase Space System which captured data at 480 frames per second from 12 cameras each of which was accurate to 1 millimeter.

4.2 Human-Inspired Outputs

Although humans seem to be the most logical place to look for intuition in bipedal robots, human data has not played a large role in robotic walking to date. Works

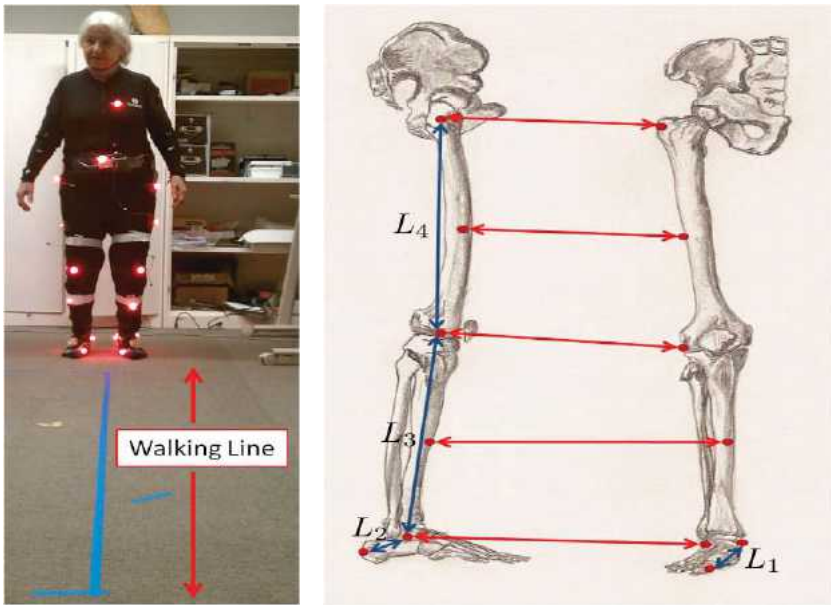

Figure 4.1: Human data experiments were conducted in which 9 subjects walked a straight line as position data was taken using LED sensors placed at strategic locations [5].

such as [4, 36, 46, 45] and the multi-contact gaits shown here are an attempt to span the gap separating the walking seen on bipedal robots today and the highly dynamic gaits humans exhibit. The analysis of the human data showed that there are certain combinations of the joint angles that accurately represent human walking(see [5]). These outputs are shown in Figure 4.3. Using the data and these simple output relationships, it was shown in [5] that the solution to a linear mass spring damper system, a function of relatively simple form, could fit many of the human outputs with a high correlation and thus could be used to define the desired behavior for corresponding robot outputs. This function, termed the *canonical human walking function* for its simple form and ability to accurately represent human walking, is simply the solution to a linear mass spring damper system and is thus a simple

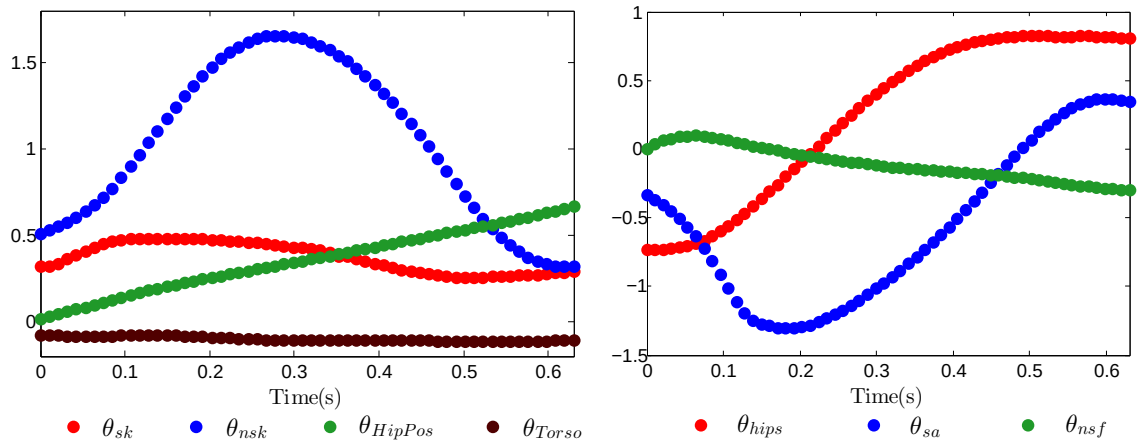

Figure 4.2: Human data taken from walking exeriments. The data sets plotted are for the human inspired outputs shown in Figure 4.3.

combination of sines, cosines, and exponentials,

$$y_{cwf}(\alpha) = e^{-\alpha_4 t}(\alpha_1 cos(\alpha_2 t) + \alpha_3 sin(\alpha_2 t)) + \alpha_5. \tag{4.1}$$

In addition, it is shown in [30] that human data for ascending and descending stairs could be represented equally as well using the solution to a linear mass spring damper system with constant sinusoidal excitation. This function was termed the *extended canonical human walking function*,

$$\begin{aligned} y_{ecwf}(\alpha) &= e^{-\alpha_4 t}(\alpha_1 cos(\alpha_2 t) + \alpha_3 sin(\alpha_2 t)) \\ &+\alpha_5 cos(\alpha_6 t) + \frac{2\alpha_4\alpha_5\alpha_6}{\alpha_4^2 + \alpha_2^2 - \alpha_6^2} sin(\alpha_6 t) + \alpha_7. \end{aligned} \tag{4.2}$$

The principle gain from using these functions to define desired trajectories comes not from fitting human data with a high correlation, but from the fact that the combination of sines, cosines, and exponential terms together create a flexible function that is well behaved even outside the desired region of operation. Using arbitrarily high

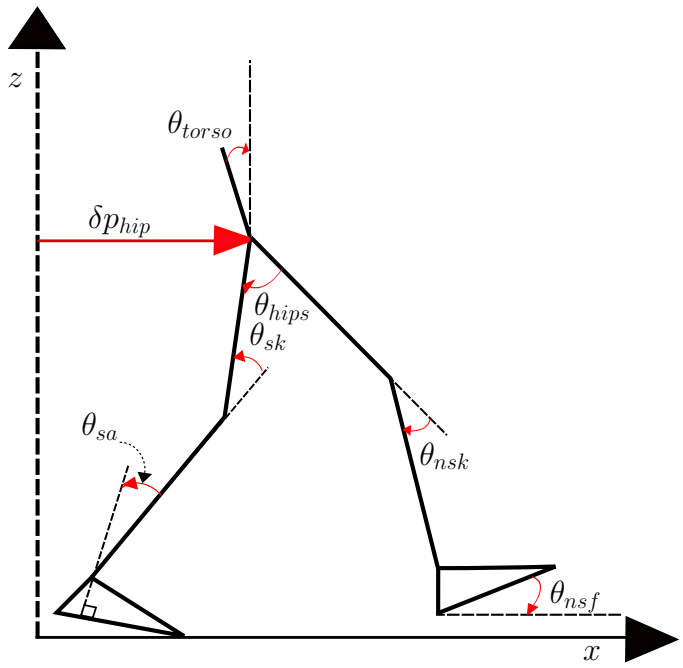

Figure 4.3: Outputs used in the control design. Some outputs such as the knee angles are body coordinates while some are combinations of body coordinates such as the torso angle, the angle between the thighs, and angle of the nonstance foot with the ground.

order polynomials, one could fit any set of human data. Bezier polynomials have in fact successfully been used to yield planar, humanlike bipedal walking *and* running on both RABBIT, MABEL, and Atrias [9, 52, 20]. In order to sufficiently represent all of the human outputs, however, polynomials of order 9 or possibly higher must be used. This results in a curve with very good behavior inside the designed region of operation and very poor behavior even slightly outside the desired region. This can be seen in Figure 4.4 in which a 9th order polynomial and the human canonical walking function are used to fit the human data for the nonstance knee over the course of one step. Also plotted in Figure 4.4a is the same data fit using the canonical human walking function. Comparing the fits in Figure 4.4 reveals the major advantage of using the canonical walking function instead of polynomials. The

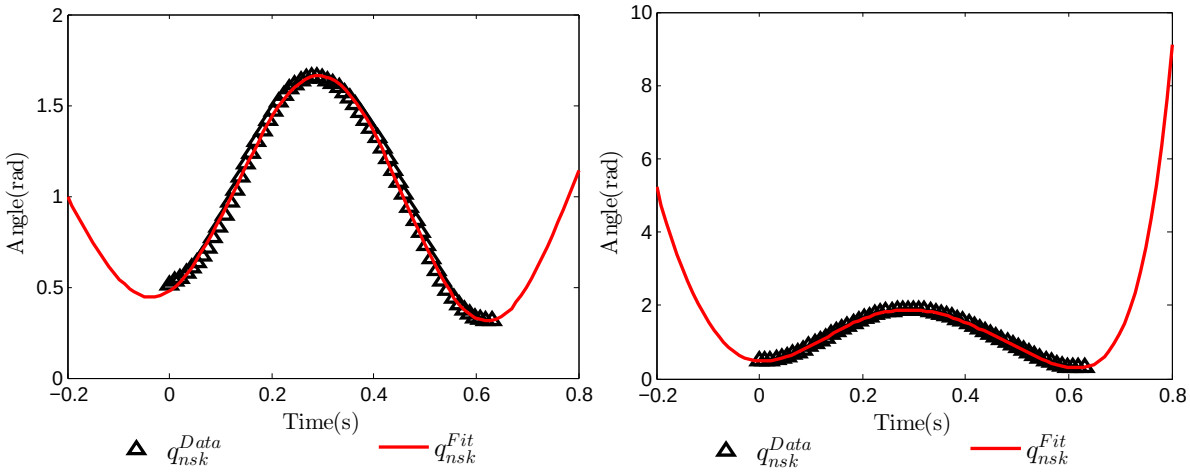

Figure 4.4: Non-stance knee data from a single step(heel strike to heel strike) fit with the canonical walking function(left) and a 9th order polynomial(right). Both functions fit the data very well, yet the polynomial quickly blows up outside the data bounds while the canonical human walking function remains well behaved.

9th order polynomial does produce a slightly better fit to the data; however, the behavior of the polynomial even slightly outside the data is poor. It is clear that the canonical walking function fits the data well and behaves reasonably well outside the data. This is an invaluable characteristic of a referrence trajectory due to the inevitable fact that during robotic locomotion, unexpected events will undoubtedly occur that will take the robot outside the designed region of operation. Ultimately the behavior of these referrence signals outside the designed region could mean the difference between a light stumble or a major failure capable of breaking a robot.

