REALIZING TORQUE CONTROLLERS FOR UNDERACTUATED BIPEDAL WALKING USING THE IDEAL MODEL RESOLVED MOTION METHOD

A Thesis

by

ERIC ANDREW COUSINEAU

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,	Nancy Amato
	Reza Langari
Head of Department,	Andreas Polycarpou

December 2014

Major Subject: Mechanical Engineering

ABSTRACT

This thesis presents an application of *hybrid zero dynamics* to realize underactuated bipedal walking on DURUS, a testbed designed and built by SRI International. The main contribution of this work is the *ideal model resolved motion method* (IM-RMM), which is a simple method to convert ideal torque controllers to PD controllers to implement on hardware. Walking was first achieved using the proven method of the hybrid zero dynamics (HZD) reconstruction, followed by the Input-Output Feedback Linearization (IO) and Rapidly Exponentially Stabilizing Control Lyapunov Function Quadratic Programs (CLF-QPs) torque controllers implemented via IM-RMM.

The simulation and experimental results are presented and compared, and the best resulting *specific cost of electrical transport* on hardware was computed as 0.63 for the CLF-QP IM-RMM controller, and the record for walking was achieved on a separate occasion with the same CLF-QP IM-RMM controller, which yielded walking for 2 hours and 53 minutes, covering 7 km.

DEDICATION

To my parents, Jim and Anna Cousineau, for never ceasing in their love, encouragement, and support as I have grown through my education.

ACKNOWLEDGEMENTS

I would first like to thank my parents for their endless and unconditional support during my entire education, from actively being involved in my education from kindergarten to high school, to continually encouraging throughout my Bachelors of Science and my Masters of Science in Mechanical Engineering. I would like to thank both of my brothers: my older brother, who when I was 8 years old introduced me to advanced mathematics and computing and started me on the track of engineering and helping me attend Texas A&M University, and my little brother, Brian, for growing into an amazing young man, now pursuing his Bachelors of Science in Electrical Engineering.

I am deeply indebted to the guidance and support from my advisor, Dr. Aaron Ames, for his enthusiasm for theoretically succinct and experimentally practical control, and his guidance throughout the projects that I had the privilege of working on. I have Dr. Suh to thank for introducing me to Dr. Ames and encouraging me to pursue graduate school. I owe many thanks to my lab mates, Ayonga Hereid, Matthew Powell, Jordan Lack, Ryan Sinnet, Shishir Yadukumar, Murali Pasupeletii, Shawanee Patrick, Wen-Loong Ma, Huihua Zhao, Jonathan Horn, Jake Reher, Aakar Mehra, Forrest Berg, Eric Ambrose, and the undergraduates who have helped, Alejandro Azocar, Nathan, John, and William Lyles. They all helped foster a productive, relaxed, and enjoyable environment.

I must thank Dr. Benjamin Morris for his guidance during the DARPA Robotics Challenge project, and for fostering the initial idea of implementing an internal model forward dynamics integration. I must also thank Dr. Gwendolyn Johnson and Dr. J.D. Yamokoski for the initial implementation of integrating forward the dynamics

from the actual state.

I also have to thank the dedicated researchers at SRI International, specifically Curt Salisbury, Paul Birkmeyer, Zachery Shivers, Allegra Shum, and Stephen Morfey; the researchers at Oregon State University, Dr. Jonathan Hurst, Christian Hubicki, Mikhail Jones, and Ryan van Why; and the dedicated and amazing researchers that I had the privilege of working with at the NASA Johnson Space Center for the DARPA Robotics Challenge: Nicholas Radford, Chris McQuin, Dr. J.D. Yamokoski, Dr. Stephen Hart, Dr. Kim Hambuchen, Phillip Strawser, James Holley, Brice Howard, Chad Tobler, Dr. Gwendolyn Johnson, Joshua Mehling, Jairo Sanchez, Kris Verdeyen, Brett Somers, Vienny Nguyen, Lydon Bridgewater, Dr. Reg Burka, Adam Parsons, Brian Wightman, Paul Dinh, Dr. Chien-Liang Fok, Nicholas Paine, Dr. Sentis, Charles Blakely, Barret Ames, David Chesney, Kevin Bass, Stuart Donnan, Jairo Sanchez, Mike Conover, Andrew Lee, Cody Iven, Shelly Storz, and Lei Niu.

I would not be where I am today without these people that have shaped my life, professionally and personally. Through the highs and the lows, they have helped me to learn what it is to be consistent, passionate, and maintain sight of what is truly important in life.

NOMENCLATURE

Note that scalars are represented in plain italics, while bold-faced characters represent either vectors or matrices, following in suite with [15]. Typically, lowercase letters are vectors, and uppercase letters are matrices, though there are a few exceptions (such as $\boldsymbol{F}$, $\boldsymbol{g}$, etc.).

Linear Algebra

$\mathbb{R}$	Set of all real scalars
$\mathbb{R}^a$	Set of all real column-vectors of size a
$\mathbb{R}^{a \times b}$	Set of all real matrices of size $a \times b$
$\boldsymbol{A}_{a \times b}$	Matrix in $\mathbb{R}^{a \times b}$
$\mathbf{0}_{a \times b}$	Matrix of zeros in $\mathbb{R}^{a \times b}$
$\mathbf{1}_{a \times b}$	Matrix of ones in $\mathbb{R}^{a \times b}$
$\boldsymbol{I}_{n \times n}$	Identity matrix in $\mathbb{R}^{n \times n}$
v_i	Element i of vector $\boldsymbol{v}$
A_{ij}	Element in row i of column j of matrix $\boldsymbol{A}$
$\boldsymbol{v}_{j:k}$	A slice of a vector $\boldsymbol{v}$ from i to k, syntax inspired from MATLAB. Formally, $[v_i]_{i \in \mathcal{O}}$, with indexing set $\mathcal{O} = \{j, j+1, ..., k\}$.
$\boldsymbol{A}_{a:b,c:d}$	Submatrix of $\boldsymbol{A}$, formally $[A_{ij}]_{i \in \mathcal{O}_1, j \in \mathcal{O}_2}$, where $\mathcal{O}_1 = \{a, ..., b\}$, $\mathcal{O}_2 = \{c, ..., d\}$.

Kinematics

n	Degrees of freedom
$\boldsymbol{q}$	Generalized coordinates, $\boldsymbol{q} \in \mathbb{R}^n$
$\boldsymbol{x}$	State-space, $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})^T \in \mathcal{TQ} \subset \mathbb{R}^{2n}$
$\mathcal{TQ}$	Tangent bundle of $\mathcal{Q}$
$\boldsymbol{J}$	Jacobian
$\dot{\square}$	Total time derivative, $\dot{\square} = \Delta t \square$
$\mathring{\square}$	Apparent (partial) time derivative, $\mathring{\square} = \frac{\partial \square}{\partial t}$
$\overset{\circ\circ}{\square}$	Second apparent (partial) time derivative, $\overset{\circ\circ}{\square} = \frac{\partial^2 \square}{\partial t^2}$
c_θ, s_θ	Abbreviations for $\cos\theta$ and $\sin\theta$, respectively
${}^{B}\boldsymbol{R}_{A}$	Linear rotation in $SO(3)$, projecting coordinates in orientation of frame A to the orientation of frame B
${}^{B}\boldsymbol{T}_{A}$	Linear homogeneous transformation in $SO(3)$ for transforming the coordinates of a point in frame A to coordinates of frame B

Kinetics

m	Number of inputs to the physical system
$\boldsymbol{u}$	Physical inputs, $\boldsymbol{u} \in \mathbb{R}^m$
$\mathcal{I}$	Scalar inertia, $\mathcal{I} \in \mathbb{R}$
$\boldsymbol{\mathcal{I}}$	Rotational inertia tensor, $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{3\times3}$
$\boldsymbol{\mathcal{M}}$	Mass matrix, including rotation and translation, $\mathcal{M} \in \mathbb{R}^{6\times6}$

Control

p	Number of outputs
$\boldsymbol{y}$	Output error
$\boldsymbol{y}^a$	Actual output
$\boldsymbol{y}^d$	Desired output
$\boldsymbol{h}(\cdot)$	Output function (error), actual minus desired
$\boldsymbol{h}^a(\cdot)$	Actual output function
$\boldsymbol{h}^d(\cdot)$	Desired output function
$L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})$	Lie-derivative of $\boldsymbol{h}(x)$ along $\boldsymbol{f}(\boldsymbol{x})$, $L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x}) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})$
$\boldsymbol{\alpha}$	Parameter set, a tuple of vectors
$\boldsymbol{\Phi}(\boldsymbol{x})$	Coordinate transform to output and zero dynamics state space coordinates
$\boldsymbol{\eta}$	Output state-space coordinates, $\boldsymbol{\eta} = (\boldsymbol{y}, \dot{\boldsymbol{y}})^T$
$\boldsymbol{\xi}$	Zero dynamics coordinates
$\boldsymbol{\mu}$	Output-space inputs
$\boldsymbol{\mathcal{A}}$	Feedback linearization decoupling matrix
$L_{\boldsymbol{f}}$	Feedback linearization feedforward term
ε	Feedback control gain, where a larger value increases the rate of convergence

Hybrid Systems

$\bar{\Sigma}$	Dynamics defining the discrete and continuous dynamics

$\boldsymbol{\Delta}_R(\boldsymbol{x})$	Reset map, mapping $\boldsymbol{x}^-$ to $\boldsymbol{x}^+$
$\boldsymbol{\Delta}_{\boldsymbol{q}}(\boldsymbol{q})$	Reset map, mapping $\boldsymbol{q}^-$ to $\boldsymbol{q}^-$
$\boldsymbol{\Delta}_{\dot{\boldsymbol{q}}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$	Reset map, mapping $\dot{\boldsymbol{q}}^-$ to $\dot{\boldsymbol{q}}^-$
$\boldsymbol{\mathcal{R}}$	Relabeling map, mapping $\boldsymbol{q}^-$ to $\boldsymbol{q}^+$, $\dot{\boldsymbol{q}}^\pm$ to $\dot{\boldsymbol{q}}^+$
$\boldsymbol{P}_{\dot{\boldsymbol{q}}}$	Dynamically-consistent null space projection, mapping $\dot{\boldsymbol{q}}^-$ to $\dot{\boldsymbol{q}}^\pm$
$\square^-$	A value at pre-impact, pre-relabeling (the end of a step in single-domain walking)
$\square^\pm$	A value at post-impact, pre-relabeling
$\square^+$	A value at post-reset, post-impact, post-relabeling (the beginning of a step in single-domain walking)
$\boldsymbol{\varphi}_t(\boldsymbol{x})$	The flow of a solution at time t starting from $\boldsymbol{x}$
$\boldsymbol{\mathcal{P}}_{\boldsymbol{\varphi}}(\boldsymbol{x}^-)$	The Poincaré return map of pre-impact state

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Robotics is seeing a large surge in revolutionary control methods to physically realize complex behaviors. Through disasters such as the Fukushima Daichii nuclear reactor meltdown [69], it has become increasingly apparent that our technology, albeit very advanced, still comes short of being effective when deployed in these types of situations. The Fukushima disaster resulted in a major accident (INES Level 7), where a substantial amount of damage from the meltdown resulted from "cascading nuclear disaster" that may have been mitigated had operators been able to restore backup generators by pumping out flood water, and turning the release valve in the first reactor room to vacate hydrogen gas into the atmosphere. The conditions proved too dangerous for humans, and too difficult for robots at the time to perform. Contests, specifically the DARPA Robotics Competition[1], have focused on this apparent gap in technology and the needs present, and have fostered the continued growth of robotics in the realm of mobile manipulation. Advances have also been seen in the area of assistance, specifically prosthetic devices [8, 24] and exoskeletons [64].

In the area of robotic locomotion, there are a plethora of successes to examine [29], including novel robots such as the WL- and WABIAN-series of robots from Waseda University [31, 74], the running Planar Biped [22], Tad McGeer's passive dynamic walker [40], Honda ASIMO [21], the HRP series [30], Spring Flamingo [51], M2-V2 [50], jogging Johnie [47], Wisse's passive walker with an upper body [70] COMAN [36, 62], the DLR biped [44], Roboray [33], PETMAN [43], SARCOS Primus [68], ATLAS [58, 14], and several others. For underactuated bipedal robots, walking using

---

[1] `http://www.theroboticschallenge.com/`

hybrid zero dynamics has been achieved on the following as early as 2003: RABBIT [65, 66, 67], MABEL [17] (2009), ATRIAS [20] (a.k.a. MARLO [11]), ERNIE [38], AMBER 1 [3, 71], and during the phase immediately preceding heel strike phase in multi-contact walking for AMBER 2 [75, 37, 35, 56].

The goal of this research is to contribute towards the technology in robotics by showing the feasibility of implementing robust, physically realizable, energy efficient walking. The main objective of this thesis is to introduce the simple concept of the Ideal Model Resolved Motion Method (IM-RMM), which implements an ideal torque control by integrating forward the ideal closed-loop dynamics starting from the actual state measured on hardware to produce desired positions and velocities to send to hardware. To paraphrase Dr. Jonathan Hurst in conversation, no matter how much fidelity a model may contain, it will never be perfect. This provides the motivation: to prototype the feedback behavior of theoretically stable, model-dependent torque controllers without the need for extensive system identification as was done in [45].

The nature of resolving a desired position and velocity from a torque control model motivates the name inherited from the resolved motion method [55] used for inverse kinematics. To the authors' knowledge, there is no other formally published evidence of underactuated biped walking that has been achieved via IM-RMM using PD control, despite the technique's simplicity. In several surveys of robotic control and locomotion, with topics covering operational-space task control [55], general motion control [25], force control [63], robust industrial control [34, 1], feedback linearization [18], ZMP control [55], and local neuromuscular control [53], there are no techniques that appear to be the same as IM-RMM. There are methods that incorporate the integration of reduced-order models to produce trajectories for planning. These methods are employed in spring-loaded inverted pendulum (SLIP) models, such as in [16], with an example of code implementation published by Oregon State

University's Dynamic Robotics Laboratory in the ATRIAS code repository.[2] Another example is with the simplified linear inverted pendulum model (LIPM), which is used to realize step-based push recovery by planning a trajectory for the center of pressure (CoP) using Model-Predictive Control in [61]. The most similar technique that the author has found is the Nonlinear Model Predictive Control methods presented in [9], referencing [2], which was applied on a simulated model of the underactuated robot, RABBIT. This method is effectively a Nonlinear Model Predictive Control method, with the caveat that the time horizon is only one control cycle forward, with no active minimization taking place with respect to the horizon. For this reason, the author believes this to be a novel approach to achieving underactuated bipedal walking.

This thesis is organized as follows: Section 2 provides a general theoretical basis for modeling robotic manipulators, modeling walking mechanisms using the formality of hybrid systems, the human-inspired control framework (including the optimization), and the ideal model resolved motion method (IM-RMM) used to prototype ideal torque controllers and realize underactuated bipedal walking on physical hardware. Section 4 first introduces the experiment platform used to realize bipedal walking: DURUS, the prototype walking platform designed and built by SRI International[3]. Given the motivation of having the low-level control system provided by SRI, a discussion of the technologies used to implement the system are briefly discussed. Afterwards, the results are discussed in Section 5. The thesis concludes in Section 6, restating the contribution of this work and presenting future directions that will be taken in the overarching project of which this thesis is a part.

---

[2] `http://bit.ly/1pwHp7S`

[3] `http://www.sri.com/`

2. THEORY

The theory for the modeling and control can be very involved, and therefore requires sufficient motivation for sacrificing time and funds towards synthesizing these tools. This section provides a simple, concise review of the concepts that are used to construct the human-inspired framework [3], and how this framework is applied to the two model configurations of the hardware testbed, DURUS. The controllers discussed in this section are constrained to three of the well-published control techniques employed in the AMBER Lab: (a) the Hybrid Zero Dynamics (HZD) reconstruction, (b) input-output partial feedback linearization, and (c) Rapidly Exponentially Stabilizing Control Lyapunov Function Quadratic Programs (CLF-QPs). Following these controllers, the ideal model resolved motion method is introduced, including how it is incorporated into the real-time control system.

2.1 Modeling

While the abstract mathematics of control is concise and can provide an amazing wealth of information, this information has no significance if the mathematics are not tied to an objective, concrete model of a system with clear indications of how the aspects of the model relate to reality.

The modeling of robotic manipulators has been covered in depth, with a variety of techniques that afford a research formal frameworks in which to specify and validate said models. Modeling is also subdivided into two sections: *kinematic modeling*, which involves the positions, velocities, and accelerations of a system given the *degrees of freedom* and *constraints* governing the kinematics; and *kinetic modeling* or *dynamics modeling* that govern the relation among the positions, velocities, and accelerations, and the internal and external forces that occur for a system due to its physical inertial parameters, relations that constrain a system's movement along surfaces defining the system's energy.

2.1.1 Kinematic Modeling

An important first step to modeling is establishing the conventions that govern the kinematics and kinetics. As stated previously, this involves stating the *degrees of freedom* and the *kinematic constraints*. In this thesis, solely kinematic degrees of freedom are considered in the modeling of mechanical system, and only *revolute* and *prismatic* joints are considered.

To facilitate a more concise definitions of these terms, the following symbols are provided: n is the number of degrees of freedom available to the system, and $\boldsymbol{q} \in \mathbb{R}^n$ is a vector of *generalized coordinates* which contain the states of the degrees of freedom with respect to the *reference configuration*, which is used for defining the "home" state of the robot. A more in-depth review of kinematic modeling may be found in Section B.

2.1.2 Kinetic Modeling

Given the positions and velocities, and assuming that the angular and linear velocity Jacobians are available, the dynamics of a system may be computed. First, for each body i, define the *body-frame spatial velocity*, $\hat{\boldsymbol{v}}_i$, at the center of mass of a body driven by joint i, as in [42, 15]:

$$\hat{\boldsymbol{v}}_i = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix}, \tag{2.1}$$

and the *mass matrix*:

$$\boldsymbol{\mathcal{M}}_i = \begin{bmatrix} m_i \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{\mathcal{I}}_i \end{bmatrix}, \tag{2.2}$$

where m_i is the body mass and $\boldsymbol{\mathcal{I}}_i \in \mathbb{R}^{3\times3}$ is the rotational inertia aligned with the body frame centered at the center of mass. The rotational and linear kinetic energy of the system can be defined as:

$$\mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\sum_{i=1}^{n} \hat{\boldsymbol{v}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}})^T \boldsymbol{\mathcal{M}}_i \hat{\boldsymbol{v}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{2.3}$$

The potential energy can be computed as:

$$\mathcal{V}(\boldsymbol{q}) = \sum_{i=1}^{n} m_i g({}^0\boldsymbol{z} \cdot {}^0\boldsymbol{p}_i), \tag{2.4}$$

where $g = 9.81\text{m/s}^2$ is the gravitational constant, ${}^0\boldsymbol{z}$ is the direction facing opposite of gravity, and ${}^0\boldsymbol{p}_i$ is the position of the center of mass of the body i.

With this information, the *Euler-Lagrange equation of motion* may be derived first using the *Lagrangian*,

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{V}(\boldsymbol{q}), \tag{2.5}$$

and incorporating it into the equation,

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}} = \boldsymbol{\Gamma}, \tag{2.6}$$

where $\boldsymbol{\Gamma}$ are the generalized external forces. In the case for control, we have $\boldsymbol{\Gamma} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}$, where $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n\times m}$ is the actuation matrix and $\boldsymbol{u} \in \mathbb{R}^m$ is the vector of a total of m physical inputs available to a mechanical system. This yields the equation for the *unconstrained dynamics*:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}, \tag{2.7}$$

where $\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})$ and $\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial \mathcal{V}}{\partial \boldsymbol{q}}(\boldsymbol{q})$.

Constrained dynamics is not use thoroughly in this thesis, but because it is a topic of importance, it is reviewed in Section C, which introduces the notation $\boldsymbol{h}_c(\boldsymbol{q}) : \mathcal{Q} \to \mathbb{R}^{n_c}$ for a *holonomic constraint*, where n_c is the number of constraints in the system.

2.1.3 Mechanical Hybrid Systems

As shown in [3], symmetric, rigid, planar, underactuated bipedal walking can most concisely represented as a one domain hybrid system. A *hybrid system* is a mathematical construct that defines a system composed of continuous dynamics in domains that are separated by discrete transitions.

A mechanical system of state space dimension $2n$ has a configuration space $\mathcal{Q} \subset \mathbb{R}^n$, with coordinates $\boldsymbol{q} \in \mathcal{Q}$, and a tangent bundle $T\mathcal{Q} \subset \mathbb{R}^{2n}$, where $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})^T$ represents the state space with $\boldsymbol{x} \in T\mathcal{Q}$. Informally, the *tangent bundle* is the set of generalized positions and velocities that are achievable within the defined domain of a system.

A single-domain and single-transition *hybrid control system* can be formally defined as the tuple:

$$\mathcal{HC} = (\mathcal{D}, \mathcal{U}, \mathcal{S}, \boldsymbol{\Delta}_R, \boldsymbol{f}, \boldsymbol{g}), \tag{2.8}$$

where:

- $\mathcal{D} \subset T\mathcal{Q}$ is the *domain*,
- $\mathcal{U} \subset \mathbb{R}^m$ is *set of admissible controls*, with $\boldsymbol{u} \in \mathcal{U}$,

- $\mathcal{S} : \mathcal{D} \to \mathbb{R}$ defines the *switching surface* at which the discrete transition occurs,
- $\boldsymbol{\Delta}_R : \mathcal{D} \to \mathcal{D}$ is a smooth *reset map* that occurs at the surface $\mathcal{S}$, and
- $\boldsymbol{f} : T\mathcal{Q} \to \mathbb{R}^{2n}$, the *drift vector field*, and $\boldsymbol{g} : T\mathcal{Q} \to \mathbb{R}^{2n\times m}$, the *input map*, are smooth vector fields that define the first-order dynamics of the system,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}. \tag{2.9}$$

In the case of a second-order rigid body system, we define the vector fields as:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{D}(\boldsymbol{q})^{-1}(\boldsymbol{B}\boldsymbol{u} - \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}})) \end{bmatrix}, \boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0}_{n\times m} \\ \boldsymbol{D}(\boldsymbol{q})^{-1}\boldsymbol{B} \end{bmatrix}. \tag{2.10}$$

The full definition of the reset map, $\boldsymbol{\Delta}_R(\boldsymbol{x}^-)$, incorporates an impact model and state relabeling, both of which are defined in Sec. 2.1.3.1 and Sec. 2.1.4 due to model-specific considerations.

2.1.3.1 Impact Model

In the context of mechanical control systems, the guard surface $\mathcal{S}$ is encountered transition due to a change in the physical constraints that are active for a given system due to impact. This thesis follows the conventions from the rest of the virtual constraints walking literature, modeling impacts as perfectly plastic and instantaneous as shown in [67, p. 55], given pre-impact velocities, $\dot{\boldsymbol{q}}^-$, a new constraint, $\boldsymbol{h}_c(\boldsymbol{q}_e)$, $\boldsymbol{h}_c : \mathbb{R}^{n_e} \to \mathbb{R}^{n_c}$, the post-impact velocities, $\dot{\boldsymbol{q}}^\pm$, are computed with the following assumptions for the impulse due to impact, where the impact:

1. occurs instantaneously,
2. does not affect the positions, $\boldsymbol{q}^\pm = \boldsymbol{q}^-$, and

3. conserves momentum,

4. satisfies the new constraint that models the impact.

Mathematically, the conservation is realized as

$$\boldsymbol{D}_e(\boldsymbol{q}_e)(\dot{\boldsymbol{q}}_e^{\pm} - \dot{\boldsymbol{q}}_e^{-}) = \boldsymbol{J}_c^T(\boldsymbol{q}_e)\boldsymbol{\mathcal{F}}_{ext} \tag{2.11}$$

where $\boldsymbol{J}_c(\boldsymbol{q}_e) = \frac{\partial \boldsymbol{h}_c}{\partial \boldsymbol{q}_e}(\boldsymbol{q}_e)$ is the Jacobian of the constraints, and $\boldsymbol{\mathcal{F}}_{ext} \in \mathbb{R}^{n_c}$ is the vector of external impulsive forces that occur over the instantaneous instant of impact due to the constraint $\boldsymbol{h}_c$. The satisfaction of the constraint is modeled as

$$\boldsymbol{J}_c(\boldsymbol{q}_e)\dot{\boldsymbol{q}}_e^{+} = \mathbf{0}_{n_c}. \tag{2.12}$$

This is a form of the Lagrange multiplier equation, which lends itself to the solution:

$$\begin{bmatrix} \dot{\boldsymbol{q}}_e^{\pm} \\ \boldsymbol{\mathcal{F}}_{ext} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_{\dot{\boldsymbol{q}}_e} \\ \boldsymbol{P}_{\boldsymbol{\mathcal{F}}} \end{bmatrix} \dot{\boldsymbol{q}}_e^{-} \tag{2.13}$$

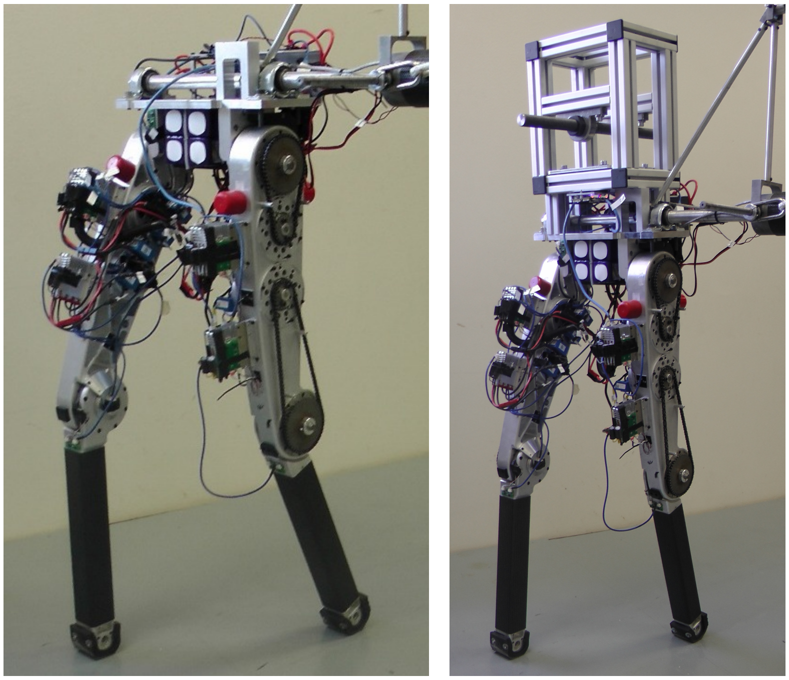
$$\boldsymbol{P}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{q}_e) = -(\boldsymbol{J}_c \boldsymbol{D}_e^{-1} \boldsymbol{J}_c^T)^{-1} \boldsymbol{J}_c \tag{2.14}$$

$$\boldsymbol{P}_{\dot{\boldsymbol{q}}_e}(\boldsymbol{q}_e) = \boldsymbol{D}_e^{-1} \boldsymbol{J}_c^T \boldsymbol{P}_{\boldsymbol{\mathcal{F}}} + \boldsymbol{I}_{n_e \times n_e} \tag{2.15}$$

where $\boldsymbol{P}_{\boldsymbol{\mathcal{F}}}$ projects the pre-impact velocities to the impulsive forces required to enforce a change in the velocities, and $\boldsymbol{P}_{\dot{\boldsymbol{q}}_e}$ projects the pre-impact velocities to the dynamically consistent null-space of the constraint yielding the null-space velocities, as stated in [27], which is equivalent to $\bar{\boldsymbol{N}}_c(\boldsymbol{q}_e)$ shown in Sec. C.1.

2.1.4 Testbed Overview

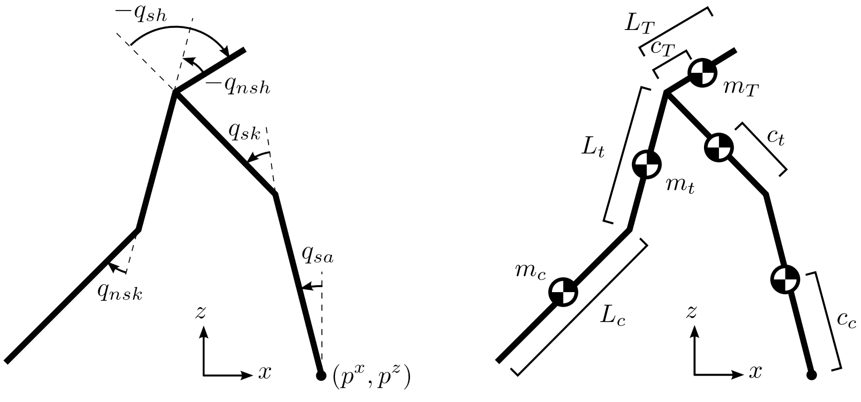
This section provides a description of the physical testbed to provide the motivation for the modeling chosen. The testbed in this thesis is DURUS, designed and built by SRI International, who also provided the low-level software necessary for real-time control. Two configurations of DURUS is shown in Fig. 2.1, one with a small torso, with a total mass of 21kg, and a configuration with a larger torso with a total mass of 31.5kg. The robot has a total of 5-links, with underactuated point feet, which motivates the use of the 5-link model with a torso. The corresponding kinematic and kinetic parameters are shown in Fig. 2.2, and the parameters themselves are defined in Table 2.1 which were obtained from measurements in SolidWorks and on the physical hardware.



(a) Light Torso

(b) Heavy Torso

Figure 2.1: Two configurations of DURUS, built by SRI International.



(a) Coordinates (b) Parameters

Figure 2.2: A diagram of the 5-link biped.

The model is considered to be planar due to the constraint of the boom attached, the same boom as was used for AMBER 2 [37], designed and built by Michael Zeagler, with different counterweights due to the removal of unneeded equipment. The testbed workstation is shown in Fig. 2.3, which includes a larger view of the boom. An important note, in this case, is that the vectors are still in three dimensions. This allows for the model to be extended to three dimensions by simply extending the model along the y-axis and using the same rotations about this axis. Such work was used in [48].

The extended configuration coordinates for this model are $\boldsymbol{q}_e = (\boldsymbol{p}, \boldsymbol{q})^T$, composed of:

$$\boldsymbol{q} = \begin{bmatrix} q_{sa} & q_{sk} & q_{sh} & q_{nsh} & q_{nsk} \end{bmatrix}^T \tag{2.16}$$

$$\boldsymbol{p} = \begin{bmatrix} p^x & p^z \end{bmatrix}^T, \tag{2.17}$$

where the “s” prefix stands for the stance (supporting) leg, the “ns” prefix stands for

Table 2.1: Model Parameters.

Label (subscript)	L (m)	m (kg)	c^x (m)	c^z (m)	$\mathcal{I}^y$ ($\text{kg} \cdot \text{m}^2$)
Light Torso					
Calf (c)	0.43	1.2	0	0.234	0.035
Thigh (t)	0.43	8.8	0	0.218	0.18
Torso (T)	0.13	1.72	0.002	0.048	0.0096
Heavy Torso					
Calf (c)	0.46	1.2	0	0.266	0.035
Thigh (t)	0.43	8.8	0	0.218	0.18
Torso (T)	0.13	11.5	0.02	0.16	0.48

nonstance (swing) leg, and the subfixes “a”, “k”, and “h” stand for ankle, knee, and hip, respectively. The coordinates p^x and p^z represent the Cartesian coordinates of the stance foot. Note that the true floating-base coordinates, $\boldsymbol{q}_b$, should be considered as $(p^x, p^z, q_{sa})^T$, due to underactuation at the ankle.

Regarding the dynamics, the standard rigid body methods are used to define the robot using a parent-child relationship as in Sec. 2.1.1, which then pave way to the rigid body equations of motion shown in Sec. 2.1.2. The reduction to the local coordinates, $\boldsymbol{q}$, is due to the model being built from the foot. Since the stance foot is pinned such that $\boldsymbol{h}_c(\boldsymbol{q}_e) = \boldsymbol{p}$, where $\boldsymbol{p}$ are direct coordinates in $\boldsymbol{q}_e$, equivalent dynamics can be achieved by reducing the coordinates. A quick review of this is shown in the appendix (Section C).

The hardware has four actuators, geared with a novel low-friction transmission from SRI International, placed at both of the knees and both of the hips. This results in the number of physical inputs being $m = 4$.

To more accurately represent the dynamics of the robot, the inertial effects of both the boom and the motors are incorporated. The boom, shown in Fig. 2.3, is

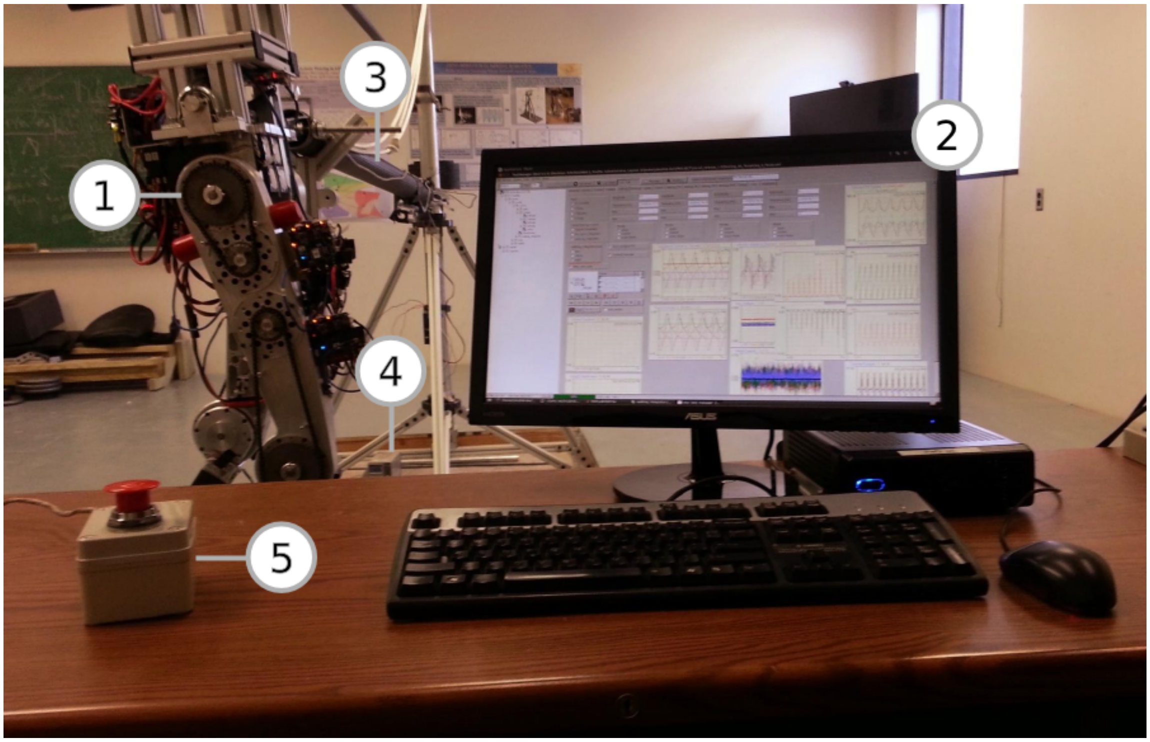


Figure 2.3: The setup used for operating DURUS (based on [37]), labeled as follows: (1) DURUS with motor power supplied by batteries onboard, (2) the control workstation, (3) the boom, used to constrain DURUS to the saggital plane, (4) logic power supply, and (5) the Emergency-Stop switch.

modeled as in [67, 37, 38], where the potential energy of the system is zero using counterweights, and the kinetic energy is incorporated into the inertia by assuming a linear relationship between the boom's roll, θ^{roll}, and yaw, θ^{yaw}, and the hip's vertical, p^{z}_{boom}, and horizontal position, p^{x}_{boom}. Given that the boom radius is $L_{boom} = 2.5$m and is attached a distance L_T above the hip, the expressions for positions are:

$$\theta^{yaw}(\boldsymbol{q}) = \frac{p^{x}_{boom}(\boldsymbol{q})}{L_{boom}} \tag{2.18}$$

$$\theta^{roll}(\boldsymbol{q}) = \frac{p^{z}_{boom}(\boldsymbol{q})}{L_{boom}}, \tag{2.19}$$

which then leads to incorporating the kinetic energy as:

$$\boldsymbol{D}_{boom}(\boldsymbol{q}) = \boldsymbol{J}_{boom}(\boldsymbol{q})^T \boldsymbol{\mathcal{I}}_{boom} \boldsymbol{J}_{boom}(\boldsymbol{q}) \tag{2.20}$$

$$\boldsymbol{\mathcal{I}}_{boom} = \begin{bmatrix} 2.53 & 0 \\ 0 & 1.66 \end{bmatrix} \mathrm{kg} \cdot \mathrm{m}^2 \tag{2.21}$$

$$\boldsymbol{J}_{boom}(\boldsymbol{q}) = \frac{\partial}{\partial \boldsymbol{q}} \begin{bmatrix} \theta^{yaw}(\boldsymbol{q}) \\ \theta^{roll}(\boldsymbol{q}) \end{bmatrix}. \tag{2.22}$$

The motors are incorporated by assuming that rotation occurs at the joints. First, the inertias are defined as $\mathcal{I}_{motor} = 3 \cdot 10^{-7} \mathrm{kg} \cdot \mathrm{m}^2$, with the gear ratios of $N_{motor} = 30.7$. This yields the apparent inertia, $N^2_{motor} \mathcal{I}_{motor}$, which is placed at each actuated joint, yielding the additional inertia matrix:

$$\boldsymbol{D}_{motor} = \begin{bmatrix} 0 & \boldsymbol{0}_{1\times m} \\ \boldsymbol{0}_{m\times 1} & N^2_{motor} \mathcal{I}_{motor} \boldsymbol{I}_{m\times m} \end{bmatrix}. \tag{2.23}$$

Let $\boldsymbol{D}_{nominal}(\boldsymbol{q})$ be the rigid body inertia derived from the nominal kinetic energy in Equation (2.3). Thus, the resulting inertia matrix, $\boldsymbol{D}(\boldsymbol{q})$, is now:

$$\boldsymbol{D}(\boldsymbol{q}) = \boldsymbol{D}_{nominal}(\boldsymbol{q}) + \boldsymbol{D}_{boom}(\boldsymbol{q}) + \boldsymbol{D}_{motor}. \tag{2.24}$$

The *switching surface* is defined for this system simply as:

$$\mathcal{S} = \{\boldsymbol{x} : h_R(\boldsymbol{q}) = 0,\ \dot{h}_R(\boldsymbol{q}, \dot{\boldsymbol{q}}) < 0\}, \tag{2.25}$$

where the guard $h_R(\boldsymbol{q})$ is defined as the vertical position of the nonstance foot,

$p^z_{nsf}(\boldsymbol{q})$. The smooth impact map is then defined using the extended coordinates as in Sec. 2.1.3.1, with the constraints:

$$\boldsymbol{h}_c(\boldsymbol{q}_e) = \begin{bmatrix} p^x_{e,nsf}(\boldsymbol{q}_e) \\ p^z_{e,nsf}(\boldsymbol{q}_e) \end{bmatrix}, \tag{2.26}$$

where $p^x_{e,nsf}(\boldsymbol{q}_e)$ and $p^z_{e,nsf}(\boldsymbol{q}_e)$ are the forward kinematics of x- and z- position of the nonstance foot using extended coordinates.