Other function forms have also been used including Gaussian curves and sinusoids [44] to name a few. Though sinusoids and Gaussian's may not be subject to the same downfalls as polynomials, these curves are only able to fit a small number of human outputs and add little to the basic understanding of how humans behave during locomotion. With the introduction of the canonical and extended canonical

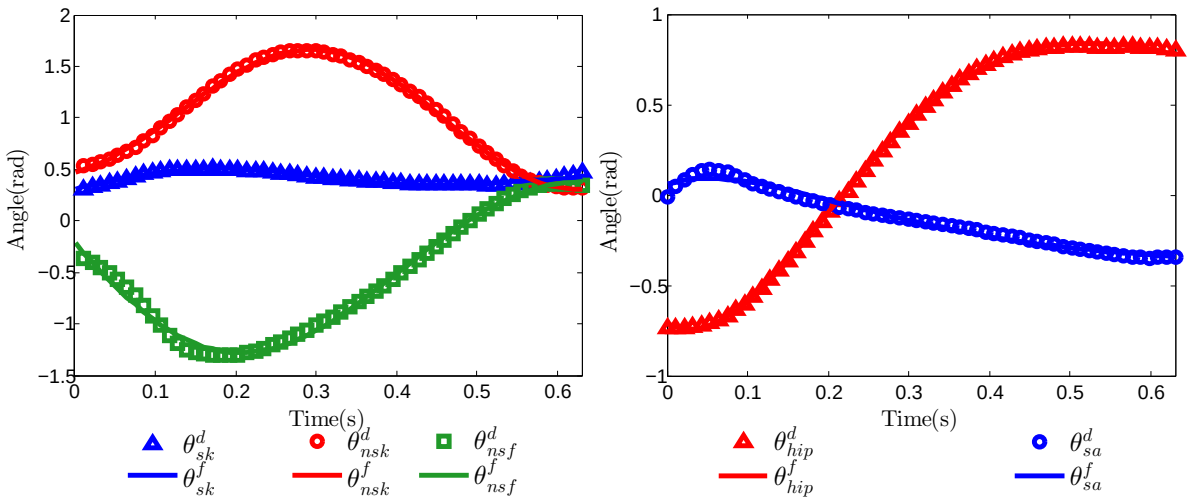

Figure 4.5: Human data for the stance and nonstance knees and the non-stance foot angles on the left and the hip angle and stance ankle angle on the right as well as the fits to each one with the canonical walking function.

walking functions in [5], it was shown that *humans behave like mass spring damper systems while locomoting*, a very important find considering it has been long known that humans do not behave like rigid systems [27, 7]. Furthermore, the fact that this function can be used to very accurately fit nearly all of the human locomotion outputs further increases the functions usefullness. Shown in Figure 4.5 are human walking data for the stance and nonstance knees, the stance ankle angle, and the hip angle with their respective fits using the canonical human walking function. The corresponding robot outputs and their conventions are shown in Figure 4.3. The fits to the human data for each control output are plotted in Figure 4.5, and the mathematical description of each output is given in Section 4.4.

4.3 Human-Inspired Control

The ultimate goal in achieving human-like robotic walking is to apply control such that the chosen robot outputs are driven to match the corresponding human inspired reference trajectories, i.e. ensuring that $y_a(q) \to y_d(t)$ as $t \to \infty$. To do

this, a human-inspired optimization will be used to automatically generate controller parameters that define these referrence trajectories while simultaneously ensuring that the resulting multiphase walking is physically realizable through constraints. Before discussing the human-inspired optimization it will be necessary to first discuss Input/Output Linearization and the various properties that make it suitable for robotic locomotion.

4.3.1 Input/Output Linearization

Input/Output Linearization is a nonlinear control method that results in the input-output relationship of a nonlinear system rendered exactly linear, as apposed to the standard jacobian linearization [14]. This is accomplished through nonlinear state feedback (a full derivation and discussion of Input/Output Linearization is given in chapter 9 of [39]). Given the affine control system in (3.11) for each $v \in V$ representing a mechanical control system with y_v a vector of relative degree one and relative degree two control outputs defined by the error between the actual outputs and corresponding referrence trajectories,

$$y_v = \begin{bmatrix} y_{v,1}(x) \\ y_{v,2}(x) \end{bmatrix} = \begin{bmatrix} y^a_{v,1}(x) \\ y^a_{v,2}(x) \end{bmatrix} - \begin{bmatrix} y^d_{v,1}(x) \\ y^d_{v,2}(x) \end{bmatrix}, \tag{4.3}$$

Differentiating (4.3) to get $\dot{y}_v$ yields,

$$\dot{y}_v = \frac{\partial y_v}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial y_v}{\partial x}\Big[f_v(x) + g_v(x)u\Big] \coloneqq L_f y_v + L_g y_v u, \tag{4.4}$$

where $L_f y_v$ and $L_g y_v$ are the Lie derivatives of y_v with respect to $f_v(x)$ and $g_v(x)$, respectively. For the relative degree one outputs, $y_{v,1}$ the control law,

$$u_{v,1} = \frac{1}{L_g y_{v,1}} \Big[- L_f y_{v,1} + \nu_1 \Big], \tag{4.5}$$

yields the following linear relationship between the new input ν and the relative degree one outputs,

$$\dot{y}_{v,1} = \nu_1. \tag{4.6}$$

For the relative degree two outputs, $L_g y_{v,2}$ will be equal to zero; Therefore, $y_{v,2}$ is differentiated again yielding,

$$\ddot{y}_{v,2} = \frac{\partial L_f y_{v,2}}{\partial x} \frac{\partial x}{\partial t} \coloneqq \frac{\partial L_f y_{v,2}}{\partial x} f_v(x) + \frac{\partial L_f y_{v,2}}{\partial x} g_v(x) u, \tag{4.7}$$

The resulting control law for the relative degree two outputs becomes,

$$u_{v,2} = \frac{1}{L_g L_f y_{v,2}} \Big[- L_f L_f y_{v,2} + \nu_2 \Big]. \tag{4.8}$$

The new inputs v_1 and v_2 are chosen to be the following polynomials,

$$\begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} = \begin{bmatrix} \psi_0 y_{v,1} \\ \phi_1 L_f y_{v,2} + \phi_0 y_{v,2}. \end{bmatrix} \tag{4.9}$$

The coefficients ψ_0 and $\{\phi_0, \phi_1\}$ of (4.9) are chosen to place the roots of the polynomials,

$$\begin{aligned} s + \psi_0 &= 0 \\ s^2 + \phi_1 s + \phi_0 &= 0 \end{aligned} \tag{4.10}$$

at $-\varepsilon$, ensuring the two polynomials are *Hurwitz*. The resulting control law for both relative degree one and relative degree two outputs becomes,

$$u_v = -A^{-1}\left(\begin{bmatrix} 0 \\ L_f L_f y_{v,2} \end{bmatrix} + \begin{bmatrix} L_f y_{v,1} \\ 2\varepsilon L_f y_{v,2} \end{bmatrix} + \begin{bmatrix} \varepsilon y_{v,1} \\ \varepsilon^2 y_{v,2} \end{bmatrix} \right) \tag{4.11}$$

where A_v, referred to as the decoupling matrix, for an output combination consisting of both relative degree one and relative degree two outputs is defined to be,

$$A_v = \begin{bmatrix} L_g y_{v,1} \\ L_g L_f y_{v,2} \end{bmatrix}. \tag{4.12}$$

For the case in which an output combination contains only relative degree two outputs, the control law becomes

$$u_v = -A_v^{-1}\Big(L_f L_f y_{v,2} + 2\varepsilon L_f y_{v,2} + \varepsilon^2 y_{v,2} \Big) \tag{4.13}$$

where for this case the decoupling matrix is,

$$A_v = L_g L_f y_{v,2} \tag{4.14}$$

It is important to note that the decoupling matrix must have full row rank since it is inverted. This is done through carefully choosing the outputs to ensure that they are mutually exclusive. The result is that $y_v \to 0$ exponentially at a rate of $-\varepsilon$. The next sections will discuss the output selection for each domain.

4.4 Human Outputs

The first step to designing human walking controllers is to choose a set of outputs that adequately represent human walking. The human inspired output functions used here are shown in Figure 4.3, and many of which have been proven to successfully yield robotic walking [36]. Motivated by both the desire to control the walking speed, the linearized forward velocity of the hip was chosen as an output and is given by,

$$\delta v_{hip} = \frac{\delta p_{hip}}{\partial q}\dot{q}, \tag{4.15}$$

where,

$$\delta p_{hip} = -L_c(\varphi_0 + q_{sa}) - L_t(\varphi_0 + q_{sa} + q_{sk}). \tag{4.16}$$

The coordinates q_{sa}, q_{sk}, and φ_0 are as defined in Figure 3.2 and L_c and L_t are the lengths of the thigh and calf links as shown in the figure on page 60.Also chosen as outputs are the angle between the legs,

$$\theta_{hips}(q) = q_{sh} - q_{nsh}, \tag{4.17}$$

the stance ankle angle, the stance and nonstance knee angles, the nonstance foot angle given by,

$$\theta_{sa}(q) = q_{sa} \tag{4.18}$$

$$\theta_{sk}(q) = q_{sk} \tag{4.19}$$

$$\theta_{nsk}(q) = q_{nsk} \tag{4.20}$$

$$\theta_{nsf}(q) = \varphi_0 + q_{sa} + q_{sk} + q_{sh} - q_{ns} - q_{nsk} - q_{nsa} \tag{4.21}$$

and finally the torso angle,

$$\theta_{torso}(q) = \varphi_0 + q_{sa} + q_{sk} + q_{sh}. \tag{4.22}$$

A combination of each of these outputs will be used to control the robot through each domain. It is important to note the linear form of each output, significance of which will be made clear later.

4.4.1 Human Locomotion(Actual) Outputs

Using the data in 4.2 from human walking experiments, if was found that certain combinations of human joint angles exhibit simple behavior. These outputs, termed human-inspired outputs, allow for a low dimensional representation of human complex human locomotion. Combinations of these outputs, shown in Figure 4.3, are used to control the robot through each domain. It is important that each combination of outputs for any particular domain is linearly independent to ensure that the decoupling matrix has full row rank.

Definition 1 *A human output combination for $v \in V$ is a tuple $Y_v^H = (Q_r, y_v^H)$ consisting of the configuration space Q_r and a set of robot outputs $y_v^H : Q_r \to \mathbb{R}$. A*

particular output combination is linearly independant if,

$$rank(y_v^H(q,\dot{q})) \leq n - c_v. \tag{4.23}$$

where n and c_v are defined in Chapter 2.

Keeping Definition 1 in mind, the outputs for each domain will now be shown.

4.4.1.1 Over-Actuation

During over actuation (see the domain graph in Figure 1.2 in Chapter 1), there are 4 contact constraints, i.e. $c_{oa} = 4$. With the number of degrees of freedom $n = 9$, this makes the number of degrees of freedom during over actuation to be $n_{oa} = n - c_{oa} := 5$. Since the has robot six actuators, $n_r > n_{oa}$ means the robot is over actuated and is only capable of independantly tracking n_{oa} trajectories. With this in mind, the outputs for over actuation are,

$$y_{oa,1}(q,\dot{q}) = \delta v_{hip}, \qquad y_{oa,2}(q) = \begin{bmatrix} \theta_{sk}(q) \\ \theta_{tor}(q) \\ \theta_{hips}(q) \\ \theta_{nsk}(q) \end{bmatrix} \tag{4.24}$$

4.4.1.2 Full Actuation

For the fully actuated domain, i.e. $v = fa$, the robots stance foot is flat on the ground and the nonstance leg is swinging. This phase is characterized by $c_{fa} = 3$, resulting in the number of degrees of freedom being $n_{fa} = n - c_{fa} = 6$ which is equal to the number of actuators on the robot. During full actuation the robot can

independantly track six outputs, resulting in the following choice of outputs,

$$y_{fa,1}(q,\dot{q}) = \delta v_{hip}, \qquad y_{fa,2}(q) = \begin{bmatrix} \theta_{sk}(q) \\ \theta_{tor}(q) \\ \theta_{hips}(q) \\ \theta_{nsk}(q) \\ \theta_{nsf}(q) \end{bmatrix}. \tag{4.25}$$

It is important to note that the outputs for full actuation are the same as for over actuation with the addition of the nonstance foot angle.