Since only one domain is considered in this symmetric model, the coordinates are relabeled when resetting. Given the symmetric modeling of the system, a *circular relabeling matrix*, $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{n \times n}$, is defined as in [67, p. 57] such that positions and velocities may be "projected" from a state when the stance leg switches, i.e., when the prior stance leg strikes the ground and the a new stance must be considered. If one considers the coordinates $\boldsymbol{q}^L$ and $\boldsymbol{q}^R$ as the stance-space coordinates when the stance is the left and right leg, respectively, then the following relations hold:

$$\begin{aligned} \boldsymbol{q}^R &= \boldsymbol{\mathcal{R}} \boldsymbol{q}^L \\ \dot{\boldsymbol{q}}^R &= \boldsymbol{\mathcal{R}} \dot{\boldsymbol{q}}^L \\ \boldsymbol{q}^L &= \boldsymbol{\mathcal{R}} \boldsymbol{q}^R \\ \dot{\boldsymbol{q}}^L &= \boldsymbol{\mathcal{R}} \dot{\boldsymbol{q}}^R, \end{aligned}$$

noting that the circular identity, $\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{R}} = \boldsymbol{I}_{n \times n}$, is held. In this case, the relabeling matrix is defined as:

$$\boldsymbol{\mathcal{R}} = \begin{bmatrix} 1 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{2.27}$$

The smooth reset map, $\boldsymbol{\Delta}_R(\boldsymbol{x}^-)$, is defined by first applying the impact map to the velocities then relabeling the angles, which yields:

$$\boldsymbol{\Delta}_R(\boldsymbol{x}^-) = \boldsymbol{x}^+ = \begin{bmatrix} \boldsymbol{q}^+ \\ \dot{\boldsymbol{q}}^+ \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}_{\boldsymbol{q}} \boldsymbol{q}^- \\ \boldsymbol{\Delta}_{\dot{\boldsymbol{q}}}(\boldsymbol{q}^-)\dot{\boldsymbol{q}}^- \end{bmatrix} \tag{2.28}$$

$$\boldsymbol{\Delta}_{\boldsymbol{q}} = \boldsymbol{\mathcal{R}} \tag{2.29}$$

$$\boldsymbol{\Delta}_{\dot{\boldsymbol{q}}}(\boldsymbol{q}^-) = \boldsymbol{\mathcal{R}} \boldsymbol{P}_{\dot{\boldsymbol{q}}}(\boldsymbol{q}^-) \boldsymbol{P}_{\dot{\boldsymbol{q}}}(\boldsymbol{q}^-) \qquad = \boldsymbol{\iota} \boldsymbol{P}_{\dot{\boldsymbol{q}}_e}(\boldsymbol{\iota}_e \boldsymbol{q}^-), \tag{2.30}$$

noting that $\dot{\boldsymbol{q}}^+ = \boldsymbol{\mathcal{R}} \dot{\boldsymbol{q}}^\pm$, where the projections:

$$\boldsymbol{\iota}_e = \begin{bmatrix} \boldsymbol{0}_{2\times n} \\ \boldsymbol{I}_{n\times n} \end{bmatrix}$$

maps from reduced coordinates to extended coordinates, and:

$$\boldsymbol{\iota} = \begin{bmatrix} \boldsymbol{0}_{n\times 2} & \boldsymbol{I}_{n\times n} \end{bmatrix}$$

maps from extended coordinates to reduced coordinates.

2.2 Ideal Controllers

This section provides the motivation for formal ideal torque controllers that guarantee exponential convergence: input-output feedback linearization (IO), and Rapidly Exponentially Stabilizing Control Lyapunov Function Quadratic Programs (CLF-QPs), presented in [6]. These controllers are then used to segue into the Human-Inspired Control section.

2.2.1 Partial Input-Output Feedback Linearization

The concept of *partial input-output feedback linearization*, as introduced in [52], provides a means to take the smooth, nonlinear dynamics of a system with a given set of inputs, provide a *control objective* of zeroing a set of *output errors*, and provide a coordinate transformation that first provides linear dynamics for the output error given an *output-space input*, and provides a means to map the output-space input to the *physical input*. Note that the outputs are also called *virtual constraints* introduced in [67, 3], which are constraints that are incorporated into the dynamics in a fashion similar to the holonomic constraints, except for the added feedback given that the constraints may not immediately be achieved.

First, we consider a system of n degrees of freedom, with a state space of size $2n$ and m inputs. We define an *output* as the error between an *actual output* of the system and the *desired output*, which may be time-based or state-based. For a *multi-input multi-output system*, one must define multiple outputs. Normally, in the interest of controlling as much of the system as possible without redundancy, $p = m$ outputs are selected.

The outputs are concatenated into a vector and represented using the function, $\boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{\alpha})$, $\boldsymbol{h} : \mathbb{T} \times T\mathcal{Q} \to \mathbb{R}^p$, where $t \in \mathbb{T}$, $\boldsymbol{\alpha} \in \mathbb{A}$ is the set of parameter vectors for each desired output, and $\mathbb{A}$ is the space of possible parameter sets. The motivation for including both time, t, and the state, $\boldsymbol{x}$, is to concisely allow purely time-based and state-based outputs. The actual outputs are defined as $\boldsymbol{h}^a(\boldsymbol{x})$, $\boldsymbol{h}^a : T\mathcal{Q} \to \mathbb{R}^p$, and the desired outputs may be defined as $\boldsymbol{h}^d(t, \boldsymbol{x}, \boldsymbol{\alpha})$, $\boldsymbol{h}^d : \mathbb{T} \times T\mathcal{Q} \times \mathbb{A}$. With these definitions, we have the relation:

$$\boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{\alpha}) = \boldsymbol{h}^a(\boldsymbol{x}) - \boldsymbol{h}^d(t, \boldsymbol{x}, \boldsymbol{\alpha}), \tag{2.31}$$

which may be defined component-wise as:

$$h_i(t, \boldsymbol{x}, \boldsymbol{\alpha}_i) = h_i^a(\boldsymbol{x}) - h_i^d(t, \boldsymbol{x}, \boldsymbol{\alpha}_i). \tag{2.32}$$

For congruence with existing control literature, we define final value of the output as $\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{\alpha})$, suppressing the parameters, where $\boldsymbol{y}^a(\boldsymbol{x}) = \boldsymbol{h}^a(\boldsymbol{x})$ and $\boldsymbol{y}^d(\boldsymbol{x}) = \boldsymbol{h}^d(t, \boldsymbol{x}, \boldsymbol{\alpha})$.

Next, once the outputs are selected, the *relative degree*, γ_i, of each output i must be determined. This is formally defined in [52] as the condition:

$$\begin{aligned} L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{(\gamma_i - 1)} y_i(\boldsymbol{x}) &\neq \mathbf{0} \\ L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{(k)} &\equiv \mathbf{0}, \quad k \leq \gamma_i - 2, \end{aligned}$$

where $L_{\boldsymbol{f}} \boldsymbol{h}(\boldsymbol{x})$ is the *Lie derivative operator* defined as in the nomenclature. However, it may be informally stated as the number of derivatives necessary such that the physical inputs are directly present in the expression without any separation due to integration. The relative degree of all outputs together is named the *vector relative degree*, $\boldsymbol{\gamma} = [\gamma_i]_{i \in \mathcal{O}_p}$, where $\mathcal{O}_p = \{1, ..., p\}$ is the *output indexing set.* For rigid mechanical systems, one may easily find the relative degree by taking derivatives until accelerations are present in the expression. The controllers considered in this thesis have all actual outputs that are purely position-based, thus all outputs are vector relative degree two, $\boldsymbol{\gamma} = \mathbf{2}$.

Next, the *coordinate transformation*, $\boldsymbol{\Phi}(\boldsymbol{x})$, is found in order to yield linear dynamics for the outputs. If we choose the full-rank, linearly independent output coordinates for partial feedback linearization:

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\eta}(\boldsymbol{x}) \\ \boldsymbol{\xi}(\boldsymbol{x}) \end{bmatrix} \tag{2.33}$$

$$\boldsymbol{\eta}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{y}(\boldsymbol{x}) \\ \dot{\boldsymbol{y}}(\boldsymbol{x}) \end{bmatrix},$$

where $\boldsymbol{\xi}(\boldsymbol{x})$ defines the zero dynamics coordinates (to be discussed in Sec. 2.3.2), our goal is then to find the dynamics of $\dot{\boldsymbol{\eta}}$. If we assume that we have control over the directly-controlled derivative as $\ddot{\boldsymbol{y}} = \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the *output-space input*, then we have the dynamics in terms of $\boldsymbol{\eta}$, $\boldsymbol{\mu}$, $\boldsymbol{F}$, and $\boldsymbol{G}$:

$$\dot{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} \mathbf{0}_{p\times p} & \boldsymbol{I}_{p\times p} \\ \mathbf{0}_{p\times p} & \mathbf{0}_{p\times p} \end{bmatrix}}_{\boldsymbol{F}} \boldsymbol{\eta} + \underbrace{\begin{bmatrix} \mathbf{0}_{p\times p} \\ \boldsymbol{I}_{p\times p} \end{bmatrix}}_{\boldsymbol{G}} \boldsymbol{\mu}. \tag{2.34}$$

Choosing the output-space feedback control law:

$$\boldsymbol{\mu}(\boldsymbol{x}) = -2\varepsilon\dot{\boldsymbol{y}}(\boldsymbol{x}) - \varepsilon^2\boldsymbol{y}(\boldsymbol{x}), \tag{2.35}$$

yields the closed-loop, *control canonical form*:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \mathbf{0}_{p\times p} & \boldsymbol{I}_{p\times p} \\ -\varepsilon^2 \boldsymbol{I}_{p\times p} & -2\varepsilon \boldsymbol{I}_{p\times p} \end{bmatrix} \boldsymbol{\eta}, \tag{2.36}$$

which is *Hurwitz* with all poles at $-\varepsilon$. This results in the time-solution of output errors that exponentially decay to $\mathbf{0}$ at a rate of ε, or more formally:

$$\boldsymbol{y}(t) = e^{-\varepsilon t}\boldsymbol{y}_0, \tag{2.37}$$

where $\boldsymbol{y}_0$ is the initial output error.

These dynamics are the desired *output dynamics*, which leads us to the means of solving for the physical inputs, $\boldsymbol{u}$, such that $\ddot{\boldsymbol{y}} = \boldsymbol{\mu}$, which we may define as transforming the output-space inputs to physical inputs. Given that we already have the output relative degree, we then solve for $\boldsymbol{u}$ with the goal of $\ddot{\boldsymbol{y}} = \boldsymbol{\mu}$. First, computing $\ddot{\boldsymbol{y}}$ in terms of the system dynamics, with the goal of obtaining the physical inputs:

$$\begin{aligned} \dot{\boldsymbol{y}}(\boldsymbol{x}) &= L_{\dot{\boldsymbol{x}}}\boldsymbol{y}(\boldsymbol{x}) = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}) = L_{\boldsymbol{f}}\boldsymbol{y}(\boldsymbol{x}) \\ \ddot{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{u}) &= L^2_{\dot{\boldsymbol{x}}}\boldsymbol{y}(\boldsymbol{x}) = \underbrace{L^2_{\boldsymbol{f}}\boldsymbol{y}(\boldsymbol{x})}_{L_{\boldsymbol{f}}(\boldsymbol{x})} + \underbrace{L_{\boldsymbol{g}}L_{\boldsymbol{f}}\boldsymbol{y}(\boldsymbol{x})}_{\boldsymbol{\mathcal{A}}(\boldsymbol{x})} \cdot \boldsymbol{u} = \boldsymbol{\mu}(\boldsymbol{\eta}), \end{aligned} \tag{2.38}$$

where $L_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})$ is the *Lie derivative operator* defined as in the nomenclature, $L_{\boldsymbol{f}}$ is the output dynamics along the physical passive dynamics, and $\boldsymbol{\mathcal{A}}$ is the *decoupling matrix* which allows for the physical inputs to be mapped to the output evolution. An important considerations is selecting $\boldsymbol{y}$ in the presence of the dynamics $\boldsymbol{f}$ and $\boldsymbol{g}$ such that $\boldsymbol{\mathcal{A}}$ is full row-rank.

This allows for $\boldsymbol{u}$ to be solved for as:

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\mathcal{A}}(\boldsymbol{x})^{-1}(-L_{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\mu}(\boldsymbol{\eta}(\boldsymbol{x}))), \tag{2.39}$$

which is an affine transformation. If $\boldsymbol{u}$ is selected according to (2.39), then the output dynamics will be realized on the physical system and yield the desired exponential convergence. This coordinate transformation, yielding an affine transformation from $\boldsymbol{\mu}$ to $\boldsymbol{u}$, can then be incorporated into more advanced forms of control. Note that the zero dynamics of the coordinate transformation, $\boldsymbol{\xi}(\boldsymbol{x})$, are discussed in the optimization in Sec. 2.3.2.

2.2.2 Rapidly Exponentially Stabilizing Control Lyapunov Function Quadratic Programs

Given the coordinate transformation from feedback linearization to linear output dynamics, an affine transformation from output-space input to physical inputs was made available. With this transformation, we may define a *Control Lyapunov Function* (CLF) to formally define convergence of the controlled outputs, and then use the *Lyapunov stability criteria* to pose a constraint in a quadratic program to formally guarantee exponential convergence [32]. Once this quadratic program is formulated, additional constraints and relaxations may be added to make a controller more physically feasible while relaxing the restrictions on rapid exponential convergence. This was first introduced in [4], and succinctly stated in [6].

First, given the coordinate transformation in (2.33) and the dynamics in (2.34), we may define the Lyapunov function for the controlled convergence rate, $\varepsilon > 0$, as:[1]

$$V_\varepsilon(\boldsymbol{\eta}) = \frac{1}{2}\boldsymbol{\eta}^T \boldsymbol{P}_\varepsilon \boldsymbol{\eta} \tag{2.40}$$

$$\boldsymbol{P}_\varepsilon = \boldsymbol{M}_\varepsilon^T \boldsymbol{P} \boldsymbol{M}_\varepsilon \tag{2.41}$$

$$\boldsymbol{M}_\varepsilon = \begin{bmatrix} \varepsilon \boldsymbol{I}_{p\times p} & \boldsymbol{0}_{p\times p} \\ \boldsymbol{0}_{p\times p} & \boldsymbol{I}_{p\times p} \end{bmatrix} \tag{2.42}$$

$$\boldsymbol{P} = \frac{1}{\sqrt{3}}\boldsymbol{I}_{2p\times 2p} + \boldsymbol{F} + \boldsymbol{F}^T \succ 0, \tag{2.43}$$

where $\boldsymbol{F}$ and $\boldsymbol{G}$ are defined in (2.34), and $\boldsymbol{P} \succ 0$ solves the *continuous time algebraic Riccati equation* (CARE):

$$\boldsymbol{F}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F} - \boldsymbol{P}\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{P} = -\boldsymbol{Q}, \tag{2.44}$$

with $\boldsymbol{Q} \succ 0$ selected as $\boldsymbol{Q} = \boldsymbol{I}_{2p\times 2p}$. Given this formulation of $V_\varepsilon(\boldsymbol{\eta})$, its derivative, $\dot{V}_\varepsilon(\boldsymbol{\eta}, \boldsymbol{\mu})$, may be found as:

$$\dot{V}_\varepsilon(\boldsymbol{\eta}, \boldsymbol{\mu}) = L_{\boldsymbol{F}} V_\varepsilon(\boldsymbol{\eta}) + L_{\boldsymbol{G}} V_\varepsilon(\boldsymbol{\eta})\boldsymbol{\mu} \tag{2.45}$$

$$L_{\boldsymbol{F}} V_\varepsilon(\boldsymbol{\eta}) = \boldsymbol{\eta}^T(\boldsymbol{F}^T\boldsymbol{P}_\varepsilon + \boldsymbol{P}_\varepsilon\boldsymbol{F})\boldsymbol{\eta} \tag{2.46}$$

$$L_{\boldsymbol{G}} V_\varepsilon(\boldsymbol{\eta}) = 2\boldsymbol{\eta}^T\boldsymbol{P}_\varepsilon\boldsymbol{G}, \tag{2.47}$$

which is affine in terms of $\boldsymbol{\mu}$. To enforce *rapid exponential stabilization*, we upper bound $\dot{V}_\varepsilon$ as:

$$\dot{V}_\varepsilon(\boldsymbol{\eta}, \boldsymbol{\mu}) \leq -\varepsilon\gamma V_\varepsilon(\boldsymbol{\eta}), \tag{2.48}$$

where $\gamma = \frac{\lambda_{min}(\boldsymbol{Q})}{\lambda_{max}(\boldsymbol{P})}$.

---

[1] The author would like to mention that the closed-form of (2.43) was found by Ryan Sinnet.

To now incorporate this as a *quadratic program*, we first choose to minimize the norm of the output error coordinates, $\boldsymbol{\eta}$, and thus have the formulation:

$$\begin{aligned} \boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \; & \boldsymbol{\mu}^T \boldsymbol{\mu} \\ s.t. \quad & \dot{V}_\varepsilon(\boldsymbol{\eta}, \boldsymbol{\mu}) \leq -\varepsilon\gamma V_\varepsilon, \end{aligned} \tag{2.49}$$

which may be relaxed by adding a relaxation, $\delta \in \mathbb{R}$, with relaxation weight $N \in \mathbb{R}$ and penalty factor $W \in \mathbb{R}$:

$$\begin{aligned} (\boldsymbol{\mu}^*, \delta^*) = \underset{(\boldsymbol{\mu},\delta)}{\operatorname{argmin}} \; & \boldsymbol{\mu}^T \boldsymbol{\mu} + W\delta^2 \\ s.t. \quad & \dot{V}_\varepsilon(\boldsymbol{\eta}, \boldsymbol{\mu}) \leq -\varepsilon\gamma V_\varepsilon + N\delta, \end{aligned} \tag{2.50}$$

for the given CLF constraint. With this formulation, it is of interest to incorporate input saturation bounds. First, we transform the problem to a input basis in terms of $\boldsymbol{u}$ using (2.38), and then add the physical input bounds $\boldsymbol{u}_{min}$ and $\boldsymbol{u}_{max}$:

$$\begin{aligned} (\boldsymbol{u}^*, \delta^*) = \underset{(\boldsymbol{u},\delta)}{\operatorname{argmin}} \; & \boldsymbol{u}^T \boldsymbol{\mathcal{A}}^T \boldsymbol{\mathcal{A}} \boldsymbol{u} + 2L_{\boldsymbol{f}}{}^T \boldsymbol{\mathcal{A}} \boldsymbol{u} + W\delta^2 \\ s.t. \quad & \boldsymbol{A}_{CLF} \boldsymbol{\mathcal{A}} \boldsymbol{u} - N\delta \leq \boldsymbol{b}_{CLF} - \boldsymbol{A}_{CLF} L_{\boldsymbol{f}} \\ & -\boldsymbol{u} \leq -\boldsymbol{u}_{min} \\ & \boldsymbol{u} \leq \boldsymbol{u}_{max}, \end{aligned} \tag{2.51}$$

where the output-space CLF constraint values are defined as:

$$\begin{aligned} \boldsymbol{A}_{CLF}(\boldsymbol{\eta}) &= L_{\boldsymbol{G}} V_\varepsilon(\boldsymbol{\eta}) \\ \boldsymbol{b}_{CLF}(\boldsymbol{\eta}) &= -\varepsilon\gamma V_\varepsilon(\boldsymbol{\eta}) - L_{\boldsymbol{F}} V_\varepsilon(\boldsymbol{\eta}). \end{aligned}$$

Note that the relaxation coefficient, N, is chosen to be zero, it will yield equivalently an unrelaxed CLF constraint on the convergence rate.

2.3 Human-Inspired Control Framework

This section provides a brief overview of the Human-Inspired Control Framework as a means for providing trajectories that yield provably stable and inherently robust dynamics walking when combined with the control techniques covered in Sec. 2.2.