4.4.1.3 Under Actuation

During the over actuated phase, the robot is rotating about its stance toe and the heel is lifting. In this configuration $c_{ua} = 2$, resulting in $n_{ua} = n - c_{ua} = 7$. With $n_{ua} > n_r$, the robot is in an under actuated configuration. During under actuation(see the domain graph in Figure 1.2), the robot does not have direct control over the forward hip velocity; therefore, during under actuation only relative degree two outputs are controlled,

$$y_{ua,2}(q) = \begin{bmatrix} \theta_{sa}(q) \\ \theta_{sk}(q) \\ \theta_{tor}(q) \\ \theta_{hips}(q) \\ \theta_{nsk}(q) \\ \theta_{nsf}(q) \end{bmatrix}. \tag{4.26}$$

The linear form of the outputs used in each domain allows the outputs to be written in a simple form as demonstrated in the following example.

Example 4 *Using linear outputs allows for a particular output combination to be written in a relatively simple form. For a domain with both relative degree one and relative degree two outputs, i.e. over actuation and full actuation, the outputs can be written as*

$$\begin{aligned} y^H_{oa,1} &= \rho_{oa}\dot{q} \\ y^H_{oa,2} &= H_{oa}q \\ y^H_{fa,1} &= \rho_{fa}\dot{q} \\ y^H_{fa,2} &= H_{fa}q \end{aligned} \tag{4.27}$$

where,

$$\rho_{oa} = \rho_{fa} = \begin{bmatrix} 0 & 0 & 0 & (-L_c - L_t) & -L_t & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.28}$$

$$H_{oa} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$H_{fa} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}$$

where L_c and L_t are the lengths of the calf and thigh as shown in the figure on page 60. The under actuated domain, however, does not have any relative degree two

outputs. This results in the outputs being written as,

$$y_{ua}^{H} = H_{ua} q \tag{4.29}$$

where,

$$H_{ua} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}$$

From the definition of a valid output combination in Definition 1, an output combination is valid if $rank(y_v^H) \leq n - c_v$. For $v = oa$, $rank(y_{oa}^H) = 5$ and $c_{oa} = 4$. With $n = 9$ for each domain, thus the output combination for y_{oa} given in (4.24) passes as a valid combination. For $v = fa$, $rank(y_{fa}^H) = 6$ and $c_{fa} = 3$, thus again $rank(y_{fa}) = n - c_{fa}$ and the outputs for the fully actuated domain given in (4.25) also pass as a valid output combination. Finally, for the underactuated domain, $rank(y_{ua}^H) = 6$ and $c_{ua} = 2$, resulting in $rank(y_{ua}) < n - c_{ua}$ implying that the output combination for under actuation given in (4.26) is valid.

4.4.2 Paramaterization of Time

It is widely known that non-time based controllers, i.e. autonomuos controllers, possess an inherent robustness that is difficult to achieve through time dependent controllers. With this in mind, time is parameterized using the evolution of the forward position of the hip, as it was shown in the human data(see Figure 4.2) to

evolve in a linear fashion yielding,

$$\tau_v(q) = \frac{p_{hip,v}(q) - d_v}{v_{hip,v}}, \tag{4.30}$$

where $d_v \in \mathcal{R}$ is used to align $\tau_v(q) = 0$ at the beginning of each domain. This parameterization is then substituted into Equations (4.1) and (4.2) yielding fully autonomous referrence trajectories.

The term d_v is a parameter chosen by the optimization to decide the initial forward position of the hip in each domain, which is directly related by Equation (4.30) to the duration of a domain. It is important to note that d_{oa} is predetermined using inverse kinematics, and thus it is not optimized (this will be made more clear in Section 5.2).

4.4.3 Desired Output Functions

It was stated earlier that during locomotion, humans behave as linear spring mass damper systems. With this in mind, a set of referrence trajectories can now be specified for the human outputs discussed in Section 4.4.1. With the goal of controlling the walking speed, the desired referrence trajectory for the relative degree one output becomes,

$$y^d_{v,1} = vhip_v \qquad v \in \{oa, fa\} \tag{4.31}$$

It is important to note again that there are no relative degree one outputs for the under actuated domain. For the relative degree two outputs during over actuation, i.e. $v = oa$, the goal of behaving like a mass spring damper systems motivates the choice of the canonical walking function. Defining an indexing set O_v for $y^d_{v,2}$ and

$v = oa$, the referrence trajectories for double support are,

$$y^d_{oa,2}(\tau_{oa}(q), \alpha) = [y_{cwf}(\tau_{oa}(q), \alpha^i_{oa})]_{i \in O_{oa}} \tag{4.32}$$

where each α^i_{oa} is a row of constants for a canonical walking function in (4.1). Similarly, the referrence trajectory for relative degree two outputs of the fully actuated and under actuated domains is the extended canonicaly function(the reason for switching to the extended canonical function for these domains will be clarified in Chapter 5). Defining indexing sets for full and under actuation, i.e. $v \in \{fa, ua\}$, to be O_{fa} and O_{ua}, respectively, the desired trajectories become,

$$\begin{aligned} y^d_{fa,2}(\tau_{fa}(q), \beta) &= [y_{ecwf}(\tau_{fa}(q), \beta^i_{fa})]_{i \in O_{fa}} \\ y^d_{ua,2}(\tau_{ua}(q), \beta) &= [y_{ecwf}(\tau_{ua}(q), \beta^i_{ua})]_{i \in O_{ua}} \end{aligned} \tag{4.33}$$

where β^i_v is a row of 7 parameters of an extended canonical walking function in (4.2).

4.4.4 Over/Fully Actuated Control

Using I/O Linearization [39], the goal is to drive a set of carefully chosen robot outputs to a set of human inspired referrence trajectories represented by the canonical human walking function. To do this, the final form of the outputs for over actuation and full actuation take the form:

$$\begin{aligned} y_{oa,1}(q, \dot{q}) &= y^a_{oa,1}(q, \dot{q}) - vhip_{oa} \\ y_{oa,2}(q) &= y^a_{oa,2}(q) - y^d_{oa,2}(\tau_{oa}(q), \alpha) \\ y_{fa,1}(q, \dot{q}) &= y^a_{fa,1}(q, \dot{q}) - vhip_{fa} \\ y_{fa,2}(q) &= y^a_{fa,2}(q) - y^d_{fa,2}(\tau_{fa}(q), \beta_{fa}) \end{aligned} \tag{4.34}$$

With both a mixture of relative degree one and relative degree two outputs for

both the over actuated and fully actuated domains, the control laws for $v \in \{oa, fa\}$ domain becomes,

$$u_v = -A_v^{-1}(q,\dot{q})\left(\begin{bmatrix} 0 \\ L_f L_f y_{v,2}(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} L_f y_{v,1}(q,\dot{q}) \\ 2\varepsilon L_f y_{v,2}(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} \varepsilon y_{v,1}(q,\dot{q}) \\ \varepsilon^2 y_{v,2}(q) \end{bmatrix} \right) \tag{4.35}$$

where the decoupling matrix for $v \in \{oa, fa\}$ is given in equation (4.12) and has full row rank by choice of a valid output combination. With this control law, the error between the robot output referrence trajectory is exponentially driven to zero at a rate of ε. It is important to note that, during over actuation, the system only has five degrees of freedom and thus can only independently track five outputs (adding more outputs will cause singularities in the decoupling matrix). This results in only five torques specified by I/O Linearization. Obviously since the robot has six actuators the robot cannot walk with one actuator outputting zero torque. This is the fundamental problem of over actuation and a method for dealing with over actuated control will be presented in a Chapter 6.

4.4.5 Under Actuated Control

During the under actuated phase, the robot no longer has full control authority over its forward velocity. This is a product of the fact that their is no actuator at the robots toe thus there is no way to independenly control the extended coordinate φ_0. It is for this reason that the forward velocity of the hip is not controlled during under actuation. With only relative degree two outputs during under actuation, the set of human inspired control outputs becomes,

$$y_{ua,2}(q) = y^a_{ua,2}(q) - y^d_{ua,2}(\tau_{ua}(q), \beta_{ua}). \tag{4.36}$$

For the case of only relative degree two outputs, the corresponding control law is shown in equation (4.13) to be,

$$u_{ua} = -A_{ua}^{-1}(q,\dot{q})\Big(L_f L_f y_{ua,2}(q,\dot{q}) + 2\varepsilon L_f y_{ua,2}(q,\dot{q}) + \varepsilon^2 y_{ua,2}(q,\dot{q})\Big) \tag{4.37}$$

where the decoupling matrix for under actuation is,

$$A_{ua}(q,\dot{q}) = L_g L_f y_{ua,2}(q,\dot{q}). \tag{4.38}$$

The next chapter discusses how a human inspired optimization is used to shape the referrence trajectories to produce planar, 3-domain multi-contact walking.

5. HUMAN INSPIRED OPTIMIZATION

This section presents a *human inspired optimization* for designing planar, 3-domain walking gaits. The output of this optimization is a set of controller parameters that yields a particular 3-domain gait which is shaped through the optimization using constraints. The objective function to the human inspired optimization is a least squares fit to corresponding outputs computed on discrete human locomotion data, reflecting the desire that the 3-domain walking be as humanlike as possible. From a subjects walking data, discrete times $t^H[k]$ and discrete values for the human output data, $y^H_{i,v}[k]$, for $i \in O_v$ where O_v are indexing sets as in Equations (4.32) and (4.33) for the outputs in each domain $v \in V$. For clarity, the final discrete point in each domain will be denoted as k_v. For example, the last discrete time for the fully actuated domain will be referred to as $t[k_{fa}]$. With this in mind, the objective function for the human inspired optimization is expressed mathematically as,

$$\begin{aligned} \text{Cost}_{\text{HD}}(\alpha, \beta_v) = &\sum_{i \in O_{oa}} \sum_{k=1}^{k_{oa}} (y_i^H[k] - y^d_{oa,i}(t^H[k], \alpha_i))^2 \\ &+ \sum_{j \in O_{fa}} \sum_{k=k_{oa}}^{k_{fa}} (y_j^H[k] - y^d_{fa,j}(t^H[k], \beta_{fa,j}))^2 \\ &+ \sum_{m \in O_{ua}} \sum_{k=k_{fa}}^{k_{ua}} (y_m^H[k] - y^d_{ua,m}(t^H[k], \beta_{ua,m}))^2 \end{aligned} \tag{5.1}$$

Minimizing this cost means creating trajectories as close to the human data as possible and thus creating human-like locomotion. Equally as important to the optimization are the equality and inequality constraints that, when satisfied, yield periodic, three domain walking. The notion of hybrid zero dynamics is key to ensuring periodicity [63] and will be explained in detail in the next section.

5.1 Hybrid Zero Dynamics

The goal of the controller in each domain is to drive the robot outputs to the corresponding human outputs, i.e. $y_v^a \to y_v^d$ or $y_v = y_v^a - y_v^d \to 0$ where y_v consists of either relative degree one or a mixture of relative degree one and two outputs as discussed in Chapter 4. This motivates the definition of the zero dynamics manifold for $v = ua$ to be defined by,

$$\mathcal{Z}_{ua} = \{(q, \dot{q}) \in TQ_R : y_{ua}(q, \dot{q}) = 0, L_f y_{ua}(q, \dot{q}) = 0\} \tag{5.2}$$

On this surface the control outputs y_{ua} and their derivatives $L_f y_{ua}$ are zero, and using the control laws in (4.35) they will remain zero for all time in the absence of disturbances(see Chapter 9 of [39] for a proof of this). Other works using the notion of zero dynamics in the context of bipedal locomotion include [3, 4]. For $v = oa$, a similar zero dynamics surface is defined; however, the relative degree one outputs are not included, thus it is termed a *partial* zero dynamics surface and is given by,

$$\mathcal{PZ}_{oa} = \{(q, \dot{q}) \in TQ_R : y_{oa,2}(q, \dot{q}) = 0, L_f y_{oa,2}(q, \dot{q}) = 0\} \tag{5.3}$$

For the fully actuated domain, the full zero dynamics surface, similar to under actuation except with the inclusion of relative degree one outputs in y_{fa}, is given by,

$$\mathcal{Z}_{fa} = \{(q, \dot{q}) \in TQ_R : y_{fa}(q, \dot{q}) = 0, L_f y_{fa}(q, \dot{q}) = 0\}. \tag{5.4}$$

These zero dynamics surfaces are invariant for the *continuous* dynamics, but are not invariant through perturbations such as impacts. The desire is to create surfaces that are invariant to impacts, thus creating surfaces that are *hybrid* zero dynamics

surfaces. With this in mind, the following optimization is defined in which the constraints are used to ensure the zero dynamics surfaces are hybrid invariant,

$$(\alpha^*, \beta_v^*, d_{fa}^*, d_{ua}^*) = \underset{\alpha, d_{fa}, d_{ua} \in \mathbb{R}^{27}}{argmin} \quad \text{Cost}_{\text{HD}}(\alpha, \beta_v) \tag{5.5}$$

$$s.t. \quad \Delta_{ua \to oa}(S_{ua} \cap \mathcal{Z}_{ua}) \subset \mathcal{PZ}_{oa} \tag{HZD1}$$

$$\Delta_{oa \to fa}(S_{oa} \cap \mathcal{PZ}_{oa}) \subset \mathcal{Z}_{fa} \tag{HZD2}$$

$$(S_{fa} \cap \mathcal{Z}_{fa}) \subset \mathcal{Z}_{ua} \tag{HZD3}$$

Notice that the constraint (HZD3) does not contain an impact map because the transition from full actuation to under actuation does not contain an impact. In the final form of this optimization, both constraints (HZD2) and (HZD3) will be satisfied by the use of a motion transition [29] which will be discussed in Section 5.4.

5.2 Inverse Kinematics

In order to obtain an initial condition on the under actuated zero dynamics surface, i.e. $(q, \dot{q})^T \in \mathcal{Z}_{ua}$, the outputs can be used along with the fact that on the edge between the under actuated and over actuated domains the height of the swing heel is zero. To construct this point, the impact invariance of the under actuated zero dynamics surface will be used. On $\mathcal{Z}_{ua}$ all relative degree two control outputs are zero by definition, i.e. $y_{ua} = y_{ua}^a - y_{ua}^d := 0$. Using this and the fact that the outputs are all linear and time is zero at the beginning of the over actuated domain, a particular configuration $\vartheta(\alpha)$ of the system at the beginning of the over actuated domain can be solved for in terms of the controller parameters,

$$\vartheta(\alpha) = q \quad s.t. \quad \begin{bmatrix} y_{ua}(\mathcal{R}_{ua \to oa} q) \\ h_{nsh}(q) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.6}$$

where $\mathcal{R}_{ua\to oa}$ is the relabeling matrix and $h_{nsh}(q)$ is the height of the nonstance heel above the walking surface. The reason for considering for considering y_{oa} at the point $\mathcal{R}_{ua\to oa}q$ is because this implies that the configuration of the robot at the beginning of the step is $q^+ = \mathcal{R}_{ua\to oa}q$ at which point $\tau_{oa}(\mathcal{R}_{ua\to oa}q^+) = 0$. This implies that $y_{oa}(\mathcal{R}_{ua\to oa}q^+) = H_{oa}\mathcal{R}_{ua\to oa}q^+ - y^d_{oa}(0)$. Using $\vartheta(\alpha)$ a point $(\vartheta(\alpha), \dot{\vartheta}(\alpha))^T \in (\mathcal{Z}_{ua} \cap S_{ua})$ can be found by first defining,

$$Y(q) = \begin{bmatrix} \frac{\partial \delta p_{hip}(q)}{\partial q} \\ \frac{\partial y_{ua}(q)}{\partial q} \end{bmatrix} \tag{5.7}$$

which yields,

$$\dot{\vartheta}(\alpha) = Y^{-1}(\vartheta(\alpha)) \begin{bmatrix} vhip_{oa} \\ 0 \end{bmatrix} \tag{5.8}$$

where $Y(\vartheta(\alpha))$ is guaranteed to be invertible for a valid output combination.

5.3 State Reconstruction

This section presents the key benefits to the zero dynamics approach to locomotion by leveraging the form of the outputs, specifically the fact that all of the robot outputs are linear. Using the zero dynamics, a low dimensional representation of the full order biped can be found [63, 1, 4] that can then be used to reconstruct the full order dynamics. Consider the following coordinates to represent the zero dynamics for the over actuated and fully actuated domains,

$$\xi_{v,1} = \delta p_{hip}(q) = \rho_v q \tag{5.9}$$

$$\xi_{v,2} = \rho_v \dot{q} \tag{5.10}$$

where $\delta p_{hip,v}(q)$ is simply the linearized forward posititon of the hip which is used to parameterize time in (4.30). With this the desired outputs become $y^d_{v,2}(\tau_v(q)) = y^d_{v,2}(\xi_1)$. The linear form of the outputs makes it possible to reconstruct the full state from $\xi_{v,1}$ and $\xi_{v,2}$. First recall the linear form of the outputs allows them to be written in the simple form given in Equation (4.27). With this, the full state of the system can be *reconstructed* to yield the reconstructed positions and velocities, denoted $(q^r, \dot{q}^r)$,

$$q^r_v = \begin{bmatrix} \rho_v \\ H_v \end{bmatrix}^{-1} \begin{bmatrix} \xi_{v,1} \\ y^d_{v,2} \end{bmatrix} \tag{5.11}$$

$$\dot{q}^r_v = \begin{bmatrix} \rho_v \\ H_v \end{bmatrix}^{-1} \begin{bmatrix} \xi_{v,2} \\ \frac{\partial y^d_{v,2}}{\partial \xi_{v,1}} \xi_{v,2} \end{bmatrix} \tag{5.12}$$

It is important to recall that, for the over actuated domain, the zero dynamics surface only has a dimension of five while the system has seven angles(see Figure 3.2). Due to this the reconstruction for q_{oa} and $\dot{q}_{oa}$ will only contain the first 10 joint angles and velocities of the system and the remaining four positions and velocities are found using inverse kinematics. The following example will help to clarify this,

Example 5 *A given state reconstruction for $v = oa$ will only contain $(q^r_{oa,1:5}, \dot{q}^r_{oa,1:5})^T$ or $(\varphi_0, q_{sa}, q_{sk}, q_{sh}, q_{nsh}, \dot{\varphi}_0, \dot{q}_{sa}, \dot{q}_{sk}, \dot{q}_{sh}, \dot{q}_{nsh})^T$. The remaining four positions and velocities can be solved for using inverse kinematics and the fact that during double support the distance between the contact points is known and constant. With this,*

the remaining elements in the state are given by,

$$q^r_{oa,6} = \Psi_1(q^r_{oa,1:5}) \tag{5.13}$$

$$q^r_{oa,7} = \Psi_2(q^r_{oa,1:6}) \tag{5.14}$$

$$\dot{q}^r_{oa,6} = \dot{\Psi}_1(q^r_{oa,1:5}, \dot{q}^r_{oa,1:5}) \tag{5.15}$$

$$\dot{q}^r_{oa,7} = \dot{\Psi}_2(q^r_{oa,1:6}, \dot{q}^r_{oa,1:6}) \tag{5.16}$$

Fortunately, the fully reconstruction during the fully actuated domain is able to reconstruct the full system; therefore, no additional inverse kinematics are needed to obtain the full state.

For the over and fully actuated domains, the control law in (4.35) fully linearizes the dynamics, resulting in the relative degree one outputs evolving according to $\dot{y}_{v,1} = -\varepsilon y_{v,1}$. This results in the zero dynamics surface evolving according to the *linear* ODE,

$$\dot{\xi}_{v,1} = \xi_{v,2} \tag{5.17}$$

$$\dot{\xi}_{v,2} = -\varepsilon(\xi_{v,2} - vhip_v), \quad v \in \{oa, fa\} \tag{5.18}$$

The advantage of representing the zero dynamics in this way is that the full state of the system can be reconstructed with the position and velocity of the hip, thus not needing to numerically integrate the system forward in time. For the under actuated domain, however, this reconstruction is not valid because there are no relative degree one outputs; therefore, the full order dynamics were integrating forward in time using Matlabs ODE45 function. This can be done since the initial condition for $v = ua$ can be obtained via a reconstruction on S_{fa} and then integrated forward until the swing heel strikes.