2.3.1 Output Selection

A set of actual outputs for the model of this thesis, the 5-link point-foot walker, were also selected such as to provide a *full rank* representation of the human-walking, as discussed in [3, 72]. They are as follows:

1. The linearized slope of the nonstance leg:

$$\delta m_{nsl}(\boldsymbol{q}) := \frac{\partial}{\partial \boldsymbol{q}} \left(\frac{p^z_{nsf} - p^z_{hip}}{p^x_{nsf} - p^x_{hip}} \right) \Bigg|_{\boldsymbol{q}=0} \cdot \boldsymbol{q} \tag{2.52}$$

$$= -q_{sa} - q_{sk} - q_{sh} + q_{nsh} + \frac{L_c}{L_c + L_t} q_{nsk} \tag{2.53}$$

2. The stance knee angle:

$$\theta_{sk}(\boldsymbol{q}) := q_{sk}$$

3. The nonstance knee angle:

$$\theta_{nsk}(\boldsymbol{q}) := q_{nsk}$$

4. The angle of the torso with respect to the vertical:

$$\theta_{tor}(\boldsymbol{q}) := q_{sa} + q_{sk} + q_{sh}.$$

.

Placing these together yield the actual output vector:

$$\boldsymbol{h}^a(\boldsymbol{q}) = \begin{bmatrix} \delta m_{nsl}(\boldsymbol{q}) \\ \theta_{sk}(\boldsymbol{q}) \\ \theta_{nsk}(\boldsymbol{q}) \\ \theta_{tor}(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & 1 & \frac{L_c}{L_c+L_t} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{q}, \tag{2.54}$$

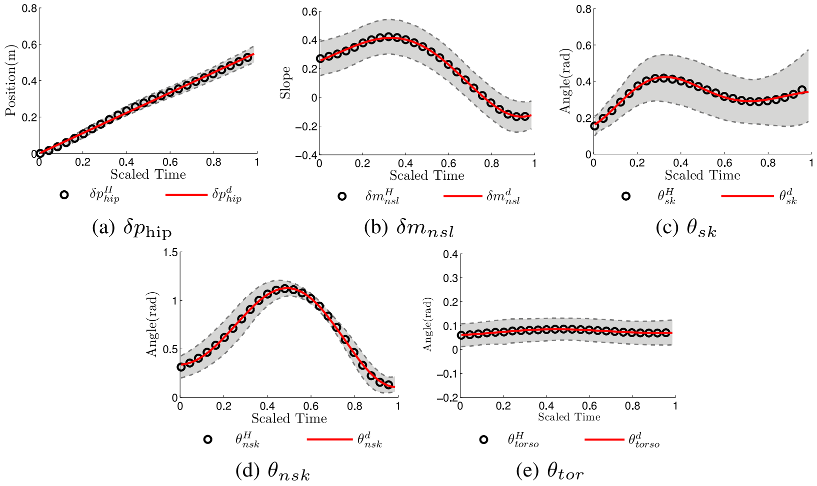
which is linear in terms of $\boldsymbol{q}$.

A successful attempt was made at finding a *canonical representation* of human walking data, results of which are used thoroughly in this lab and this thesis. Data collection was performed at the University of California, Berkley, where 9 subject's walking gaits were recorded using motion capture technology, and subsequently analyzed to yield joint angles [7]. Plots of the mean human data may be found in Fig. 2.4, and an image of the experiment is depicted in Fig. 2.5. This was first presented in [57], where the solution to a second-order spring-mass-damper solution was selected as the desired output, in the form:

$$h_{CWF}(t, \boldsymbol{b}) = e^{-b_4 t} \left(b_1 \cos(b_2 t) + b_3 \sin(b_2 t) \right) + b_5, , \tag{2.55}$$

where $\boldsymbol{b} \in \mathbb{R}^5$ is a set of parameters, where for the spring-mass-damper system, b_1 represents the initial position, b_2 the damped natural frequency, b_3 incorporates the initial velocity, b_4 is the decay defined by product of the damping factor and the natural frequency, and b_5 is the steady state "deflection" due to gravity.

To simplify the design of robust controllers, state-based desired outputs are desired. This goal is achieved by selecting a phasing variable that is monotonically



(a) δp_{hip} (b) δm_{nsl} (c) θ_{sk}

(d) θ_{nsk} (e) θ_{tor}

Figure 2.4: A plot of the mean human data from the experiments in [7], with dots representing the mean data points, the red line representing the data fitted with the canonical walking function, and the bounds showing one standard deviation from the mean. Graphs originally from [3].

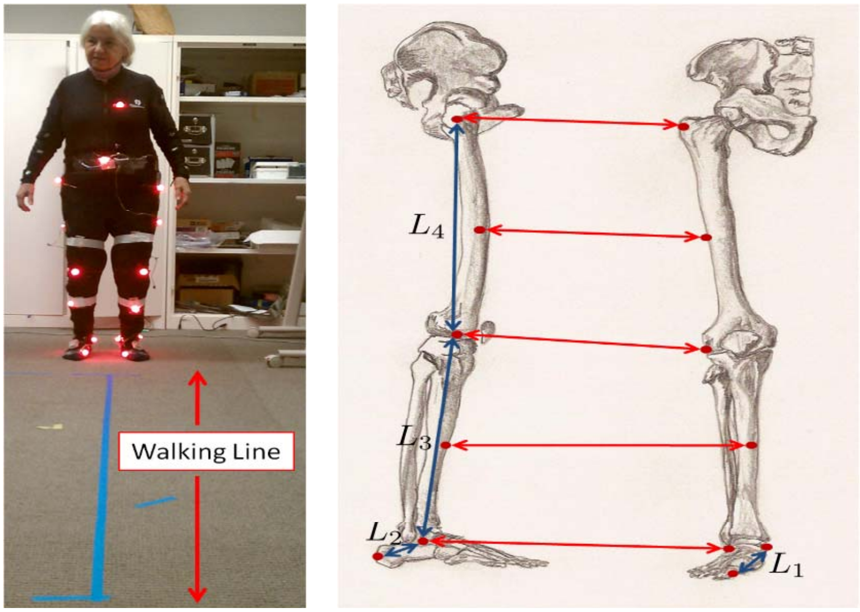


Figure 2.5: A snapshot of the motion capture experiment and a diagram depicting the measurements used to define the kinematics used to solve for the actual outputs during walking. Picture originally from [7].

increasing. In this case, the linearized x-position of the hip, $\delta p^x_{hip}(\boldsymbol{q}) = \boldsymbol{J}^x_{hip}\boldsymbol{q}$, where $\boldsymbol{J}^x_{hip} = \frac{\partial p^x_{hip}}{\partial \boldsymbol{q}}\big|_{\boldsymbol{q}=\boldsymbol{0}}$, is selected as a monotonically increasing variable, as seen in Fig. 2.4. The value is then set to zero by subtracting its initial value, and scaled to act as "time" by dividing by the desired hip velocity, v_d, yielding the function:

$$\tau(\boldsymbol{q}, v_d) = \frac{\delta p^x_{hip}(\boldsymbol{q}) - \delta p^x_{hip}(\boldsymbol{q}^+)}{v_d}. \tag{2.56}$$

This can then be substituted in for time to yield the state-based desired value:

$$\boldsymbol{h}^d(\boldsymbol{q}, \boldsymbol{\alpha}) = \begin{bmatrix} h_{CWF}(\tau(\boldsymbol{q}, v_d), \boldsymbol{\alpha}_{nsl}) \\ h_{CWF}(\tau(\boldsymbol{q}, v_d), \boldsymbol{\alpha}_{sk}) \\ h_{CWF}(\tau(\boldsymbol{q}, v_d), \boldsymbol{\alpha}_{nsk}) \\ h_{CWF}(\tau(\boldsymbol{q}, v_d), \boldsymbol{\alpha}_{tor}) \end{bmatrix}, \tag{2.57}$$

where the parameter set $\boldsymbol{\alpha}$ is defined as:

$$\boldsymbol{\alpha} = (v_d, \boldsymbol{\alpha}_{nsl}, \boldsymbol{\alpha}_{sk}, \boldsymbol{\alpha}_{nsk}, \boldsymbol{\alpha}_{tor}). \tag{2.58}$$

The next challenge in designing the desired trajectories is finding the parameters that yield provably robust, stable walking.

2.3.2 Human-Inspired Optimization

Given the formal specification of the control objective through virtual constraints, the human-inspired optimization [3] can be used to produce a parameter set and initial condition for provably stable, steady-state walking. The human-inspired optimization itself is based upon the virtual constraints optimization, described in [67] and also summarized in [38]. When the control objective is met such that $\boldsymbol{y} = \boldsymbol{0}_p$, the system is said to be on the zero-dynamics surface [67], formally defined as

$$\mathbf{Z}_{\boldsymbol{\alpha}} := \{\boldsymbol{x} : \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{\alpha}) = \mathbf{0}_p,\, L_{\boldsymbol{f}}\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{\alpha}) = \mathbf{0}_p\}. \tag{2.59}$$

With the feedback linearization controller in mind, once the control objective is satisfied in a single domain of stepping, the set of states under control is *forward invariant*, meaning that all states that start within this state remain in the state. Thus, for dynamic bipedal walking, the challenge is to find a walking gait such that the control objective is always satisfied, even after impact. This motivates the definition of *hybrid invariance* [67], where the *hybrid zero dynamics (HZD) condition* is defined as:

$$\mathbf{HZ}_{\boldsymbol{\alpha}} := \boldsymbol{\Delta}_R(\mathcal{S} \cap \mathbf{Z}_{\boldsymbol{\alpha}}) \subset \mathbf{Z}_{\boldsymbol{\alpha}}, \tag{HZD}$$

which implies the gait is designed such that control objective incorporates the post-impact behavior of the physical system, or, more formally, that the control objective is invariant through impact.

To give a brief overview of the optimization, the general problem formulation is

$$\begin{aligned} \boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, &\mathrm{Cost}(\boldsymbol{\alpha}) \\ &\mathbf{HZ}_{\boldsymbol{\alpha}} \\ &\boldsymbol{A}_{iq}^{phys}(\boldsymbol{\alpha}) \leq \mathbf{0} \end{aligned}$$

where $\mathrm{Cost}(\boldsymbol{\alpha})$ computes the 1-norm mechanical cost of transport, c_{mt}^1, for one step given the walking gait defined by $\boldsymbol{\alpha}$ (see discussion in Sec. 5.1), and $\boldsymbol{A}_{iq}^{phys}(\boldsymbol{\alpha})$ are nonlinear inequalities for physical constraints.

To implement the (HZD) constraint condition, the *approximate fixed point inverse* *kinematics* can be used to compute pre-impact state of the system [5]. The pre-impact coordinates are chosen due to the fact that the impact map involves a null-space, and thus the transformation is non-invertible. The conditions given are that:

1. the pre-impact guard is satisfied, $\boldsymbol{x}^- \in \mathcal{S}$, with the concrete condition, $h_R(\boldsymbol{q}^-) = 0$, and
2. the post-impact relabeled state satisfies the control objective defined by $\boldsymbol{\alpha}$, as $\boldsymbol{h}(\boldsymbol{\mathcal{R}}\boldsymbol{q}^-, \boldsymbol{\alpha}) = \mathbf{0}_p$.

This is defined in [3], with:

$$\boldsymbol{\vartheta}(\boldsymbol{\alpha}) = \boldsymbol{q}^- : \boldsymbol{h}(\boldsymbol{\mathcal{R}}\boldsymbol{q}^-, \boldsymbol{\alpha}) = \mathbf{0}_p,\ h_R(\boldsymbol{q}^-) = 0. \tag{2.60}$$

The corresponding velocities on the zero dynamics surface can be computed using the linear relationship:

$$\dot{\boldsymbol{\vartheta}}(\boldsymbol{\alpha}) = \begin{bmatrix} \boldsymbol{J}^x_{hip} \\ \boldsymbol{J} \end{bmatrix}^{-1} \begin{bmatrix} v_d \\ \mathbf{0}_p \end{bmatrix}, \tag{2.61}$$

where $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}(\boldsymbol{q})$ is the Jacobian of the output errors.

For compactness, define $\boldsymbol{x}^- = \boldsymbol{\vartheta}_{\boldsymbol{x}}(\alpha) = (\boldsymbol{\vartheta}(\alpha), \dot{\boldsymbol{\vartheta}}(\boldsymbol{\alpha}))^T$. Using the reduced dynamics surface, the time-to-impact, T_I, can be computed, and then an estimation of the solution, $\hat{\boldsymbol{x}}(t, \boldsymbol{\alpha})\ \forall\ t \in [0, T_I]$ can be computed using the propositions stated in [3]. To summarize these propositions, let zero dynamics of the system be defined, pulling notation and conventions from [52] and [67]. To concisely state the fixed point in terms of the full-order dynamics, we first define the *flow* of the dynamics as

$\boldsymbol{\varphi}_t(\boldsymbol{x}_0)$, with Poincaré return map, $\boldsymbol{\mathcal{P}}_{\boldsymbol{\varphi}}(\boldsymbol{x})$, defined as:

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{\varphi}}(\boldsymbol{x}) = \boldsymbol{\varphi}_{T_I(\boldsymbol{x})}(\boldsymbol{x}) \tag{2.62}$$

$$T_I(\boldsymbol{x}) = \inf\{t \geq 0 : \boldsymbol{\varphi}_t(\boldsymbol{x}) \in \mathcal{S}\}, \tag{2.63}$$

the fixed point, $\boldsymbol{x}^-$, is defined such that:

$$\boldsymbol{x}^- = \boldsymbol{\mathcal{P}}_{\boldsymbol{\varphi}}(\boldsymbol{x}^-). \tag{2.64}$$

Note that due to numerical approximations implementing the math in the following sections, $\boldsymbol{\vartheta}_{\boldsymbol{x}}(\boldsymbol{\alpha}) = \boldsymbol{x}^-$ may not be the precise fixed point.

With this approximate fixed point solution, the (HZD) constraint can be realized as:

$$\boldsymbol{y}(\boldsymbol{x}^-) = \mathbf{0}_p \tag{HZD Position}$$

$$\dot{\boldsymbol{y}}(\boldsymbol{\Delta}_R \boldsymbol{x}^-) = \mathbf{0}_p, \tag{HZD Velocity}$$

together with the fact that the pre-impact positions and velocities are satisfied by the inverse kinematics, enforce the hybrid invariance (HZD).

Assuming that the system state is on the zero dynamics surface, $\boldsymbol{\eta} = \mathbf{0}$, let $\boldsymbol{\xi}$ define the zero dynamics coordinates, the coordinates not directly controlled as a result of the transformation in (2.33). The dynamics of $\boldsymbol{\xi}$ are defined as:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}|_{\mathbf{Z}}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \boldsymbol{f}|_{\mathbf{Z}}(\mathbf{0}, \boldsymbol{\xi}). \tag{2.65}$$

In this case of a mechanical system with $n = 5$ degrees of freedom, $n_c = 0$ constraints, and $p = 4$ outputs, $\boldsymbol{\xi}$ is of dimension $2(n - n_c - p) = 2$, thus $\boldsymbol{\xi} \in \mathbb{R}^2$. Let the coordinates be chosen as in [3] which was motivated by [67]:

$$\xi_1(\boldsymbol{q}) = \delta p^x_{hip}(\boldsymbol{q}) = \boldsymbol{J}^x_{hip}\boldsymbol{q} \tag{2.66}$$

$$\xi_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\gamma}_0(\boldsymbol{q})\dot{\boldsymbol{q}} := \gamma(\boldsymbol{x}) \tag{2.67}$$

where $\boldsymbol{\gamma}_0(\boldsymbol{q}) = \boldsymbol{D}_{1,*}(\boldsymbol{q})$ is the first row of the inertia matrix corresponding to the dynamics of the non-actuated degree of freedom, q_{sa}. Note that this coordinate transformation, $\gamma(\boldsymbol{x})$, satisfying:

$$L_{\boldsymbol{g}}\gamma(\boldsymbol{x}) \equiv 0,$$

is favored in place of the more straightforward transformation $\xi_2 = \dot{\xi}_1$ in that the control input is required to define $\ddot{\xi}_1$, as stated in [67], which destroys the autonomous nature of these reduced coordinates. A more detailed explanation is reviewed in the appendix (Section D).

Given that $\xi_1 = \delta p^x_{hip}(\boldsymbol{q})$, we may rewrite τ as:

$$\tau(\xi_1, v_d) = \frac{\xi_1 - \xi_1^+}{v_d}, \tag{2.68}$$

where $\xi_1^+ = p^x_{hip}(\boldsymbol{q}^+)$. Thus we have the desired outputs,

$$\boldsymbol{y}^d(\xi_1) = \boldsymbol{h}^d(\tau(\xi_1, v_d), \boldsymbol{\alpha}). \tag{2.69}$$

To find the physical coordinates on the zero dynamics surface, $\boldsymbol{x}^r = \boldsymbol{x}|_{\mathbf{Z}}$, we perform the *hybrid zero dynamics reconstruction*, which first performs inverse kinematics parameterized by ξ_1 to solve for the reconstructed configuration, $\boldsymbol{q}^r$ and incorporates either the dynamics of ξ_2 or $\dot{\xi}_1$ to solve for the velocities $\dot{\boldsymbol{q}}^r$. The state of $\boldsymbol{x}^r$ allows one to easily find the coordinates that are on the zero dynamics surface satisfying $\boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{0}$.

In solving for the configuration, the full rank transformation is stated as:

$$\boldsymbol{q}^r = \boldsymbol{\Phi}_{\mathbf{Z}}(\xi_1) = \begin{bmatrix} \boldsymbol{J}^x_{hip} \\ \boldsymbol{J}^a \end{bmatrix}^{-1} \begin{bmatrix} \xi_1 \\ \boldsymbol{y}^d(\xi_1) \end{bmatrix}, \tag{2.70}$$

where $\boldsymbol{J}^a = \frac{\partial \boldsymbol{y}^a}{\partial \boldsymbol{q}}$ is a constant matrix due to the choice of outputs that are linear in $\boldsymbol{q}$. The velocity coordinates may be solved for one of two ways: first, using the zero dynamics coordinates as:

$$\dot{\boldsymbol{q}}^r = \boldsymbol{\Psi}_{\mathbf{Z}}(\xi_1)\xi_2 = \begin{bmatrix} \boldsymbol{\gamma}_0(\xi_1) \\ \boldsymbol{J}^a - \boldsymbol{J}^d(\xi_1) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \boldsymbol{0}_p \end{bmatrix} \xi_2, \tag{2.71}$$

where:

$$\boldsymbol{J}^d(\xi_1) = \frac{\partial \boldsymbol{y}^d(\xi_1)}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{y}^d(\xi_1)}{\partial \xi_1} \boldsymbol{J}^x_{hip} \in \mathbb{R}^{p \times 1}.$$

The other method, purely kinematic as a time derivative of $\boldsymbol{\Phi}_{\mathbf{Z}}(\xi_1)$, may also be used:[2]

$$\dot{\boldsymbol{q}}^{rk} = \dot{\boldsymbol{\Phi}}_{\mathbf{Z}}(\xi_1, \dot{\xi}_1) = \begin{bmatrix} \boldsymbol{J}^x_{hip} \\ \boldsymbol{J}^a \end{bmatrix}^{-1} \begin{bmatrix} \dot{\xi}_1 \\ \boldsymbol{J}^d(\xi_1)\dot{\xi}_1 \end{bmatrix}. \tag{2.72}$$

It must be noted that for both of the velocity reconstruction methods, the low-dimensional representation of the system's velocity, ξ_2 for $\dot{\boldsymbol{q}}^r(\xi_1, \xi_2) = \boldsymbol{\Psi}(\xi_1)\xi_2(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and the derivative $\dot{\xi}_1(\dot{\boldsymbol{q}})$ for $\dot{\boldsymbol{q}}^r(\xi_1, \dot{\xi}_1) = \dot{\boldsymbol{\Phi}}_{\mathbf{Z}}(\xi_1, \dot{\xi}_1)$, are needed.