5.4 Motion Transitions

At the end of the over actuated phase, the stance toe strikes the ground, thus imparting another impulsive force to the robot. There is also a change in the degrees of actuation as the robot transitions from over actuation to full actuation. Due to the change in degrees of actuation and this impact, constraining the partial zero dynamics surface to be impact invariant through toe strike is not possible because the robot can track six outputs during full actuation compared to five outputs during over actuation. It is for this reason that the notion a *motion transition* [29] will be introduced. First consider the *extended canonical walking function* in (4.2) which is used for the desired trajectories for the relative degree two outputs in both the fully and under actuated phases, i.e. $v \in \{fa, ua\}$,

$$y^d_{v,2} = y_{ecwf}(\beta^i_v) = [e^{-\beta^i_{v,4}t}(\beta^i_{v,1}cos(\beta^i_{v,2}t) + \beta^i_{v,3}sin(\beta^i_{v,2}t)) \tag{5.19}$$
$$+ \beta^i_{v,5}cos(\beta^i_{v,6}t) + \frac{2\beta^i_{v,4}\beta^i_{v,5}\beta^i_{v,6}}{(\beta^i_{v,4})^2 + (\beta^i_{v,2})^2 - (\beta^i_{v,6})^2}sin(\beta^i_{v,6}t) + \beta^i_{v,7}]_{i\in O_v}$$

where O_v is an indexing set of the outputs for each $v \in \{fa, ua\}$. This function, being the solution to a linear mass spring damper system subject to sinusoidal forcing, is very similar to the canonical walking function in (4.1) with the additional parameters. The goal is to use these extra parameters along with the partial hybrid zero dynamics surface from the over actuated domain and to solve for the parameters of (5.19) that will *connect* the surfaces together. Figure 5.1 shows the closed loop geometry of the connected zero dynamics surfaces generated by motion transitions. In order to do this, first consider the state of the system at the beginning and end of the over actuated phase, denoted x^0_{oa} and x^f_{oa}, respectively, where it is important to note that x^f_{oa} is immediately after the toe strikes. Using these states, the positions and

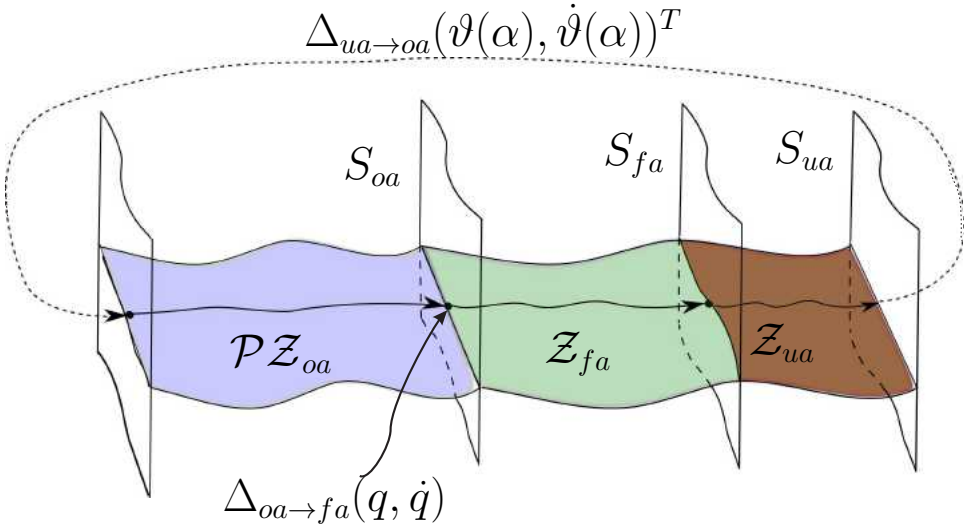

Figure 5.1: The geometry of the close loop that results from connecting the three zero dynamics surfaces.

velocities of any output combination can be computed. In this case the interest lies in connecting all of the outputs used in both the fully actuated and under actuated domains to the over actuated domain. To do so, first notice the linear dependence of the extended canonical walking function on the terms $\beta^i_{,1}, \beta^i_{,3}, \beta^i_{,5}$, and $\beta^i_{,7}$. This linearity along with the parameterization of time, $t = \frac{\xi_{v,1}-d_v}{v_{hip,v}}$, allows (5.19) to be written as,

$$y_{ecwf}(\beta^i_v, \xi_{v,1}, d_v, v_{hip,v}) = Y_{ecwf}(\beta^i_{v,2}, \beta^i_{v,4}, \beta^i_{v,6}, \xi_{v,1}, d_v, v_{hip,v}) \begin{bmatrix} \beta^i_{v,1} \\ \beta^i_{v,3} \\ \beta^i_{v,5} \\ \beta^i_{v,7} \end{bmatrix} \tag{5.20}$$

where,

$$Y_{ecwf}(\beta^i_{v,2}, \beta^i_{v,4}, \beta^i_{v,6}, \xi_{v,1}, d_v, v_{hip,v}) = \tag{5.21}$$
$$\begin{bmatrix} e^{-\beta^i_{v,4}\tau_v(\xi_{v,1}, v_{hip,v}, d_v)} cos(\beta^i_{v,2}\tau_v(\xi_{v,1}, v_{hip,v}, d_v)) \\ e^{-\beta^i_{v,4}\tau_v(\xi_{v,1}, v_{hip,v}, d_v)} sin(\beta^i_{v,2}\tau_v(\xi_{v,1}, v_{hip,v}, d_v)) \\ cos(\beta^i_{v,6}\tau_v(\xi_{v,1}, v_{hip,v}, d_v)) + \frac{2\beta^i_{v,4}\beta^i_{v,6}}{(\beta^i_{v,4})^2+(\beta^i_{v,2})^2-(\beta^i_{v,6})^2} sin(\beta^i_{v,6}\tau_v(\xi_{v,1}, v_{hip,v}, d_v)) \\ 1 \end{bmatrix}^T$$

Recall from the previous section that $\xi_{v,1}$ is the forward position of the hip, and thus can be directly substituted into Equation (4.30) for the parameterization of time. Next the positions and velocities of all relative degree two outputs are computed at the two points being connected yielding,

$$y^0_i = y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_v) \tag{5.22}$$
$$\dot{y}^0_i = \frac{d}{d\xi_{v,1}} y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_v))\Big|_{\xi_{v,1}=\xi^0_{v,1}} \tag{5.23}$$
$$y^f_i = y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_v) \tag{5.24}$$
$$\dot{y}^f_i = \frac{d}{d\xi_{v,1}} y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_v))\Big|_{\xi_{v,1}=\xi^f_{v,1}} \tag{5.25}$$

for $i \in Outputs_v$. Using (5.20), the following matrix is defined:

$$\mathbb{Y} = \begin{bmatrix} Y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_{v,2}, \beta^i_{v,4}, \beta^i_{v,6}) \\ \frac{d}{d\xi_{v,1}} Y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_{v,2}, \beta^i_{v,4}, \beta^i_{v,6})\Big|_{\xi_{v,1}=\xi^0_{v,1}} \\ Y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_{v,2}, \beta^i_{v,4}, \beta^i_{v,6}) \\ \frac{d}{d\xi_{v,1}} Y_{ecwf}(\tau_v(\xi_{v,1}, v_{hip,v}, d_v), \beta^i_{v,2}, \beta^i_{v,4}, \beta^i_{v,6})\Big|_{\xi_{v,1}=\xi^f_{v,1}} \end{bmatrix} \tag{5.26}$$

By selecting $\beta^i_{v,2}$, $\beta^i_{v,4}$, and $\beta^i_{v,6}$ such that $(\beta^i_{v,2})^2+(\beta^i_{v,4})^2-(\beta^i_{v,6})^2 \neq 0$, the parameters $\beta^i_{v,1}$, $\beta^i_{v,3}$, $\beta^i_{v,5}$, and $\beta^i_{v,7}$ can be solved for in close form that will produce a surface that connects to $\mathcal{PZ}_{oa}$ on both edges,

$$\begin{bmatrix} \beta^i_{v,1} \\ \beta^i_{v,3} \\ \beta^i_{v,5} \\ \beta^i_{v,7} \end{bmatrix} = \mathbb{Y}^{-1} \begin{bmatrix} y^0_i \\ \dot{y}^0_i \\ y^f_i \\ \dot{y}^f_i \end{bmatrix} \tag{5.27}$$

Performing a motion transition for each output results in a matrix with each row containing the parameters to an extended canonical function pertaining to each relative degree two output. Furthermore, the desired hip velocity for the fully actuated phase is simply chosen to be the velocity of the hip after toe strike. Defining $(q^+_{oa}, \dot{q}^+_{oa})^T = \Delta_{oa\to fa}(x^f_{oa})$, the desired hip velocity for the fully actuated domain becomes,

$$vhip_{fa} = \frac{\partial \delta p_{hip}(q^+_{oa})}{\partial q^+_{oa}} \dot{q}^+_{oa} \tag{5.28}$$

Since toe strike will cause the velocity of the hip to undergo a discrete jump, this simply makes the desired hip velocity equal to the actual hip velocity after toe strike. Using these motion transitions, controller parameters can be solved for in closed form that will connect a source domain to a target domain while maintaining on the zero dynamics surface.

5.4.1 Over Actuation Constraints

In order to ensure that the resulting walking is physically realizable, constraints must be placed on the contact forces. To compute the contact forces, standard meth-

ods described in [16] are used to obtain the contacting forces and moments for the over actuated phase. During over actuation, both the stance heel and nonstance toe are in contact with the walking surface. Recall that a domain contains a set of contact points c_v as well as a jacobian, $J_v(q)$, that is then used to complete the constrained dynamical model in (3.7). It follows from this that the contact forces and moments can then be computed using equation (3.9)(refer to Example 3 for an example showing how to compute the jacobian). From this, the following vector of forces and moments are obtained: $F_{oa} = \{F^x_{sh,oa}, F^z_{sh,oa}, \mathcal{M}^y_{sh,oa}, F^x_{nst,oa}, F^z_{nst,oa}, \mathcal{M}^y_{nst,oa}\}^T$. Since the robot is rotating about the toe and heel, constraints on the direction of the reaction forces are needed to ensure the robot can not pull up on the ground,

$$F^z_{sh,oa} \geq 0 \tag{C1}$$

$$F^z_{nst,oa} \geq 0. \tag{C2}$$

The domain following over actuation is the fully actuated domain in which the robots stance foot is flat while the nonstance leg is swinging. In order to ensure that at the end over actuation the robot transitions into the fully actuated domain, the height of the stance toe must be zero and the force with which the nonstance toe is pushing into the ground must also be zero. Considering a reconstructed state $x^r = (q^r, \dot{q}^r)^T \in (S_{oa} \cap D_{oa})$, the height of the stance toe and the vertical reaction force at the nonstance toe at that point, denoted $h^e_{st,oa}$ and $F^{z,e}_{nst,oa}$, must be zero,

$$F^{z,e}_{nst,oa} = 0 \tag{C3}$$

$$h^e_{st,oa} = 0. \tag{C4}$$

5.4.2 Full Actuation Constraints

During the fully actuated domain the robots stance foot is flat on the ground and the nonstance leg is swinging. As with over actuation, constraints are needed to ensure the physical realizability of the walking. Since the stance foot is flat during full actuation, the reaction forces at the stance heel and stance toe must be pushing into the ground,

$$F^z_{st,oa} \geq 0 \tag{C5}$$

$$F^z_{sh,oa} \geq 0. \tag{C6}$$

This also prevents the robot from rolling forward or backward on either edge of the foot, which is synonomous with making sure the center of pressure(COP) [38] remains within the edges of the stance foot. With the domain following full actuation being under actuation in which the stance heel is lifting, a constraint must be imposed so that, on the edge of the two domains, the vertical reaction force on the stance heel is zero. As was done with under actuation, consider a state obtained from the reconstruction $x^r = (q^r, \dot{q}^r)^T \in (S_{fa} \cap D_{fa})$, the vertical reaction force at the stance heel, denoted $F^{z,e}_{sh,fa}$, must be zero,

$$F^{z,e}_{sh,fa} = 0. \tag{C7}$$

5.4.3 Under Actuation Constraints

During under actuation the only point on the robot in contact with the ground is the stance toe. It is thus necessary to constrain the vertical reaction force at the

toe, denoted $F^z_{st,ua}$, to be positive,

$$F^z_{st,ua} \geq 0. \tag{C8}$$

As was done with over actuation and full actuation, this constraint is needed to ensure that the robot does not attempt to pull against the ground.