A method presented introduced by [67] is the use of energy methods to produce

---

[2] This formulation was first employed by Matthew Powell in the implementation control methods for his Master's Thesis, and later by Ayonga Hereid on ATRIAS.

a relatively accurate approximation of the reduced coordinates. This method is also reviewed in more detail in the appendix (Section D). In this review, the physical constraints for a nontrivial underactuated walking orbit are introduced, with the primary values in this case being the reduced coordinate reset map, Δ_{ξ_2}, and the minimum momentum-like energy that defines that amount of energy that may be removed before the gait does not have enough energy to complete the state, $E_2^{\bar{*}\min}$, for the fixed point periodic orbit. The constraints are:

$$0 < \Delta_{\xi_2} < 1 \tag{2.73}$$

$$E_2^{\bar{*}\min} > 0. \tag{2.74}$$

With this nontrivial periodic orbit, the approximate solutions of the full order system, $\boldsymbol{x}^r(\xi_1, \boldsymbol{\alpha}) = (\boldsymbol{q}^r, \dot{\boldsymbol{q}}^r)^T$, may be computed as:

$$\boldsymbol{q}^r(\xi_1) = \boldsymbol{\Phi}_{\mathbf{Z}}(\xi_1) \tag{2.75}$$

$$\dot{\boldsymbol{q}}^r(\xi_1) = \boldsymbol{\Psi}_{\mathbf{Z}}(\xi_1)\hat{\xi}_2(\xi_1), \tag{2.76}$$

where $\xi_1 \in [\xi_1^+, \xi_1^-]$, and

$$\hat{\xi}_2(\xi_1) = \sqrt{2\hat{E}_2(\xi_1)}, \tag{2.77}$$

is the inverse of the coordinate transformation in (D.13) computed from the energy. With the reconstruction being an estimated solution, physical constraints can be placed on the system, such as joint limits, torque limits, and energy constraints.[3]

In the context of these constraints, define the pre-impact and post-impact depen-

---

[3] This construct comes from the implementation by Dr. Ames.

dent upon $\boldsymbol{\alpha}$ as:

$$\boldsymbol{x}^- = \boldsymbol{\vartheta}_{\boldsymbol{x}}(\boldsymbol{\alpha}) \tag{2.78}$$

$$\xi_1^- = \xi_1(\boldsymbol{x}^-) \tag{2.79}$$

$$\boldsymbol{x}^+ = \boldsymbol{\Delta}_R(\boldsymbol{x}^-) \tag{2.80}$$

$$\xi_1^+ = \xi_1(\boldsymbol{x}^+) \tag{2.81}$$

define the zero dynamics trajectory as $\mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}}$ using the reconstruction stated in (2.75):

$$\mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}} = \{\boldsymbol{x}^r(\xi_1, \boldsymbol{\alpha}) : \xi_1 \in [\xi_1^+(\boldsymbol{\alpha}), \xi_1^-(\boldsymbol{\alpha})]\}. \tag{2.82}$$

As shown in [3], there is a constraint on the foot height which uses the reconstruction. First, the normalized polynomial is defined, $f_{poly}(s, b)$, under the following constraints: (a) $f_{poly}(0, b) = f_{poly}(1, b) = 0$ and (b) $f_{poly}(\frac{1}{2}, b) = b$. This yields the simple logistic map polynomial:

$$f_{poly}(s, b) = 4bs(1 - s). \tag{2.83}$$

In this case, we normalize p_{nsf}^x on the interval $[{p_{nsf}^x}^+, {p_{nsf}^x}^-]$, and use the peak minimum foot height, h_{foot}, yielding the state-based polynomial:

$$\hat{f}_{poly}(\boldsymbol{q}, h_{foot}) = f_{poly}\left(\frac{p_{nsf}^x(\boldsymbol{q}) - {p_{nsf}^x}^+}{{p_{nsf}^x}^- - {p_{nsf}^x}^+}, h_{foot}\right), \tag{2.84}$$

where the goal is then to ensure that $h_R(\boldsymbol{q}) \geq \hat{f}_{poly}(\boldsymbol{q})$.

The physical constraints for each of the configurations are variants of the following:

$$L^{E_2^{\bar{*}\min}} \leq E_2^{\bar{*}\min} \leq U^{E_2^{\bar{*}\min}} \tag{2.85}$$

$$0 \leq \Delta_{\xi_2} \leq U^{\Delta_{\xi_2}} \tag{2.86}$$

$$\max_{\boldsymbol{x}^r \in \mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}}} \|\dot{\boldsymbol{q}}^r\|_\infty \leq U^{\dot{\boldsymbol{q}}} \text{ rad/s} \tag{2.87}$$

$$\hat{f}_{poly}(\boldsymbol{q}) - h_R(\boldsymbol{q}) \leq 0 \text{ m} \tag{2.88}$$

$$\max_{\boldsymbol{x}^r \in \mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}}} |\boldsymbol{u}_{\boldsymbol{\alpha},\varepsilon}(\boldsymbol{x}^r)| \leq U^{\boldsymbol{u}} \text{ Nm} \tag{2.89}$$

$$\min_{\boldsymbol{x}^r \in \mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}}} |\theta_{sk}(\boldsymbol{q}^r)| \geq L^{\theta_{sk}} \text{ rad} \tag{2.90}$$

$$\max_{\boldsymbol{x}^r \in \mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}}} |\theta_{sk}(\boldsymbol{q}^r)| \leq U^{\theta_{sk}} \text{ rad} \tag{2.91}$$

$$\max_{\boldsymbol{x}^r \in \mathcal{D}_{\mathbf{Z}_{\boldsymbol{\alpha}}}} |\theta_{tor}(\boldsymbol{q}^r)| \leq U^{\theta_{tor}} \text{ rad,} \tag{2.92}$$

where $L^{\square}$ and $U^{\square}$ represent the lower and upper bounds, respectively, of a given value, $\square$, and $\boldsymbol{u}_{\boldsymbol{\alpha},\varepsilon}(\boldsymbol{x}^r)$ represents the state-based controller torque for feedback linearization given the gait $\boldsymbol{\alpha}$ and control gain ε. Note that, for implementation, these constraints are appropriately incorporated into the vector-values function, $\boldsymbol{A}_{iq}^{phys}(\boldsymbol{\alpha})$. The parameters in Table 2.2 shows the parameters used to achieve the gait that yielded walking on hardware, in addition to the resulting parameters in Table 2.3.

Table 2.2: Optimization parameters for both configurations. A bound denoted by "·" indicates that the corresponding constraint is unused.

Configuration	h_{foot}	ε	$L^{E_2^{\bar{*}\min}}$	$U^{E_2^{\bar{*}\min}}$	$U^{\Delta_{\xi_2}}$	$U^{\dot{\boldsymbol{q}}}$	$U^{\boldsymbol{u}}$	$L^{\theta_{sk}}$	$U^{\theta_{sk}}$	$U^{\theta_{tor}}$
Light Torso	0.05	10	7	100	0.9	4	50	0.15	·	0.1
Heavy Torso	0.05	10	60	100	0.9	4	60	0.1	0.2	0.1

Table 2.3: Gait parameters generated from the human-inspired optimization.

$\boldsymbol{\alpha}_i$	b_1	b_2	b_3	b_4	b_5
Light Torso					
v_d	0.711	·	·	·	·
$\boldsymbol{\alpha}_{nsl}$	0.19	7.63	0.05	-1.08	0.04
$\boldsymbol{\alpha}_{sk}$	0.08	11.97	0.05	5.27	0.12
$\boldsymbol{\alpha}_{nsk}$	-0.31	-11.58	0.01	0.51	0.43
$\boldsymbol{\alpha}_{tor}$	-0.001	10.47	0.01	3.29	-0.09
Heavy Torso					
v_d	0.695	·	·	·	·
$\boldsymbol{\alpha}_{nsl}$	0.19	7.62	0.04	-0.85	0.04
$\boldsymbol{\alpha}_{sk}$	0.12	12.26	0.01	3.71	0.21
$\boldsymbol{\alpha}_{nsk}$	-0.28	-11.11	0.02	0.67	0.51
$\boldsymbol{\alpha}_{tor}$	-0.01	10.46	0.07	3.27	-0.03

3. HARDWARE CONTROLLERS

This chapter presents the basic math for the controllers implemented on hardware that yielded bipedal walking.

3.1 Position Control with the HZD Reconstruction

An elegant result of the Hybrid Zero Dynamics (HZD) reconstruction is the ability to use the actual states of the system, $\boldsymbol{x}^a$, to compute the desired states, $\boldsymbol{x}^d$, to (hopefully) be realized by the next control cycle. In an nutshell, the HZD reconstruction reduces the $2n = 10$ states to $n_{\mathbf{Z}} = 2$ coordinates for position and velocity, and the computes the full order state using the inverse kinematics states in (2.70) and either (2.71) or (2.72). These states can then be passed to local, decoupled proportional-derivative (PD) controllers.

First, define the diagonal matrices of proportional and derivative gains, $\boldsymbol{K}_P, \boldsymbol{K}_D \in \mathbb{R}^{m\times m}$, respectively, the effort for a generic input $\boldsymbol{\nu} \in \mathbb{R}^m$ that effects the actuated joints $\boldsymbol{q}_m$ may be stated as:

$$\boldsymbol{\nu} = \boldsymbol{K}_P(\boldsymbol{q}_m^a - \boldsymbol{q}_m^r(\hat{\xi}_1)) + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_m^a - \dot{\boldsymbol{q}}_m^{rk}(\hat{\xi}_1, \dot{\hat{\xi}}_1)), \tag{3.1}$$

noting that the kinematic desired velocities, $\dot{\boldsymbol{q}}^{rk}$, from (2.72) are used.

3.2 Position Control with the Ideal Model Resolved Motion Method

Although the HZD reconstruction is elegant and simple in its implementation, its properties as a PD control may yield to some issues in control, namely discontinuities in desired signals and the inability to easily incorporate velocity constraints. This section provides the novel contribution of this thesis, the *ideal model resolved motion method* (IM-RMM), which takes the actual state of the hardware, computes one time step of dynamics, integrates forward, and yields the desired positions and velocities in the next time instant. These values are then passed to a PD controller as the given objective.

To begin, the closed loop dynamics can be formed by incorporating the state-based control input $\boldsymbol{u}(\boldsymbol{x})$ into the vector field yielding the autonomous system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_{cl}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}). \tag{3.2}$$

Integration can then be performed using any valid method for explicitly solving an ordinary differential equation (ODE) given a fixed time step Δt. To provide a brief discrete perspective, given the current cycle k at time $t[k]$, the current state may be defined as $\boldsymbol{x}^a[k] = (\boldsymbol{q}^a[k], \dot{\boldsymbol{q}}^a[k])^T$, and the desired state that will be sent corresponding to this frame is defined as $\boldsymbol{x}^d[k] = (\boldsymbol{q}^d[k], \dot{\boldsymbol{q}}^d[k])^T$. With this notation, the flow of data can be defined as

$$\boldsymbol{x}^d[k] = \mathrm{ODE}(\boldsymbol{f}_{cl}(\cdot), \boldsymbol{x}^a[k], t[k], \Delta t),, \tag{3.3}$$

where $\mathrm{ODE}(\cdot)$ takes the state, $\boldsymbol{x}^a[k]$ at time $t[k]$ and integrates it forward a time step of Δt, yielding the desired state $\boldsymbol{x}^d[k]$ which is intended to be achieved at $t[k+1] = t[k] + \Delta t$, implying the intent that $\boldsymbol{x}^a[k+1] \rightarrow \boldsymbol{x}^d[k]$. If this objective is achieved, then this would impose the dynamics of $\boldsymbol{f}_{cl}(\boldsymbol{x}^a)$ on the physical system. These desired position can then be used in a PD controller in the form of (3.1).

The development process for going from hardware, to a hybrid system model, to the human-inspired framework, and ultimately implementing these behaviors on hardware, is shown in Fig. 3.1.

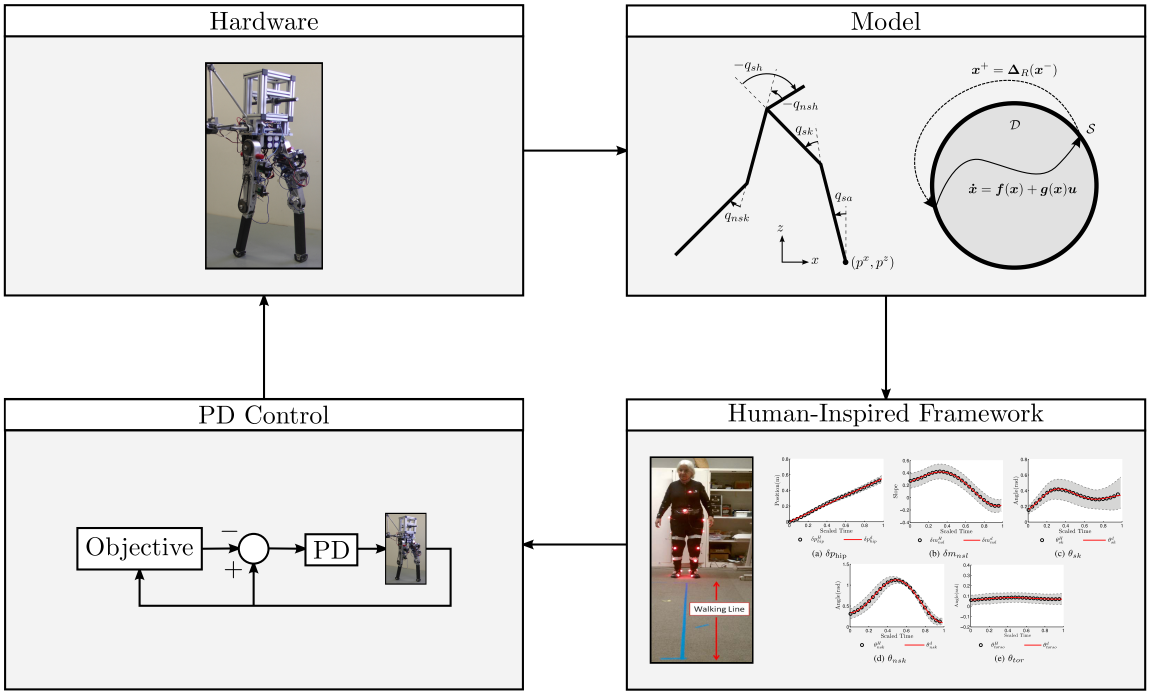


Figure 3.1: High-level diagram of process for realizing the theory.

4. IMPLEMENTATION

The controllers stated in Section 2 are first validated in simulation, and then transitioned to real-time controllers for controlling the hardware. This chapter introduces the techniques that were used to implement the simulation and optimization, then leads into the specifications for the low-level real-time control framework provided by SRI International and the high-level interfaces that were implemented. For maintaining both the simulation and optimization code, `git`[1] was used to facilitate distributed, non-invasive version control.

4.1 Simulation and Optimization

The simulation framework that provided the AMBER 1 models was effectively used in the context of this project [71], in order to provide provably stable walking gaits with sufficiently small errors. This framework combined prior work of Ryan Sinnet in the implementation of Dr. Murray's `Screws` software package[2] and initial MATLAB implementation of the simulation for hybrid systems, Dr. Aaron Ames's additions for the simulation, his implementation of the Human-Inspired Optimization, Shishir Yadukumar's modifications, and human walking data that were analyzed by Huihua Zhao and Shu Jiang. Another variant was also implemented using Roy Featherstone's `spatial_v2` library[3]. However, the optimization had not yet been implemented with this variant and it was deemed unnecessary to sacrifice additional effort to reproduce the same functionality.

The optimization was implemented by Dr. Aaron Ames using MATLAB, and modified by Shishir Yadukumar, Matthew Powell, and the author. The core mathe-

---

[1] `http://git-scm.com/`

[2] `http://www.cds.caltech.edu/~murray/books/MLS/software.html`

[3] `http://royfeatherstone.org/spatial/v2/index.html`

matical expressions were generated using Mathematica. Wolfram Mathematica was used in conjunction with the `Screws` package to produce the expressions for the kinematics, dynamics, and zero dynamics in MATLAB. The simulation then used these expressions to produce the vector fields for the dynamics of the system using the `ode45` variable step integrator with event detection to handle the detection of the switching surface. The HZD reconstruction, feedback linearization, and the CLF-QP were all implemented in the simulation.

4.2 Real-Time Control

This section provides additional descriptions of the hardware and software provided by SRI International to control DURUS, in addition to the software designed by the real-time software developed by the author in order to achieve bipedal walking.

There are a total of five 13-bit incremental encoders, with four placed at the output shaft of each motor, calibrated to incorporate the gear ratio and measure the angle and velocity of the joint j as q_j^H and $\dot{q}_j^H$. and one to measure the boom angle and velocity, q_{boom} and $\dot{q}_{boom}$. The hardware configuration itself, $\boldsymbol{q}^H \in \mathbb{R}^n$, is defined as:

$$\boldsymbol{q}^H = \begin{bmatrix} q_{lk}^H & q_{lh}^H & q_{rh}^H & q_{rk}^H & q_{tor}^H \end{bmatrix}^T, \tag{4.1}$$

with the convention that positive rotation is counterclockwise about axis along the motor shaft facing outward.

To map from hardware to stance coordinates, the hardware coordinates, $\boldsymbol{q}^H$, are first converted to coordinates with the left leg as stance, $\boldsymbol{q}^L$, and then relabeled as necessary to yield $\boldsymbol{q}$. This results in the linear transformation:

$$\boldsymbol{q}^L = \begin{bmatrix} -1 & -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{q}^H \tag{4.2}$$

$$\boldsymbol{q} = \begin{cases} \boldsymbol{q}^L, & \text{left leg is stance} \\ \boldsymbol{\mathcal{R}} \boldsymbol{q}^L, & \text{right leg is stance,} \end{cases} \tag{4.3}$$

which, due to its linearity, may also be applied to velocity and torque. Note that the stance ankle position and velocity in left stance coordinates are computed as:

$$q^L_{sa} = q^H_{boom} - q^L_{sk} - q^L_{sh} \tag{4.4}$$

$$\dot{q}^L_{sa} = \dot{q}^H_{boom} - \dot{q}^L_{sk} - \dot{q}^L_{sh}. \tag{4.5}$$

At each joint, there is a strain gage to measure the actual torque, u^a_j, experienced at the coupling between each joint's predecessor and successor link.

There are a total of five microcontrollers: one per joint, controlling the motors and processing sensor data from incremental encoders, absolute encoders, and load cell sensor data, and one in order to process encoder information detecting the rotation of the boom. The microcontrollers and sensors are powered by an off-board logic power supply, while the motors are powered by four on-board batteries, which provide a total of approximately 500 Wh of energy. The microcontrollers communicate with a real-time enabled Linux host computer running a real-time process. The real-time process, designed by SRI International, is implemented using the EtherLAB[4] software coupled with MATLAB Simulink Coder[5], and is set to run at 1 kHz.

The high-level controllers were implemented using C++ for the purpose of having complete control over the math and discrete transitions. The Eigen[6] library was used for linear algebra, and general care was taken to avoid unnecessary memory allocations for temporary working variables. The ROS catkin[7] build system was used due to its nature as a modular and tightly integrated build system supporting CMake, unittests, and `pkg-config`. The ROS parameter server[8] was used as a simple mechanism for providing atomic configuration for the controllers in order to easily capture experiment configurations every time data were recorded.

The symbolic expressions from Mathematica were also ported to Eigen C++ code and used to implement feedback linearization, the HZD reconstruction, and the CLF-QP. In order to realize the quadratic programs, a few small modifications were made to Ben Stephen's EQuadProg++[9] in order to better accommodate error tolerances and add additional error handling. This was selected in lieu of CVXGEN [39] for dynamic reconfiguration and for speed improvements using the Eigen library (which can perform 1 - 12 times as fast as CVXGEN with optimization enabled). For more information on these benchmarks, please see the Appendix.

IM-RMM was implemented using the ODEINT[10] template library which interfaced seamlessly with the Eigen data types. For both the input-output linearization and the CLF-QP cases, the control rates were fast enough on the host computer such that the fourth-order Runge-Kutta method (`runge_kutta4`) could be used while still

---

[4] `http://www.etherlab.org/`

[5] `http://www.mathworks.com/products/simulink-coder/`

[6] `http://eigen.tuxfamily.org/`

[7] `http://wiki.ros.org/catkin`

[8] `http://wiki.ros.org/rosparam`

[9] `http://www.cs.cmu.edu/~bstephe1/`

[10] `http://headmyshoulder.github.io/odeint-v2/`

running the control at 1 kHz. In the case of a more complex controller, the simpler Euler method (`euler`) could be employed.

In order to test the ideal controller implementations, the integration can be performed as

$$\boldsymbol{x}_d[k] = \mathrm{ODE}(\boldsymbol{f}_{cl}(\cdot), \boldsymbol{x}_d[k-1], t[k], \Delta t), \tag{4.6}$$

where, with k starting at 1, the initial value value $\boldsymbol{x}_d[0]$ is the fixed point of the gait provided by the optimization. This modification of (3.3) turns the controller into an open-loop playback of simulation trajectories. This can be used to assess the stability of the simulation implemented on hardware and heuristically check that the controller will run within the desired control rate.

To allow for controllers that may need to run at a slower rate, a separate thread is created whose control rate can be regulated at runtime. Since only position and velocity commands are considered in this paper, it was possible to interpolate these commands at the higher mid-level real-time rate, 1 kHz, to prevent step inputs from increasing the effort required by the position control loops. Care had to be taken, however, to minimize delay using multi-threading. The resulting high-level diagram of the position controller options, either the HZD reconstruction or the torque controllers through IM-RMM, are shown in Fig. 4.1.

With the three control variants, two different loop rates were used for the PD control on the hardware. For the HZD reconstruction, a 1 kHz PD controller producing current commands ran in the host's real-time process. This produced a more compliant command and minimized feedback oscillations due to the state-based τ. While there was substantial tracking error, robust and efficient walking was still achieved. When using IM-RMM, the tighter, the 10 kHz PD controller at the embedded level was used to achieve tighter tracking at the local position and velocity level and gave

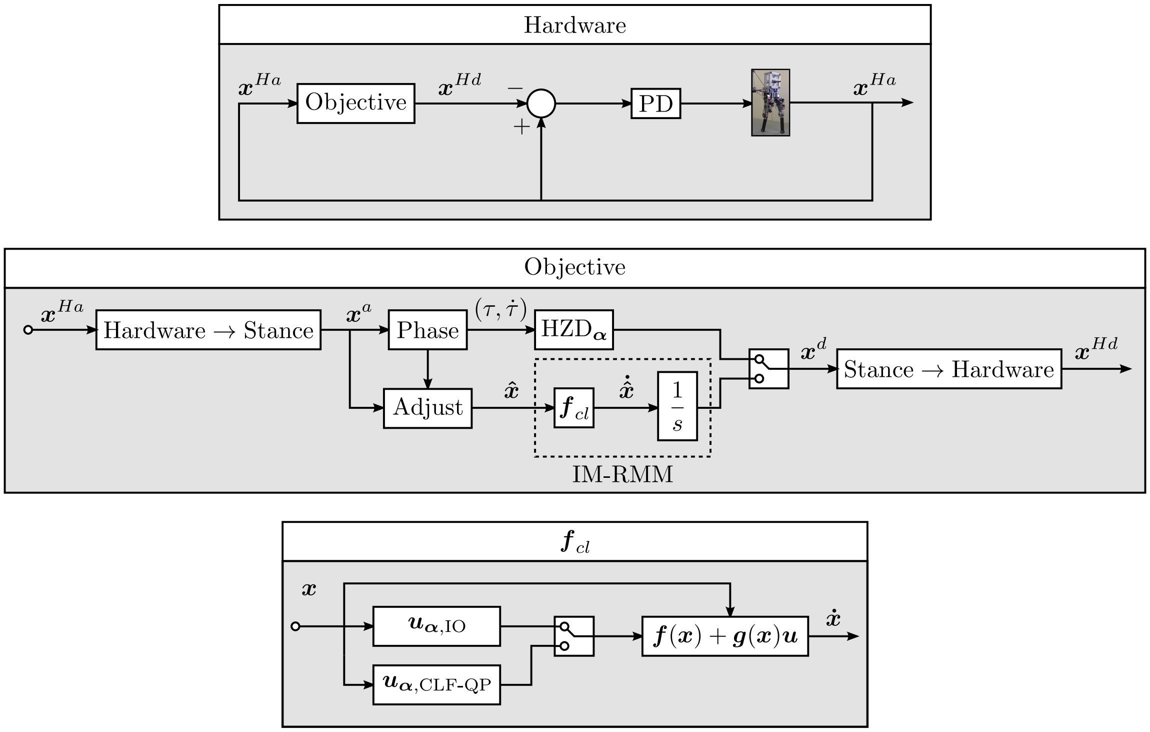


Figure 4.1: High-level diagram of the continuous time control.

the method more control authority.

4.2.1 Deviations from Theory

In order to achieve walking on the hardware, certain adjustments had to be made to the phase variables so that relatively smooth desired signals were produced. The phase variable, τ, was saturated between zero and its value at impact, τ^- as in [65]:

$$\tau = \begin{cases} 0, & \tau(\boldsymbol{q}) < 0 \\ \tau^-, & \tau(\boldsymbol{q}) > \tau^- \\ \tau(\boldsymbol{q}), & \text{otherwise.} \end{cases} \tag{4.7}$$

It was found that using the velocity data read from the system produced desired velocities that caused the system to oscillate. Two fixes were made: saturation, and then replacing with a polynomial fit as a function of τ.

The saturation was placed on $\dot{\tau}$ with a lower bound of 0 and upper bounded of its maximum value encountered in simulation, $\dot{\tau}^{\max}$:

$$\dot{\tau} = \begin{cases} 0, & \dot{\tau}(\dot{\boldsymbol{q}}) < 0 \\ \dot{\tau}^{\max}, & \dot{\tau}(\dot{\boldsymbol{q}}) > \dot{\tau}^{\max} \\ \dot{\tau}(\dot{\boldsymbol{q}}), & \text{otherwise.} \end{cases} \tag{4.8}$$

This saturation on $\dot{\tau}$ was also enforced as a rate limit for τ, enforcing that τ be monotonic and avoid excessive oscillations due to feedback between the actuation and the phasing variable.

The polynomial fit of $\dot{\tau}$ as a function of τ was made using a fifth-order polynomial of $\dot{\tau}$, with coefficients parameterized by $\boldsymbol{p}^{\tau}$, such that the final resulting value of $\dot{\tau}$ was computed after adjustments were made to τ itself:

$$\dot{\tau}_p(\tau) = \sum_{i=0}^{5} p^{\tau}_{6-i} \tau^{i}. \tag{4.9}$$

The value of these coefficients, and the corresponding limits, are shown in Table 4.1.

Table 4.1: Gait coefficients for light and heavy torso cases.

Configuration	τ^{-}	$\dot{\tau}^{\max}$	p^{τ}_1	p^{τ}_2	p^{τ}_3	p^{τ}_4	p^{τ}_5	p^{τ}_6
Heavy Torso	0.53	2.01	109	-167	77.6	3.40	-6.202	1.611
Light Torso	0.55	1.323	106.074	-179.979	96.496	-8.186	-4.071	1.113

It was also found that, if τ and $\dot{\tau}$ were very inconsistent, the system would encounter oscillations that significantly increased the cost of transport. Given the saturation of τ and the recalculation of $\dot{\tau}$, the values of ξ_1 and $\dot{\xi}_1$ were redefined as:

$$\hat{\xi}_1(\tau(\boldsymbol{q}^a)) = \xi_1^+ + v_d\tau \tag{4.10}$$

$$\dot{\hat{\xi}}_1(\dot{\tau}(\boldsymbol{q}^a, \dot{\boldsymbol{q}}^a)) = v_d\dot{\tau}, \tag{4.11}$$

such that they solve for the position of the hip from τ. The state of the system, $\boldsymbol{x}$, also had to be consistent to prevent additional feedback oscillations. To resolve this issue, the state was recomputed as $\hat{\boldsymbol{x}}$:

$$\hat{\boldsymbol{q}} = \begin{bmatrix} \multicolumn{2}{c}{\boldsymbol{J}_\tau} \\ \boldsymbol{0}_{4\times1} & \boldsymbol{I}_{4\times4} \end{bmatrix}^{-1} \begin{bmatrix} \tau - \frac{{p^x_{hip}}^+}{v_d} \\ \boldsymbol{q}^m \end{bmatrix} \tag{4.12}$$

$$\dot{\hat{\boldsymbol{q}}} = \begin{bmatrix} \multicolumn{2}{c}{\boldsymbol{J}_\tau} \\ \boldsymbol{0}_{4\times1} & \boldsymbol{I}_{4\times4} \end{bmatrix}^{-1} \begin{bmatrix} \dot{\tau} \\ \dot{\boldsymbol{q}}^m \end{bmatrix}, \tag{4.13}$$

which solves for $(q_{sa}, \dot{q}_{sa})$ in terms of $(\tau, \dot{\tau}, \boldsymbol{q}^m, \dot{\boldsymbol{q}}^m)$, where $\boldsymbol{J}_\tau = \frac{\partial\tau}{\partial\boldsymbol{q}}$.

5. RESULTS

This chapter first introduces the cost metrics used to measure progress in performing experiments, and then compares the resulting behaviors from simulation and experiments.

5.1 Cost Metrics

The *specific energetic cost of transport*,[1] c_{et}, is computed as in [12], which is total energy consumed over the product of weight and distance traveled, represented for step i as:

$$c_{et,i} = \frac{1}{mgd_i} \int_{t_i^+}^{t_i^-} P_{et}(t)dt \tag{5.1}$$

$$P_{et}(t) = P_{el} + P_{em}(t) \tag{5.2}$$

$$P_{em}(t) = \sum_{j=1}^{4} I_j(t)V_j(t), \tag{5.3}$$

where $P_{el} = 58.25\text{W}$ is the logic power consumed by the host computer (37.5W, measured using a Kill-a-Watt meter with the real-time process, the GUI, and data logging) and five microcontrollers (20.75W), and $I_j(t)$ and $V_j(t)$ are the currents and voltages recorded for the j^{th} motor, with the sum resulting in the electrical motor power, $P_{em}(t)$. The distance traveled is computed as

$$d_i = p_{nsf}^x(\boldsymbol{q}_i^-) - p_{nsf}^x(\boldsymbol{q}_i^+), \tag{5.4}$$

where $\boldsymbol{q}_i^+$ is the post-impact configuration at the beginning of the step, and $\boldsymbol{q}_i^-$ is the pre-impact configuration at the end of the step. In addition to the specific

---

[1] Also called the *total cost of transport* (TCOT) in [10].

costs of transport, the values of average amount of electrical power consumed by all of the motors, $\bar{P}_{em}$, and the ratio of negative (regenerated) W^{-}_{em} versus consumed positive (consumed) W^{+}_{em} motor electrical work, $r_{em} = W^{-}_{em}/W^{+}_{em}$, are computed and presented here.

Given that the load cell torque, u^{a}_{j}, and velocity, $\dot{\boldsymbol{q}}^{a}_{j}$ were measured for each actuated joint j, the cost of mechanical transport is computed as

$$c_{mt,i} = \int_{t_i^+}^{t_i^-} \sum_{j=1}^{n} P_{mt,j}(t)dt, \tag{5.5}$$

where $P_{mt,j}(t) = \Gamma_j \dot{q}_j$ is the nominal mechanical power required for the degree of freedom j, which may then be modified by the three options mentioned in [28]:

1. Total power, $c^{\pm}_{mt}$, where $P^{\pm}_{mt,j}(t) = \text{regen}_{C_m}(P_{mt,j}(t))$, where $\text{regen}_{C}(x) = [x]^{+} - C[-x]^{+}$ accounts for regeneration with loss, $[x]^{+}$ takes the value of x if it is positive or zero otherwise, and $C_m = 1$ is the coefficient of mechanical energy loss, used by [12];
2. Positive-only power, c^{+}_{mt}, where $P^{+}_{mt,j}(t) = [P_{mt,j}(t)]^{+}$, used by [28, 60]; and
3. The 1-norm of power, c^{1}_{mt}, where $P^{1}_{mt,j}(t) = |P_{mt,j}(t)|$, used by [38, 51].

The electrical cost, c_{et}, was chosen as the primary metric after discussion among SRI, the Dynamic Robotics Laboratory, and the AMBER Lab.

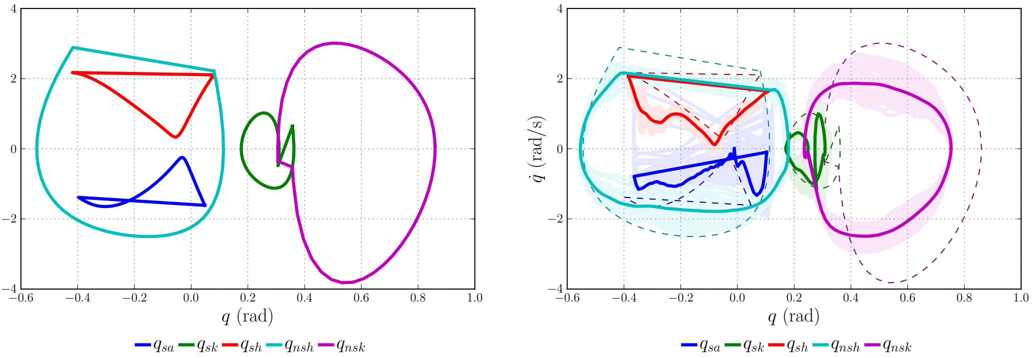
5.2 Experiments and Results

Each controller was run at 1 kHz, and when performing IM-RMM for these experiments, the fourth-order Runge-Kutta method was employed given feedback linearization ran in IM-RMM with an average run time of 60μs, while the CLF-QP ran with an average run time of 0.13ms. For each of the experiments, data were logged at 200 Hz using EtherLAB's TestManager interface and processed in MATLAB.

For each configuration, Light Torso and Heavy Torso, the resulting limit cycles are shown in Fig. 5.1 and Fig. 5.2, the local joint-level tracking is shown in Fig. 5.3, Fig. 5.4 Fig. 5.7, Fig. 5.8. The *nominal* output tracking comparison and *variations*, for the overall control objective, are shown in Fig. 5.5, Fig. 5.6, Fig. 5.9, and Fig. 5.10. Note that the nominal outputs in this case are the those defined by the zero dynamics surface, and are different the ultimate positions and velocities to be tracked by the IM-RMM desired positions and velocities. The variations are the errors with respect to the nominal outputs which are effectively replanned by the embedding of torque controllers via IM-RMM.

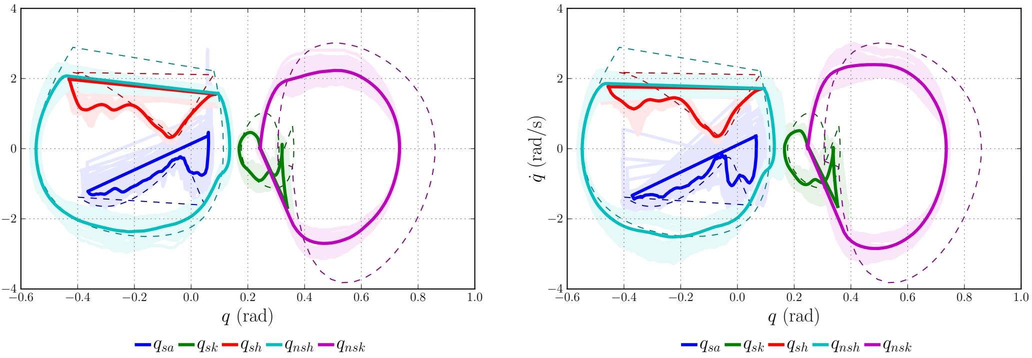
The worst joint-level tracking errors occur once again in the nonstance joints, and the joint-level tracking error for the stance knee is large at the beginning of the step due to the large error in the nonstance leg in the previous step, prior to the relabeling that occurs at impact. Note that the joint-level tracking errors for IM-RMM are substantially less than for the HZD reconstruction cases, due to the fact that IM-RMM produces desired positions and velocities that are only a small distance away from the actual state of the system, effectively replanning the desired trajectories which are much more easily achieved. The worst nominal output surface tracking occurred at the non-stance leg, for both δm_{nsl} and θ_{nsk}, while the best nominal output tracking occurred for the stance joints, θ_{sk} and θ_{tor}. This was due to the large travel experienced by the nonstance joints as a function of the phase variable, τ, and their reduced physical impedance.

The power consumption, torques, phase variables, and the values of V_ε and $\dot{V}_\varepsilon$ are included in the appendix in Section A. It may be noted that the torques are of similar magnitude, and τ, according to the monotonic constraints, evolved forward. Although the results are not shown here, no saturation nor rate limiting was necessary



(a) Simulation, IO

(b) Experiment, HZD



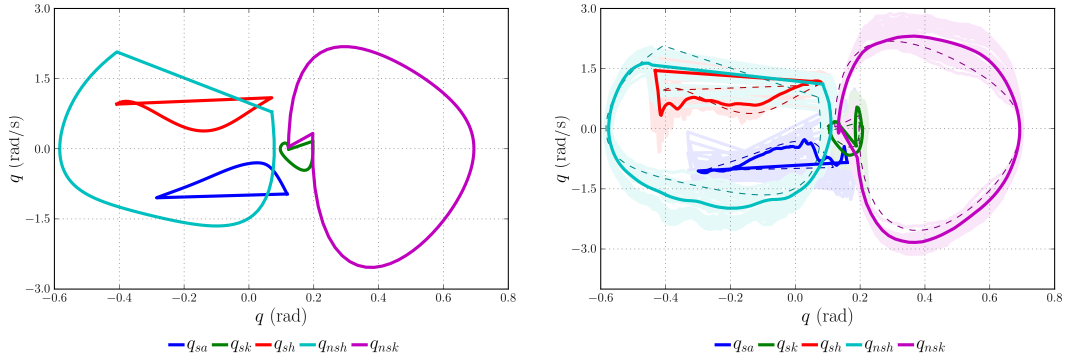
(c) Experiment, IO

(d) Experiment, CLF

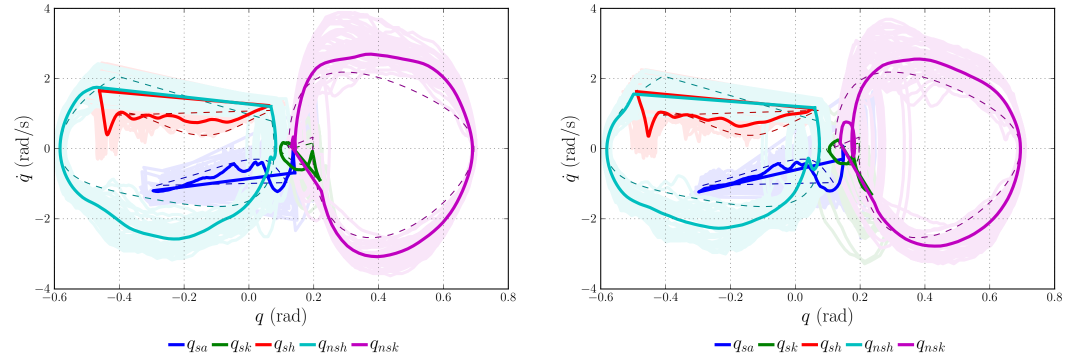
Figure 5.1: Comparison of limit cycles for the Light Torso configuration. The dotted limit cycles indicate an overlay of the feedback linearization cycle. The lighter solid lines represent the raw limit cycles, while the darker solid lines represent the average limit cycles.

for τ when using IM-RMM. The resulting cost of transport numbers for each method and configuration are shown in Table 5.1.