5.5 Optimization Formulation

With the critical optimization constraints outlined, the final form of the optimization becomes,

$$\begin{aligned}(\alpha^*, \beta_v^*, d_{fa}^*, d_{ua}^*) \quad &= \quad \underset{\alpha, d_{fa}, d_{ua} \in \mathbb{R}^{27}}{argmin} \quad \text{Cost}_{\text{HD}}(\alpha, \beta_v) \\ &\qquad s.t. \quad (\text{C1}, \text{C2}, ..., \text{C8}, \text{HZD1})\end{aligned} \tag{5.29}$$

Upon completion, this optimization yields periodic, 3-domain planar bipedal walking that is physically realizable and thus experimentally implementable. The optimization is solved using Matlab's *FMINCON* function, which solves nonlinear, constrained optimization problems. Using a standard desktop computer/laptop equipped with a quadcore processor, the optimization generally finishes in 0.5-1.5 hours if using Matlab's parallel computing toolbox allowing the optimization to utilize all four processor cores. The overall flow of the human inspired optimization is summarized in a concise itemized list as follows:

1) By fitting the canonical function to human walking data, provide initial guesses for controller parameters, α, and the domain transition points, d_{fa}, and d_{ua}.

2) Solve the inverse kinematics problem, yielding $(\vartheta(\alpha), \dot{\vartheta}(\alpha))$– the initial condition to the double support domain on the partial hybrid zero dynamics surface,

$\mathcal{PZ}_{oa}$.

3) Compute the state reconstruction for the double support domain and use this reconstruction to compute constraints on the domain of admissibility D_{oa} for double support.

4) Compute the reconstruction of the system on the guard from double support to full actuation $S_{oa\to fa}$ and apply the resetmap $\Delta_{oa\to fa}$ to obtain the state after toe strike.

5) Compute parameters for the full actuation domain, β_{fa}, that construct a motion transition (5.27) from the state of the system immediately after toe strike to the initial condition $(\vartheta(\alpha), \dot{\vartheta}(\alpha))$.

6) Compute the state reconstruction for the fully actuated domain and use this reconstruction to compute inequality constraints on the domain of admissibility, D_{fa}, for full acutation.

7) Compute the state reconstruction on the guard from full actuation to under actuation, $S_{fa\to ua}$ and apply the resetmap $\Delta_{oa\to fa}$.

8) Compute parameters for the full actuation domain, β_{ua}, that construct a motion transition from the state of the system immediately after heel lift to the initial condition $(\vartheta(\alpha), \dot{\vartheta}(\alpha))$.

9) Integrate through the underactuated domain to compute inequality constraints on the domain of admissibility D_{ua} for under actuation.

10) Evaluate $\text{Cost}_{\text{HD}}(\alpha, \beta_v)$. If local minimum has been reached and constraints are satisfied, exit; otherwise, use gradient based method to initialize new values for α, d_{fa}, and d_{ua} and return to 2).

6. QUADRATIC PROGRAMS

One of the more difficult aspects of designing controllers for humanlike locomotion is attempting to tie together domains of differing degrees of actuation. Systems that dynamically change the level of actuation are not often encountered in controls; however, it is a problem that is inherent to bipedal locomotion. The key difficulty in controlling such systems arises in applying control during over actuated domains. As stated in Section 4.4.4, the control law for over actuation in Equation (4.35) does not allow torques to be specified for each joint; because, during over actuation, the degrees of actuation are less than the number of actuators on the robot, and attempting to track more outputs during over actuation would result in singularities in the decoupling matrix.

The reason the decoupling matrix goes singular during over actuation is that, with more actuators than degrees of freedom, there are essentially an infinite number of ways to apply actuation to meet the control objective. In situations in which there are many solutions such as with the human inspired optimization in (5.29), the simplest way to find a viable solution is to pose an optimization. In this case, the optimization needs to be able to solve fast enough to be ran in real time on a robot. This can be done using a quadratic program(QP) [2] which can be solved extremely quickly and have been shown to yield success experimentally [23, 53]. One of the more recent methods described in [2] involves the use of Lyapunov functions to define the convergence characteristics. The key benefits to this Lyapunov based formulation is the way that it allows for tasks to be prioritized, meaning that the convergence of some outputs can be allowed to converge at slower rates if needed, allowing the QP more room to satisfy constraints. For the purposes of this thesis, a somewhat more

simple quadratic program formulation will be used.

Since torque saturation is a common issue in bipedal robots, the desire is to pose a quadratic program that will meet the control objective while being optimal with respect to the amount of actuator torque requested. First recall that for $v \in \{oa, fa\}$, the control law for these domains is given in Equation (4.37). This can be rewritten in the following form,

$$-A_v(q,\dot{q})u_v = \left(\begin{bmatrix} 0 \\ L_f L_f y_{v,2}(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} L_f y_{v,1}(q,\dot{q}) \\ 2\varepsilon L_f y_{v,2}(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} \varepsilon y_{v,1}(q,\dot{q}) \\ \varepsilon^2 y_{v,2}(q) \end{bmatrix} \right) \quad \text{(QPC1)}$$

in which it can be posed as a constraint to a quadratic program, resulting in the following control law for $v \in \{oa, fa\}$,

$$\begin{aligned} u_v^* = \underset{u_v \in \mathbb{R}^9}{argmin} \quad & u_v^T W_v u_v \\ s.t. \quad & \text{(QPC1)} \end{aligned} \tag{6.1}$$

where W_v is a 9×9 torque distribution matrix that can be used to distribute the torques to allow certain actuators to do more of the work than others if some joints have more powerful actuators than others which is commonly the case. For the robot model being used here, the torque distribution matrix takes the form,

$$W_v = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{6\times 3} & (\omega_v)_{m_r \times m_r} \end{bmatrix} \tag{6.2}$$

where m_r is the number of actuators on the robot and ω_v is a $m_r \times m_r$ diagonal matrix in which the diagonal elements reflect the desired torque distribution. If each actuator is to be weighted equally, ω_v is the identity matrix. If there are no relative

degree one outputs as with the under actuated domain, the IO control law in (4.37) is rewritten as,

$$-A_{ua}(q,\dot{q})u_v = \Big(L_f L_f y_{ua,2}(q,\dot{q}) + 2\varepsilon L_f y_{ua,2}(q,\dot{q}) + \varepsilon^2 y_{ua,2}(q,\dot{q})\Big). \qquad \text{(QPC2)}$$

With this, the resulting control law for the under actuated domain becomes,

$$u^*_{ua} = \underset{u_v \in \mathbb{R}^9}{argmin} \;\; u_{ua}^T W_{ua} u_{ua} \tag{6.3}$$

$$s.t. \quad \text{(QPC2)}$$

These quadratic programs not only allow for the distribution of torque, they also do not require the inversion of the decoupling matrix, meaning that during over actuation the quadratic program can specify torques for each actuator that minimizes torque while meeting the convergence objectives. The following section will present simulation results from a 3-domain gait obtained using the human inspired optimization.

7. SIMULATION RESULTS

This chapter presents a simulation example showing how the optimization algorithm of Chapter 5 was used to produce a multi-domain walking gait for the planar biped Amber 2. Amber 2 is a planar bipedal robotic testbed that was designed and partially machined within AMBER lab at Texas A&M University. Amber 2 is a seven link robot supported by a light weight, carbon fiber boom that restricts motion to the saggital plane. The boom is counter weighted so as to not introduce any mass to the robot; however, there is an inertial load introduced to the torso link that is negligable due to the low friction bearings used in the construction of the boom as well as a long moment arm (approx. 8ft) separating the robot from the center of rotation of the boom. The mass and length parameters of Amber 2 are shown in Table 7.1 with variable conventions as defined in Figure 7.1. The optimization algorithm of Chapter 5 was implemented using MATLAB'S FMINCON command. In the underactuated phase, the integration algorithm ode45 was used to integrate the dynamics forward in time. In both the double support and fully actuated phases, the solution at each timestep was found in closed-form using the reconstruction algorithm discussed in Section 5.3. The walking gait that results

Table 7.1: AMBER 2.0 Mass & Length Parameters

Link	Mass(g)	Length(mm)	Width(mm)
Foot	204.42	177.8	47.63
Calf	1119.43	343.13	50.8
Thigh	1172.57	298.45	50.8
Torso	2154.79	104.01	285.75

from the optimization is plotted as a series of tiles in Figure 7.2. An animation is available at [`http://www.youtube.com/watch?v=OY-QsaIglQY`]. The gait shown

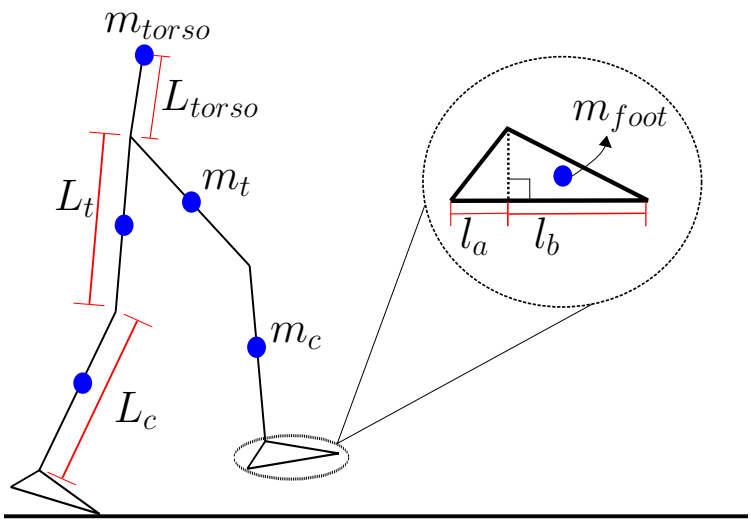

Figure 7.1: Notation and and layout of the robot model parameters.

here has an average velocity of 0.42 m/s, with a step length of 0.373 m (58 pct of leg length) and a period of $T = 0.88$ sec. The step time can be further broken down, with 0.139 s (16 pct) in double support / over actuation, 0.19 s (21 pct) in full actuation, and 0.56 s (63 pct) in under actuation. The initial condition for the gait on the guard of the under actuated domain,i.e. $x_0 = (q_0, \dot{q}_0)^T \in S_{ua \to oa}$, expressed in the form $(q_0, \dot{q}_0)^T = \{p_x, p_y, \varphi_0, q_r, \dot{p}_x, \dot{p}_y, \dot{\varphi}_0, \dot{q}_r\}^T$, is

$$q_0 = (0, 0, -0.162, -0.298, 0.303, 0.073, -0.546, 0.469, -0.366)^T \tag{7.1}$$

$$\dot{q}_0 = (0, 0, -0.628, -0.069, 0.036, 0.661, -0.262, -4.985, -2.246)^T \tag{7.2}$$

The actual and desired outputs for the gait are shown in Figure 7.3. Recall that the hip velocity is not invariant through impact and thus the actual hip velocity is not equal to the desired hip velocity at the beginning of the step. Also note the discrete jump in the velocity of the hip is due to the impact at toe strike. Shown in Figure 7.4 on the left is a phase plot of the 3-domain walking over 9 steps displaying the

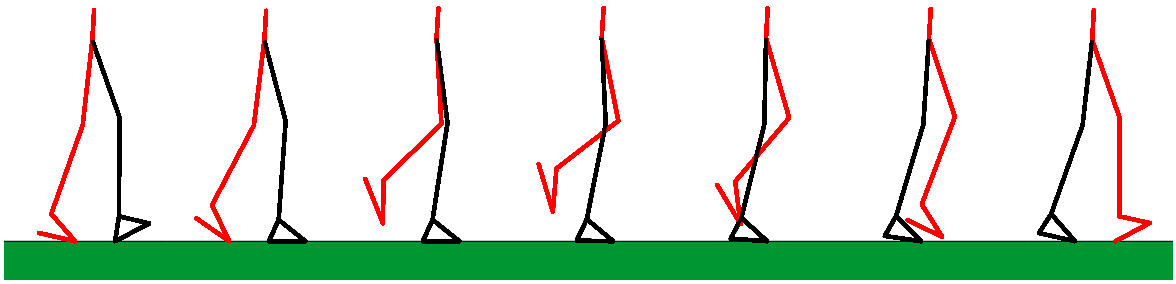

Figure 7.2: Gait tiles for 3 domain walking gait.

periodic nature of the walking. Also in Figure 7.4 are plots of the joint angles and joint velocities in the top right and bottom right, respectively.

As stated in Chapter 5, I/O Linearization is used to design the multi-domain walking gaits. Recall the limitations with I/O Linearization and over actuation discussed in Section 4.4.4. With the over actuated domain here consisting of 5 degrees of freedom, a torque cannot be specified for the nonstance ankle. This can be seen in Figure 7.5 on the left in which the joint torques are shown over the course of a single step using simple I/O Linearization. The figure on the right is also the torques over a single step using the quadratic program from Chapter 6 that is able to specify a torque at all six actuators to track the outputs. It is also important to note that the torque across domain transitions is not continuous. For experimental implementation, however, low pass filters would be applied to the commanded torques which would smooth out the discontinuities.

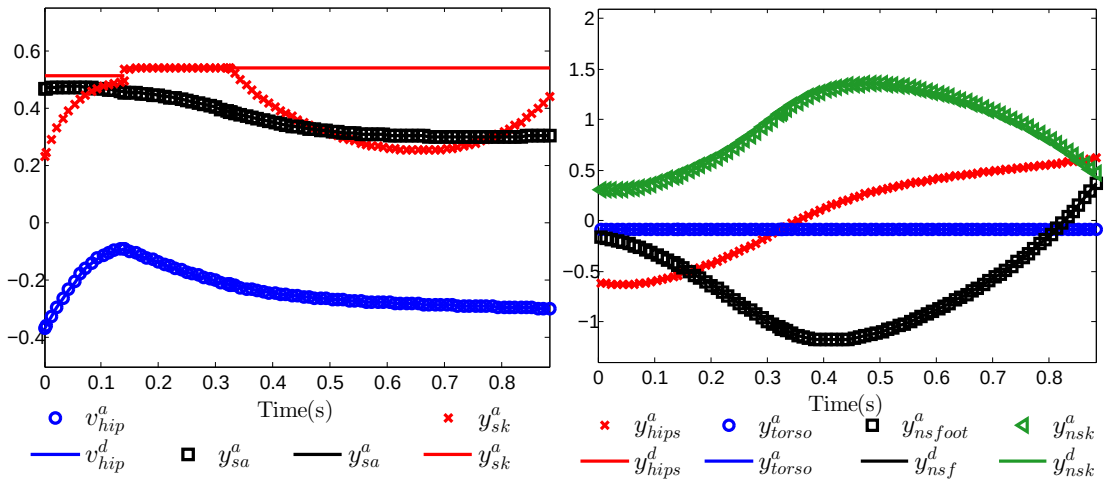

Figure 7.3: Actual and desired outputs over one step for a 3 domain walking gait.

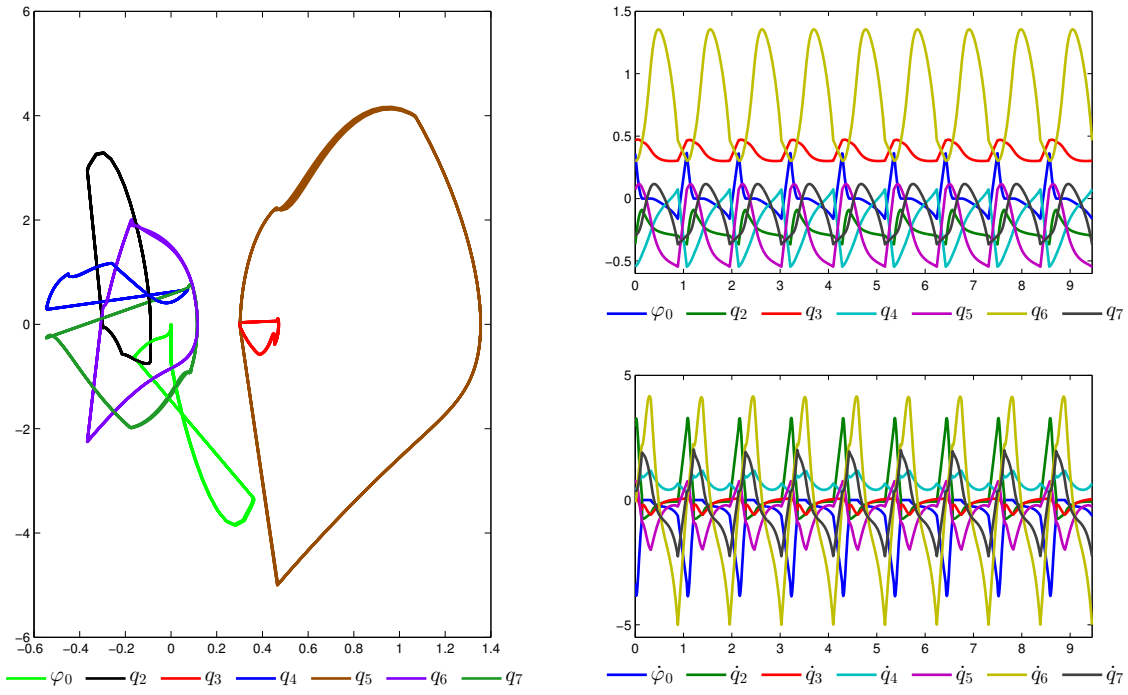

Figure 7.4: On the left is a phase plot for 9 steps of a 3 domain walking gait which shows the periodicity of the gait. The joint angles and velocities are shown in the plots in the top right and bottom right, respectively.

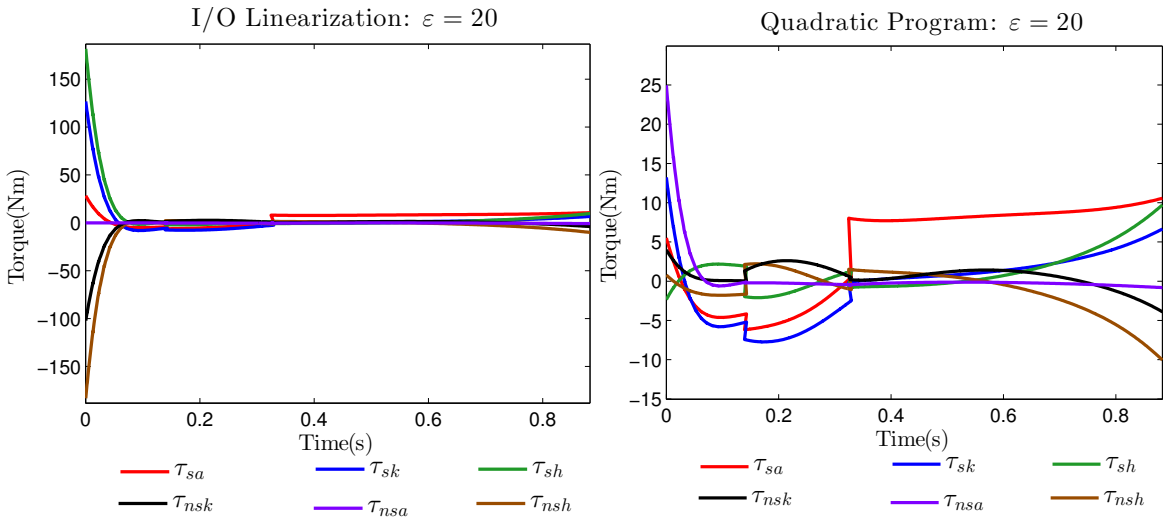

Figure 7.5: Joint torques over one step using I/O Linearization(left) and the quadratic program(right).

8. CONCLUSION

The objective of this thesis has been to present one of the first formal methods with which to design multi-domain walking gaits. It began with the introduction of multi-domain hybrid systems followed by the introduction of the floating base model. The principle contribution lies, however, in the multi-domain human inspired optimization. Using the human inspired optimization, walking gaits and other behaviors can be easily designed and shaped by simply altering constraints while ensuring the gaits remain physically realizable. In addition to the constraints, alterations to the cost could be made that could result in gaits that are optimal with respect to energy or a number of other important metrics. The human-like walking presented here was achieved using Input/Output Linearization, but was augmented with an online quadratic program to introduce a sense of optimality to the applied control, allowing for a single control law to be used for domains with differing degrees of actuation. Online quadratic programs are becoming increasingly attractive in controls with the dramatic increase in computational speed. These online optimizations have endless potential in the field of robot control and could help streamline the transition from simulation to hardware.

Future work is needed to improve the optimization described here so that a single set of outputs can be used rather than connecting domains and switching outputs as well as extending the optimization to 3-dimensions. Finally, it is important to start researching reactive walking behaviors in which any behavior design is done via online optimizations possibly with a quadratic program. This kind of real time planning would allow for the variation of the walking surface as well as the ability to start, stop, and change directions effortlessly. For this to become realizable, researchers

need to better understand how to include the external forcing into the design of the controllers since interaction with the environment plays such a substancial role in locomotion.

REFERENCES

[1] Ames, A. First steps toward underactuated human-inspired bipedal robotic walking. In *IEEE International Conference on Robotics and Automation* (St. Paul, MN), pp. 1011–1017.

[2] Ames, A., and Powell, M. *Towards the Unification of Locomotion and Manipulation through Control Lyapunov Functions and Quadratic Programs.*, vol. 449 of *Lecture Notes in Control and Information Sciences*. Springer, New York, NY, 2012.

[3] Ames, A. D. *First Steps Toward Automatically Generating Bipedal Robotic Walking from Human Data*, vol. 422 of *LNICS*. Springer, New York, NY, 2012.