An Analysis of Variance (ANOVA) test [23] was performed on the data with a 95% confidence interval. The costs of transport were computed for each step (where the values reported in Table 5.1 are the mean) and the variable of interest was the walking method: either the HZD reconstruction, IO via IM-RMM, or CLF-QP via IM-RMM. The results of ANOVA are displayed in Table 5.2. As seen, all p-



(a) Simulation, IO (b) Experiment, HZD

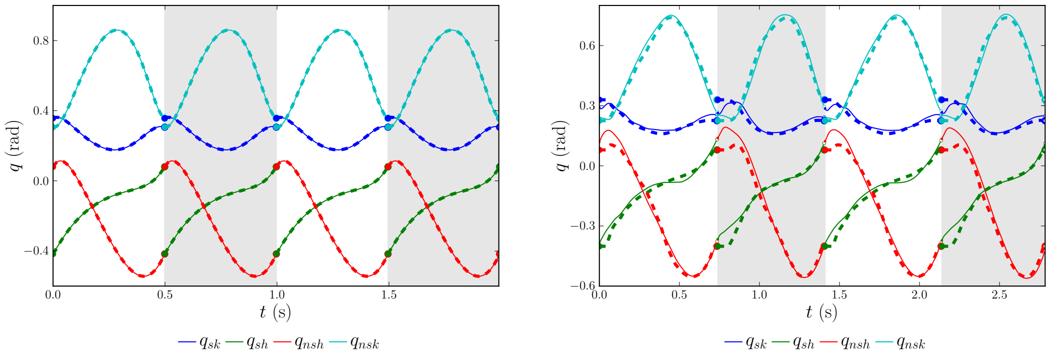


(c) Experiment, IO (d) Experiment, CLF

Figure 5.2: Comparison of limit cycles for the Heavy Torso configuration. The dotted limit cycles indicate an overlay of the feedback linearization cycle. The lighter solid lines represent the raw limit cycles, while the darker solid lines represent the average limit cycles.

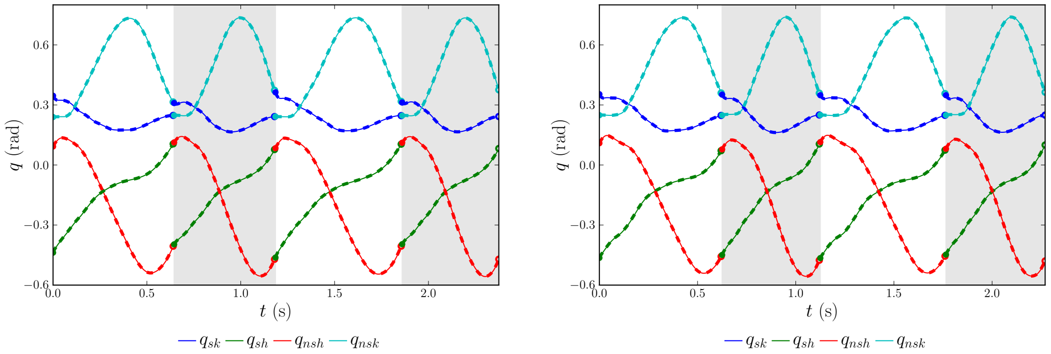
values indicate that the null-hypothesis (that the data sets have similar means) is rejected, thus implying that there is a *statistically significant difference* in the costs of transport for each of the walking methods for the experiments presented.

An important note with IM-RMM is that the controllers applied more effort than in the open-loop ideal model because the ideal dynamics were clearly not achieved on hardware using the IM-RMM. Additionally, the gains had to be raised according to the selected control rate, Δt. For $\Delta t = 0.001$, the IO controller walked with



(a) Simulation, IO

(b) Experiment, HZD



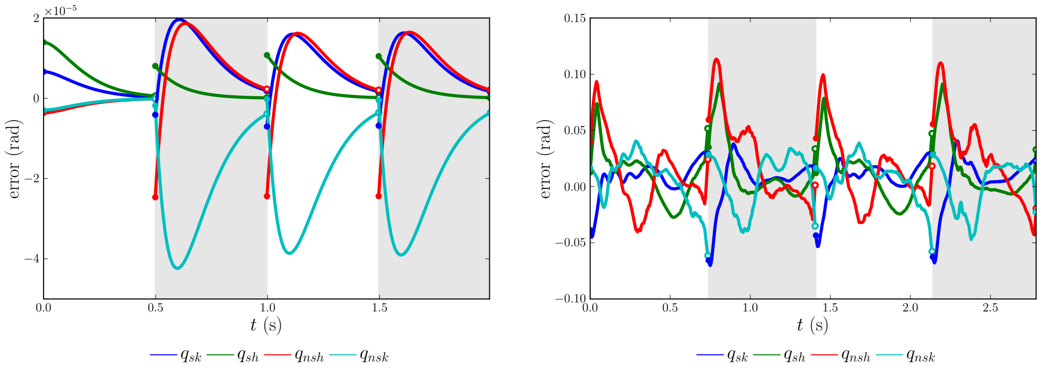
(c) Experiment, IO

(d) Experiment, CLF

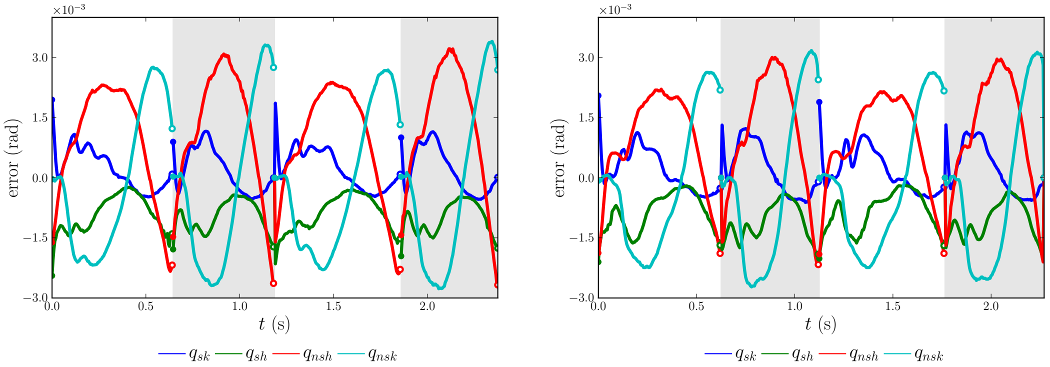
Figure 5.3: Comparison of position tracking for the Light Torso configuration where the solid lines are the actual positions and the dotted lines are the desired positions. The regions shaded with gray represent steps where the right leg is stance, and the unshaded regions represent where the left leg is stance.

minimal oscillation with $\varepsilon = 80$, and the unrelaxed, unbounded CLF-QP walked with minimal oscillation with $\varepsilon = 120$. When increasing $\Delta t = 0.002$, the IO gain had to be decreased to $\varepsilon = 50$, while the c_{et} increased for the walking, possibly due to increased phase delay for the feedback.

The walking itself shown in Figure 5.11. For the phase variables, it may be noted that IM-RMM did not require any saturation of the phase variable itself in order to achieve walking. The record for longest duration and distance traveled for walking



(a) Simulation, IO (b) Experiment, HZD

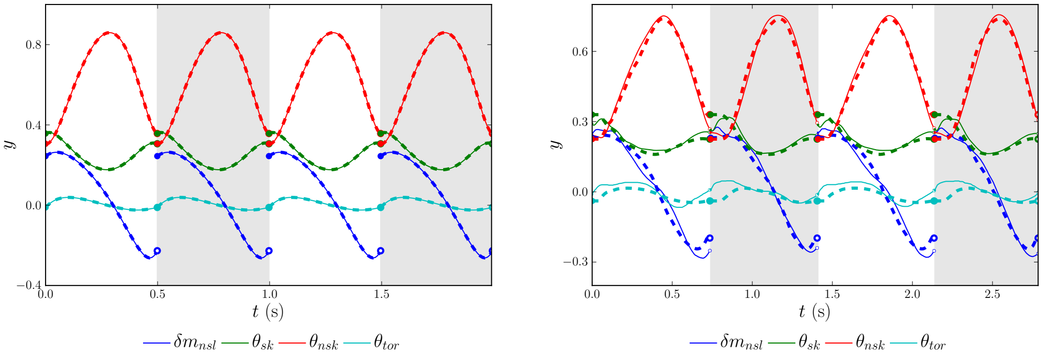


(c) Experiment, IO (d) Experiment, CLF

Figure 5.4: Comparison of position errors for the Light Torso configuration.

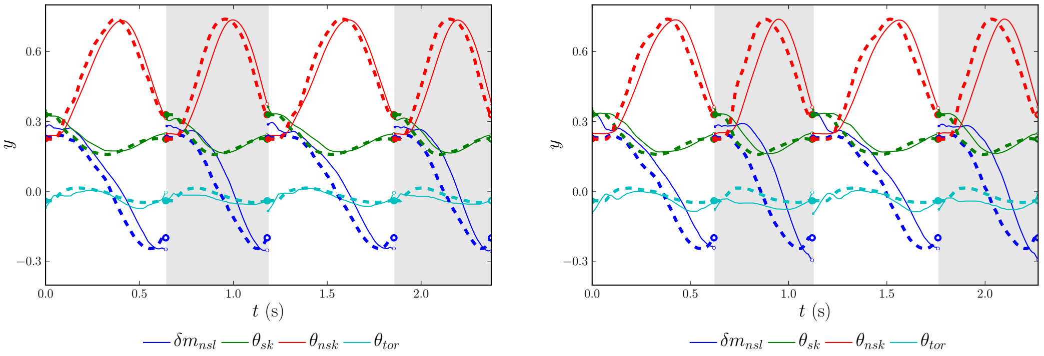
was using the CLF-QP method via IM-RMM for Heavy Torso configuration, lasting 2 hours and 53 minutes and covering approximately 7 km (425 laps). The walking duration for this experiment was too long to record, so the lap count was estimated by diving the total duration by the average time to complete one lap, 24.4 s. The experiment only stopped due to low voltages on one cell of the motor power batteries.

In comparing to literature, Table 5.3 shows the corresponding Cost of Transport numbers.



(a) Simulation, IO

(b) Experiment, HZD



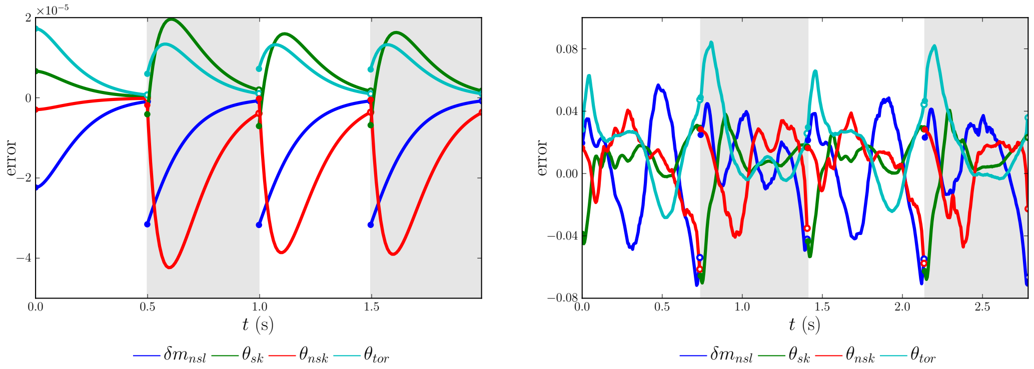
(c) Experiment, IO

(d) Experiment, CLF

Figure 5.5: Comparison of actual outputs versus nominal desired outputs for the Light Torso configuration. The solid lines are actual values, while the dashed lines are desired values.

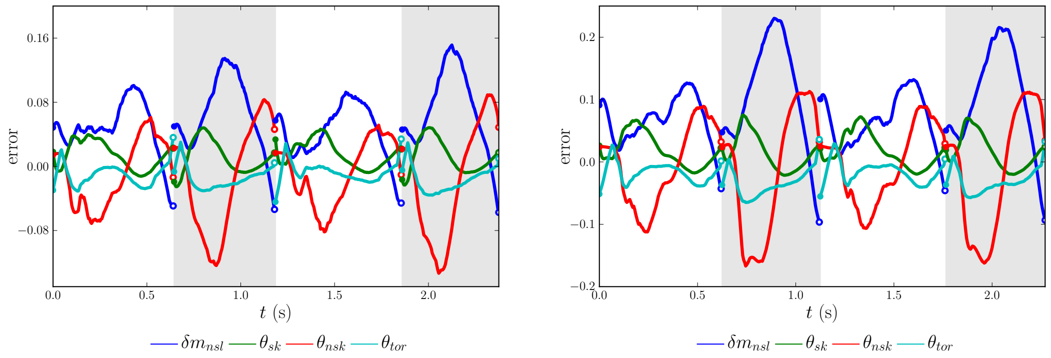
Table 5.1: Resulting gait performance, including costs of transport and power consumption.

Method	c_{et}	c^1_{mt}	c^+_{mt}	$c^\pm_{mt}$	$\bar{P}_{em}$(W)	r_{em}	$\bar{v}_{\text{hip}}$ (m/s)	n_{step}	$\bar{t}_{step}$(s)
Light Torso									
HZD	1.1	0.19	0.1	0.02	64.56	0.07	0.53	65	0.78
IO	0.86	0.25	0.12	-0.01	64.03	0.14	0.66	66	0.6
CLF-QP	0.85	0.27	0.14	0.02	71.91	0.13	0.74	64	0.58
Heavy Torso									
HZD	0.73	0.14	0.05	-0.05	71.77	0.06	0.6	64	0.75
IO	0.68	0.18	0.08	-0.03	83.38	0.15	0.7	64	0.63
CLF-QP	0.63	0.18	0.08	-0.03	67.72	0.13	0.68	63	0.68



(a) Simulation, IO

(b) Experiment, HZD



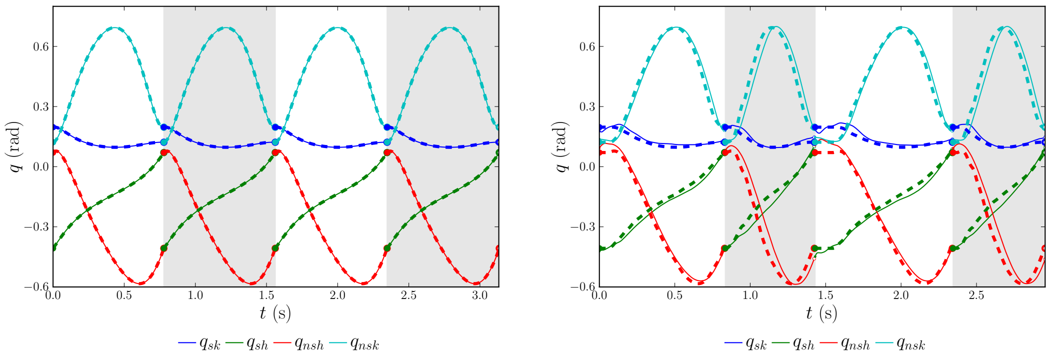
(c) Experiment, IO

(d) Experiment, CLF

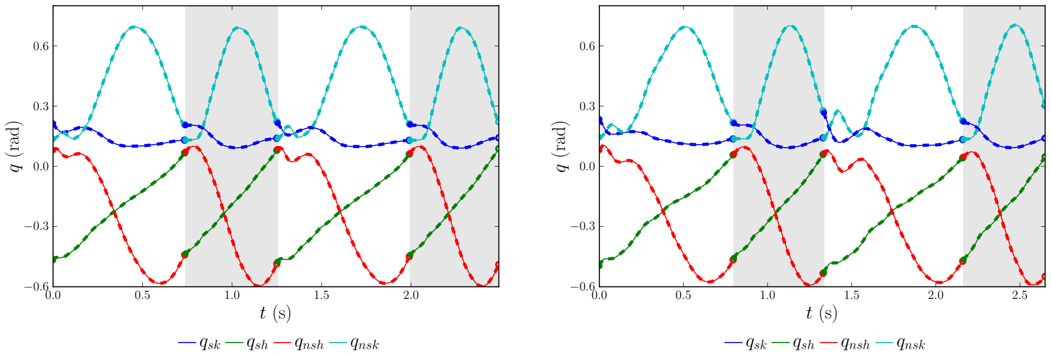
Figure 5.6: Comparison of variations in the nominal outputs for the Light Torso configuration. Note that the values of the joint-level tracking errors in Fig. 5.4 are substantially less for IM-RMM.

Table 5.2: Null hypothesis values ($p > 0.05$ that data sets are similar) resulting from ANOVA comparison of the three controllers for the costs of transport for 60 steps.

	p			
Method	c_{et}	c_{mt}^{1}	c_{mt}^{+}	$c_{mt}^{\pm}$
Light Torso	$3.17 \cdot 10^{-63}$	$7.39 \cdot 10^{-12}$	$4.24 \cdot 10^{-17}$	$2.07 \cdot 10^{-09}$
Heavy Torso	$1.62 \cdot 10^{-13}$	$7.38 \cdot 10^{-05}$	$2.18 \cdot 10^{-18}$	$1.52 \cdot 10^{-05}$

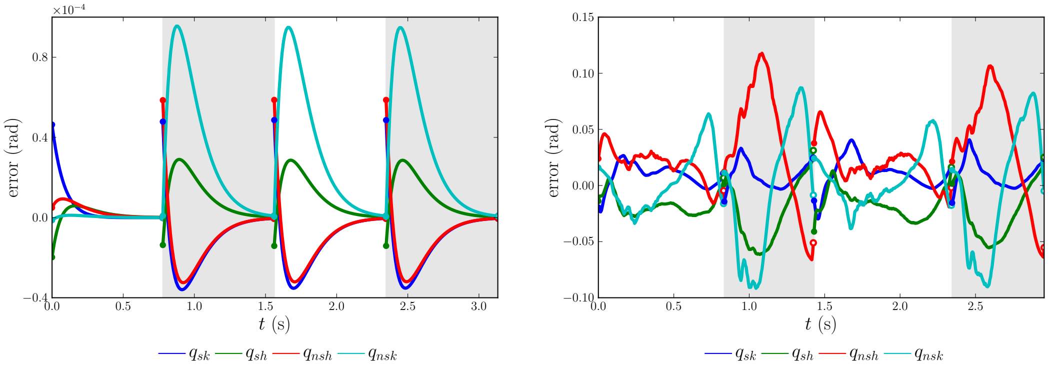


(a) Simulation, IO (b) Experiment, HZD



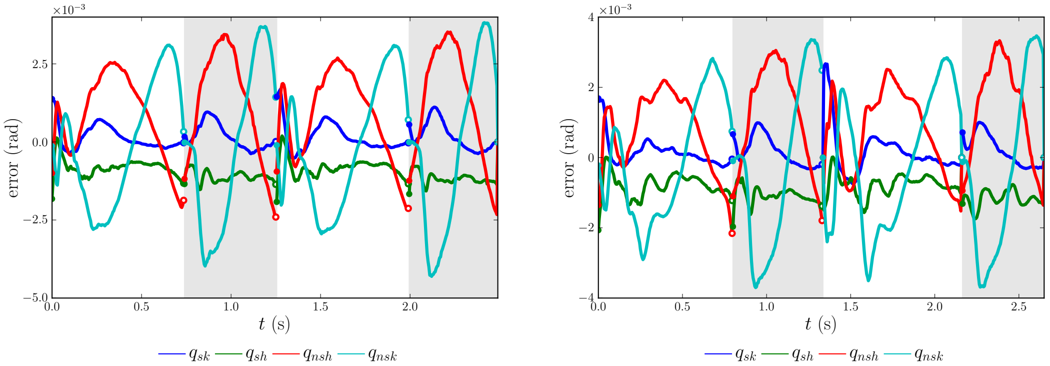
(c) Experiment, IO (d) Experiment, CLF

Figure 5.7: Comparison of position tracking for the Heavy Torso configuration where the solid lines are the actual positions and the dotted lines are the desired positions. The regions shaded with gray represent steps where the right leg is stance, and the unshaded regions represent where the left leg is stance.