[4] Ames, A. D., Cousineau, E. A., and Powell, M. J. Dynamically stable robotic walking with NAO via human-inspired hybrid zero dynamics. In *Hybrid Systems: Computation and Control* (Ny, New York, 2012), pp. 135–144.

[5] Ames, A. D., Vasudevan, R., and Bajcsy, R. Human-data based cost of bipedal robotic walking. In *IEEE Conference on Hybrid Systems: Computation and Control* (Chicago, IL, 2011), pp. 153–162.

[6] Au, S. K., Dilworth, P., and Herr, H. An ankle-foot emulation system for the study of human walking biomechanics. In *IEEE Intl. Conf. Robotics and Automation* (Orlando, May 2006), pp. 2939–2945.

[7] C. T. Farley, H. H. P. Houdijk, C. V. S., and Louie, M. Mechanism of leg stiffness adjustment for hopping on surfaces of different stiffnesses. *Journal of Applied Physiology 85* (1998), 1044–1055.

[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] Chevallereau, C., G. Abba, Y. A., Plestan, F., Westervelt, E. R., de Wit, C. C., and Grizzle, J. W. Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine 23*, 5 (June 2003), 57–79.

[10] Collins, S., Ruina, A., Tedrake, R., and Wisse, M. Efficient bipedal robots based on passive-dynamic walkers. *Science 307*, 5712 (Feb 2005), 1082–1085.

[11] Cousineau, E., and Powell, M. Small humanoid robot nao walking with human-like gait obtained via human-inspired optimization. `http://www.youtube.com/watch?v=OBGHU-e1kc0/`, Oct. 16, 2011.

[12] Espiau, B., and Goswami, A. Compass gait revisited. In *IFAC Symposium on Robot Control* (Capri, Sept. 1994), pp. 839–846.

[13] Featherstone, R. *Rigid Body Dynamics Algorithms*. Springer, New York, NY, 2008.

[14] Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback Control of Dynamic Systems 6th Edition*. Prentice Hall, Upper Saddle River, NJ, 2009.

[15] Goswami, A., Thuilot, B., and Espiau, B. A study of the passive gait of a compass-like biped robot: Symmetry and chaos. *IJRR 17*, 12 (Dec. 1998), 1282–1301.

[16] Grizzle, J. W., Chevallereau, C., Ames, A. D., and Sinnet, R. W. 3D bipedal robotic walking: models, feedback control, and open problems. In *IFAC Symposium on Nonlinear Control Systems* (Bologna, Italy, 2010).

[17] 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* (2008), pp. 5206–5213.

[18] Hamill, J., and Knutzen, K. M. *Biomechanical Basis of Human Movement*. Lippincott Williams & Wilkins, Philadelphia, Feb. 2003.

[19] Heller, M. O., Bergmann, G., Deuretzbacher, G., Dürselen, L., Pohl, M., Claes, L., Haas, N. P., and Duda, G. N. Musculo-skeletal loading conditions at the hip during walking and stair climbing. *Journal of Biomechanics 34*, 1 (July 2001), 883–893.

[20] Hurst, J., and Grizzle, J. Atrias walking with an efficient, human-like walking gait. `http://www.youtube.com/watch?v=uswiqPbFLZ0`, Sep. 22, 2012.

[21] Kagami, S., Nishiwaki, K., Jr, J. J. K., Kuniyoshi, Y., Inaba, M., and Inoue, H. Online 3d vision, motion planning and bipedal locomotion control coupling system of humanoid robot: H7. In *IEEE Conference on Intelligent Vision Robots and Systems.* (Lausanne Switzerland, 2002), vol. 3, pp. 2557–2562.

[22] 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 Intl. Conf. on Intelligent Robots and Systems* (Beijing, P.R. China, 2006), pp. 2993–2999.

[23] Kolavennu, S., Palanki, S., and Cockburn, J. C. Nonlinear control of nonsquare multivariable systems. *Chemical Engineering Science 56* (2001), 2103–2110.

[24] Koolen, T., Boer, T. D., Rebula, J., Goswami, A., and J.Pratt. Capturability-based analysis and control of legged locomotion. part 1: Theory and application to three simple gait models. In *International Journal of Robotics Research, 2012* (2012), vol. 31, pp. 1094–1113.

[25] Lee, S.-H., and Goswami, A. Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground. In *International Conference on Intelligent Robots and Systems* (Taipei, Taiwan, 2010), pp. 3157–3162.

[26] McGeer, T. Passive dynamic walking. *Intl. J. of Robotics Research 9*, 2 (Apr. 1990), 62–82.

[27] McMahon, T., and Cheng, G. The mechanics of running: How does stiffness couple with speed? *Journal of Biomechanics 23* (1990), 65–78.

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

[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 *International Conference on Robotics and Automation* (Karlsruhe, Germany, 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 *IEEE Intl. Conf. Robotics and Automation* (Minnesota, 2012), pp. 543–549.

[31] Pratt, J., Carff, J., and Drakunov, S. Capture point: A step toward humanoid push recovery. In *in 6th IEEE-RAS International Conference on Humanoid Robots* (Genoa, Italy, 2006), pp. 200–207.

[32] Pratt, J., Koolen, T., Boer, T. D., Rebula, J., Cotton, S., Carff, J., Johnson, M., and Neuhaus, P. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. In *International Journal of Robotics Research* (2012), vol. 31, pp. 1117–1133.

[33] Rengifo, C., Aoustin, Y., Chevallereau, C., and Plestan, F. A penalty-based approach for contact forces computation in bipedal robots. In *9th IEEE International Conference on Humanoid Robots* (Paris, Dec 2009), pp. 121–127.

[34] Riener, R., Rabuffetti, M., and Frigo, C. Stair ascent and descent at different inclinations. *Gait and Posture 15* (2002), 32–44.

[35] Rodgers, M. M. Dynamic biomechanics of the normal foot and ankle during walking and running. *Physical Therapy 68*, 12 (Dec. 1988), 1822–1830.

[36] S. Nadubettu Yadukumar, M. P., and Ames, A. D. Human-inspired underactuated bipedal robotic walking with amber on flat-ground, up-slope and uneven terrain. In *International Conference on Intelligent Robots and Systems* (2012), pp. 2478–2483.

[37] Sandia National Laboratories. Lifelike, cost-effective robotic sandia hand can disable ieds. `https://share.sandia.gov/news/resources/news_releases/robotic_hand/`, Aug. 5, 2012.

[38] Sardain, P., and Bessonnet, G. Forces acting on a biped robot. center of pressure-zero moment point. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions o 34*, 5 (Sep 2004), 630–637.

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

[40] Sentis, L., Park, J., and Khatib, O. Compliant control of multi-contact and center of mass behaviors in humanoid robots. *IEEE Transactions on Robotics 26* (Dec 2010), 1–14.

[41] Shih, C., Grizzle, J. W., and Chevallereau, C. Asymptotically stable walking of a simple underactuated 3D bipedal robot. In *33rd Annual Conf. of the IEEE Industrial Electronics Society (IECON)* (Taipei, Nov. 2007), pp. 2766–2771.

[42] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. *Robotics: Modelling, Planning and Control*. Springer, London, England, 2008.

[43] Sinnet, R. Hybrid geometric reduction feedback control of three-dimensional bipedal walkers with knees and feet. Master's thesis, Texas A&M University, College Station, TX, May 2011.

[44] Sinnet, R., Powell, M., Shah, R., and Ames, A. D. A human-inspired hybrid control approach to bipedal robotic walking. In *18th IFAC World Congress* (Milano, Italy, 2011), pp. 6904–6911.

[45] Sinnet, R. W., and Ames, A. D. 2D bipedal walking with knees and feet: A hybrid control approach. In *Joint 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf.* (Shanghai, Dec. 2009), pp. 3200–3207.

[46] Sinnet, R. W., and Ames, A. D. 3D bipedal walking with knees and feet: A hybrid geometric approach. In *48th IEEE Conference on Decision and Control* (Shanghai, P.R. China, 2009), pp. 3208–3213.

[47] Smirnov, G. *Introduction to the Theory of Differential Inclusions*. American Mathematics Society, Providence, Rhode Island, 2002.

[48] Spong, M. W. Passivity based control of the compass gait biped. In *IFAC World Congress* (Beijing, Nov 1999).

[49] Spong, M. W., and Bullo, F. Controlled symmetries and passive walking. *IEEE TAC 50*, 7 (2005), 1025–1031.

[50] Spong, M. W., Hutchinson, S., and Vidyasagar, M. *Robotic Modeling and Control*. John Wiley and Sons, 2005.

[51] Spong, M. W., and Vidyasagar, M. *Robotic Dynamics and Control*. John Wiley and Sons, New York, NY, 1989.

[52] Sreenath, K., Park, H., I.Poulakakis, and Grizzle, J. W. Design and experimental implementation of a compliant hybrid zero dynamics controller for walking on mabel. In *49th IEEE Conference on Decision and Control(CDC)* (Dec. 2010), pp. 280–287.

[53] Stephens, B., and Atkeson, C. Push recovery by stepping for humanoid robots with force controlled joints. In *International Conference on Humanoid Robots* (Nashville, Tennessee, 2010), pp. 52–59.

[54] Sutherland, D. H., Kaufman, K. R., and Moitoza, J. R. *Human Walking*, 3 ed. Lippincott Williams & Wilkins, Baltimore, Dec 2005.

[55] Takubo, T., Imada, Y., Ohara, K., Mae, Y., and Arai, T. Rough terrain walking for bipedal robot by using zmp criteria map. In *IEEE Conference on Robotics and Automation.* (Kobe Japan, May 2009), pp. 788–793.

[56] Vapnyarskii, I. *Lagrange Multipliers.* Encyclopedia of Mathematics. Berlin,Germany, July 1995.

[57] Vukobratović, M., and Borovac, B. Zero-moment point—thirty-five years of its life. *Intl. Journal of Humanoid Robotics 1*, 1 (2005), 157–173.

[58] Wei, Q. F., Krishnaprasad, P. S., and Dayawansa, W. P. Modeling of impact on a flexible beam. In *2nd IEEE Conference on Decision and Control* (San Antonio, TX, 1993), vol. 2, pp. 1377–1382.

[59] 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.

[60] Westervelt, E. R., Grizzle, J. W., and Koditschek, D. E. Hybrid zero dynamics of planar biped walkers. *IEEE TAC 48*, 1 (2003), 42–56.

[61] Yujiang, X., Jasbir, A., Hyun-Joon, C., Hyun-Jung, K., Salam, R., Rajankumar, B., and Karim, A. Predictive simulation of human walking transitions using an optimization formulation. *Structural and Multidisciplinary Optimization 45*, 1–14.

[62] Zhang, F. *The Schur Complement and Its Applications*, vol. 4 of *Numerical Methods and Algorithms Series.* Springer, NY, March 2005.

[63] Zhau, H., Yadukumar, S. N., and Ames, A. D. Bipedal robotic running with partial hybrid zero dynamics and human-inspired optimization. In *IEEE International Conference on Intelligent Robots and Systems* (Portugal, Oct. 2012), pp. 1821–1827.