(a) Simulation, IO

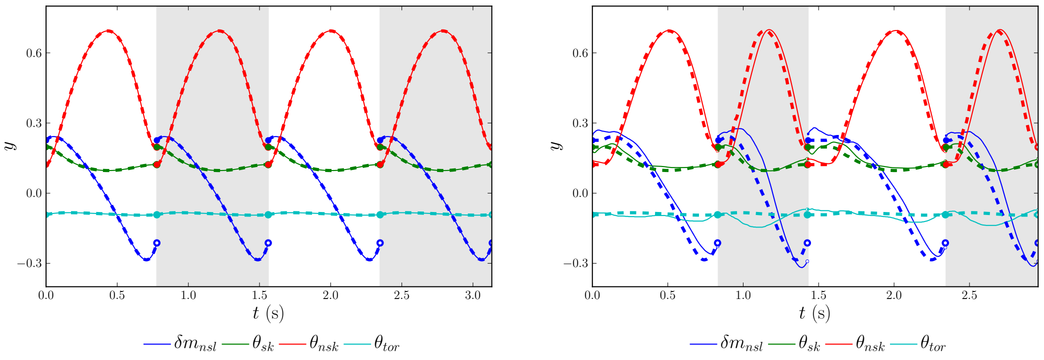
(b) Experiment, HZD



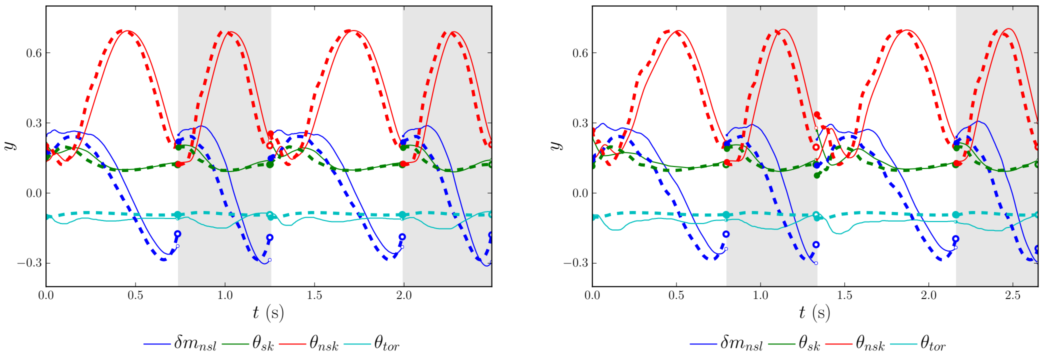
(c) Experiment, IO

(d) Experiment, CLF

Figure 5.8: Comparison of position errors for the Heavy Torso configuration.

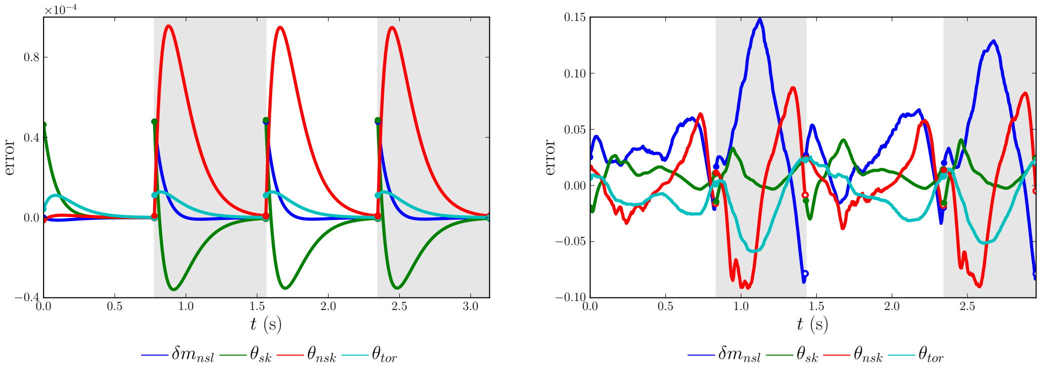


(a) Simulation, IO (b) Experiment, HZD



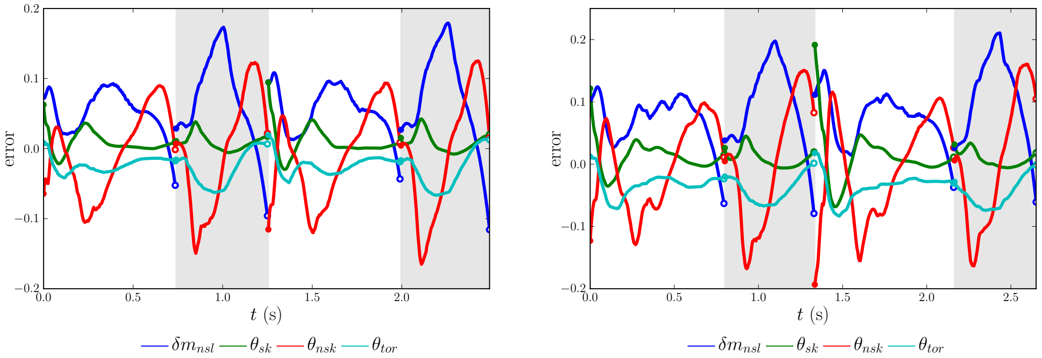
(c) Experiment, IO (d) Experiment, CLF

Figure 5.9: Comparison of actual outputs versus nominal desired outputs for the Heavy Torso configuration. The solid lines are actual values, while the dashed lines are desired values.



(a) Simulation, IO

(b) Experiment, HZD



(c) Experiment, IO

(d) Experiment, CLF

Figure 5.10: Comparison of variations in the nominal outputs for the Heavy Torso configuration. Note that the values of the joint-level tracking errors in Fig. 5.8 are substantially less for IM-RMM.

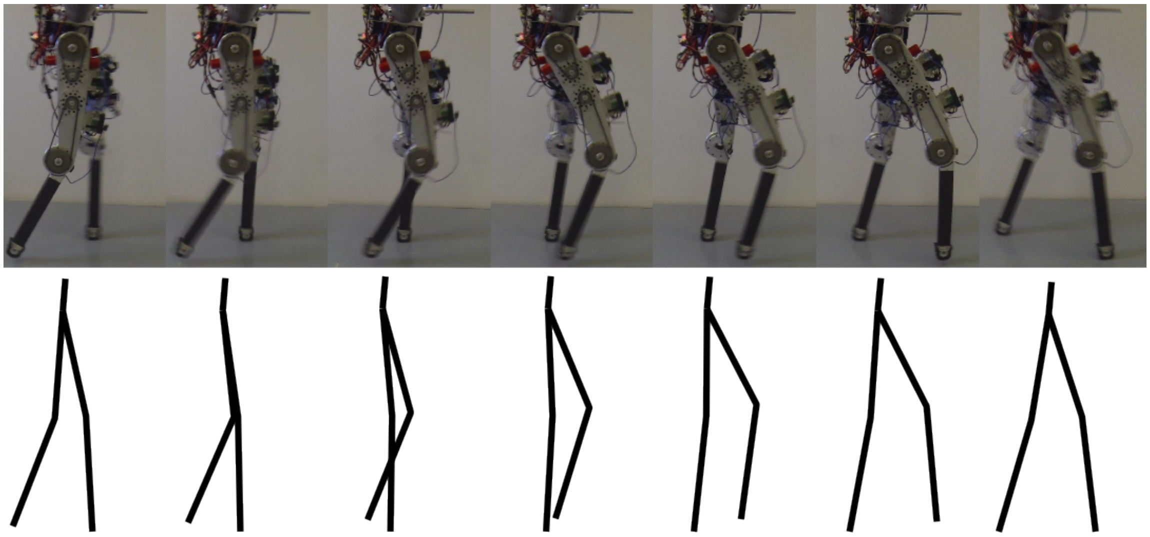


Figure 5.11: Walking tiles of the behavior on hardware and in simulation for feedback linearization (IO). Mirrored to so that walking occurs left to right.

Table 5.3: Cost of Transport Metrics for various Robots.

Name	m (kg)	v (m/s)	L_{leg} (m)	c_{et}	c^{1}_{mt}	c^{+}_{mt}	$c^{\pm}_{mt}$
Human [12]	·	·	·	0.2	·	0.05	·
Cornell Biped [12]	13	0.4	·	0.2	·	0.055	·
Cornell Ranger [10]	9.9	·	·	0.19	·	·	·
Spring Flamingo [51, 60]	14.2	1.25	0.9	0.93 [a]	0.07	·	·
RABBIT [45, 66, 60]	32	1.2	0.8	·	·	0.38	·
MABEL [45, 60]	60	1.5	1	·	·	0.15	·
ERNIE [38]	21.5	0.75	0.76	·	0.34	·	·
AMBER 1 [73]	3.3	0.44	0.6	1.88	·	·	·
ATRIAS [26]	62	0.85	·	1.13	·	·	·
ASIMO [12]	52	0.44	·	3.23	·	1.6	·
ATLAS [10]	102	2	·	5	·	·	·
DURUS	31.5	0.68	0.89	0.63	0.18	0.08	-0.03

---

[a] Estimated by multiplying motor COT by ratio of power measured at wall outlet vs. energy consumed by motor: $\frac{200\text{ W}}{15\text{ W}}$.

6. CONCLUSION

This thesis completes the objective of developing IM-RMM and obtaining bipedal underactuated walking using IM-RMM as a method to test ideal torque controllers and achieve better cost metrics than using the HZD reconstruction. Using two different physical configurations of the hardware, walking was achieved using a total of three different techniques, with the lowest specific cost of electrical transport being 0.63. The controllers were used to achieve walking for a record of 2 hours and 53 minutes using the CLF-QP controller implemented via IM-RMM, covering 425 laps yielding approximately 6.7 km covered.

6.1 Future Directions

The challenges that now lay ahead are to minimize the zero dynamics surface tracking error, implement the actual torque controllers to compare performance, refine the control using advanced MPC techniques [49], and extend these results first to compliant systems [41, 19] that may more gracefully handle the impacts and next to more complex, 3D humanoid robots.

Though lofty, the implementation of this and future work could be improved through integration with the `drake` framework[1]; better utilization of MATLAB Simulink Coder; integration with the IHMC Yobotics[2], Gazebo[3], and V-Rep[4] simulators; and incorporating additional design components from the ATRIAS control framework [46] and other real-time enabled software frameworks [13].

---

[1] `http://drake.mit.edu`

[2] `http://www.ihmc.us/groups/scs/`

[3] `http://gazebosim.org/`

[4] `http://www.coppeliarobotics.com/`

REFERENCES

[1] C. T. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi. Survey of robust control for rigid robots. *Control Systems Magazine*, 11(2):pp. 24–30, 1991.

[2] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. Nonlinear predictive control and moving horizon estimationan introductory overview. In *Advances in Control*, pages 391–449. Springer, 1999.

[3] A. Ames. Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control*, 59(5):1115–1130, May 2014.

[4] A. Ames, K. Galloway, K. Sreenath, and J. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, April 2014.

[5] A. D. Ames, E. A. Cousineau, and M. J. Powell. Dynamically stable bipedal robotic walking with nao via human-inspired hybrid zero dynamics. In T. Dang and I. Mitchell, editors, *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*, pages 135–144, New York, NY, USA, 2012. ACM.

[6] A. D. Ames and M. Powell. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. In D. C. Tarraf, editor, *Control of Cyber-Physical Systems*, volume 449 of *Lecture Notes in Control and Information Sciences*, pages 219–240. Springer International Publishing, 2013.

[7] A. D. Ames, R. Vasudevan, and R. Bajcsy. Human-data based cost of bipedal robotic walking. In E. Frazzoli and R. Grosu, editors, *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, HSCC

’11, pages 153–162, New York, NY, USA, 2011. ACM.

[8] S. Au, M. Berniker, and H. Herr. Powered ankle-foot prosthesis to assist level-ground and stair-descent gaits. *Neural Networks*, 21(4):654 – 666, 2008. Robotics and Neuroscience.

[9] C. Azevedo, P. Poignet, and B. Espiau. Artificial locomotion control: from human to robots. *Robotics and Autonomous Systems*, 47(4):203–223, 2004.

[10] P. A. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. G. D. Karssen, C. Paul, and A. Ruina. Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge. *The International Journal of Robotics Research*, 33(10):1305–1321, 2014.

[11] B. G. Buss, A. Ramezani, K. A. Hamed, B. A. Griffin, K. S. Galloway, and J. W. Grizzle. Preliminary walking experiments with underactuated 3d bipedal robot marlo. *IROS*, 2014. (submitted).

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

[13] N. Dantam, D. M. Lofaro, A. Hereid, P. Oh, A. Ames, and M. Stilman. Multi-process communication and control software for humanoid robots. *IEEE RAS Robotics and Automation Magazine*, 2015. (accepted).

[14] M. F. Fallon, M. Antone, N. Roy, and S. Teller. Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing. *Humanoids*, 2014. (to appear).

[15] R. Featherstone. *Rigid body dynamics algorithms.* Springer, 2008.

[16] G. Garofalo, C. Ott, and A. Albu-Schaffer. Walking control of fully actuated robots based on the bipedal slip model. In L. Parker, editor, *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1456–1463. IEEE, 2012.

[17] J. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath. Mabel, a new

robotic bipedal walker and runner. In *American Control Conference*, pages 2030–2036. IEEE, 2009.

[18] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames. Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica*, 50(8):1955–1988, 2014.

[19] A. Hereid, C. Hubicki, E. Cousineau, J. Hurst, and A. D. Ames. Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking. *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. (under review).

[20] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames. Dynamic multi-domain bipedal walking with atrias through slip based human-inspired control. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control (HSCC)*, pages 263–272, New York, NY, USA, 2014. ACM Press.

[21] M. Hirose and Y. Haikawa. Development of humanoid robot asimo. In *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, October 2001.

[22] J. K. Hodgins and M. H. Raibert. Biped gymnastics. *The International Journal of Robotics Research*, 9(2):115–128, 1990.

[23] R. V. Hogg and J. Ledolter. *Engineering statistics*, volume 358. MacMillan New York, 1987.

[24] J. Horn, J. Reher, H. Zhao, V. Paredes, and A. D. Ames. Ampro: Translating robotic locomotion to a powered transfemoral prosthesis. *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. (under review).

[25] S. S. Hsu. Motion control. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 133–159. Springer, 2008.

[26] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Sproewitz, A. Abate, and

J. Hurst. Atrias: Enabling agile biped locomotion with a template-driven approach to robot design. *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. (under review).

[27] M. Hutter, C. D. Remy, M. Hopflinger, and R. Siegwart. Slip running with an articulated robotic leg. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 4934–4939, 2010.

[28] R. Jafari, L. L. Flynn, A. Hellum, and R. Mukherjee. Energy-conserving gaits for point-foot planar bipeds: A five-dof case study. In *ASME Dynamic Systems and Control Conference*, page V001T10A001, 2013.

[29] S. Kajita and B. Espiau. Legged robots. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 361–389. Springer, 2008.

[30] K. Kaneko, S. Kajita, F. Kanehiro, K. Yokoi, K. Fujiwara, H. Hirukawa, T. Kawasaki, M. Hirata, and T. Isozumi. Design of advanced leg module for humanoid robotics project of meti. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 38–45 vol.1, 2002.

[31] T. Kato, A. Takanishi, H. Jishikawa, and I. Kato. The realization of the quasi-dynamic walking by the biped walking machine. In *Fourth Symposium on Theory and Practice of Walking Robots*, pages 341–351, 1981.

[32] H. Khalil. *Nonlinear systems*. Prentice Hall PTR, 2002.

[33] J. Kim, H. Kwak, H. Lee, K. Seo, B. Lim, M. Lee, J. Lee, and K. Roh. Balancing control of a biped robot. In *Systems, Man, and Cybernetics (SMC), IEEE International Conference on*, pages 2756–2761, Oct 2012.

[34] C. Kuo and S.-P. T. Wang. Nonlinear robust industrial robot control. *Journal of dynamic systems, measurement, and control*, 111(1):24–30, 1989.

[35] J. Lack. Planar multicontact locomotion using hybrid zero dynamics. Master's thesis, Texas A&M University, 2013.

[36] Z. Li, B. Vanderborght, N. G. Tsagarakis, and D. G. Caldwell. Fast bipedal walk using large strides by modulating hip posture and toe-heel motion. In *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, pages 13–18, 2010.

[37] W.-L. Ma, H.-H. Zhao, S. Kolathaya, and A. D. Ames. Human-inspired walking via unified pd and impedance control. *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[38] A. E. Martin, D. C. Post, and J. P. Schmiedeler. Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet. *The International Journal of Robotics Research*, 33(7):988–1005, 2014.

[39] J. Mattingley and S. Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

[40] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[41] B. Morris and J. Grizzle. Hybrid invariance in bipedal robots with series compliant actuators. In *Decision and Control, IEEE Conference on*, pages 4793–4800, 2006.

[42] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[43] G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, and M. Raibert. Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30(4):372–377, 2012.

[44] C. Ott, M. A. Roa, and G. Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *Humanoid Robots (Humanoids),*

IEEE-RAS International Conference on, pages 26–33, 2011.

[45] H.-W. Park, K. Sreenath, J. W. Hurst, and J. W. Grizzle. Identification of a bipedal robot with a compliant drivetrain. *Control Systems, IEEE*, 31(2):63–88, 2011.

[46] A. Peekema, D. Renjewski, and J. Hurst. Open-source real-time robot operation and control system for highly dynamic, modular machines. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V07AT10A063–V07AT10A063, 2013.

[47] F. Pfeiffer, K. Loffler, and M. Gienger. The concept of jogging johnnie. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3129–3135, 2002.

[48] M. Powell. Robot locomotion controller generation through human inspired optimization. Master's thesis, Texas A&M University, 2013.

[49] M. Powell, E. Cousineau, and A. D. Ames. Model predictive control of underactuated bipedal robotic walking. *International Conference on Robotics and Automation*, 2015. (under review).

[50] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. *The International Journal of Robotics Research*, 31(10):1117–1133, 2012.

[51] J. E. Pratt. Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots. Technical report, DTIC Document, 2000.

[52] S. Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer New York, 1999.

[53] A. Schepelmann, H. Geyer, and M. Taylor. Development of a testbed for robotic

neuromuscular controllers. In N. Roy, P. Newman, and S. Srinivasa, editors, *Robotics: Science and Systems*, 2012.

[54] L. Sentis and O. Khatib. Control of free-floating humanoid robots through task prioritization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1718–1723. IEEE, 2005.

[55] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2008.

[56] R. Sinnet. Hybrid geometric feedback control of three-dimensional bipedal robotic walkers with knees and feet. Master's thesis, Texas A&M University, 2011.

[57] R. W. Sinnet, M. J. Powell, R. P. Shah, and A. D. Ames. A human-inspired hybrid control approach to bipedal robotic walking. In T. Başar and R. Tempo, editors, *18th IFAC World Congress*, pages 6904–11, 2011.

[58] J. Smith, D. Stephen, A. Lesman, and J. Pratt. Real-time control of humanoid robots using openjdk. In *Proceedings of the 12th International Workshop on Java Technologies for Real-time and Embedded Systems*, page 29. ACM, 2014.

[59] M. Spong. *Robot dynamics and control*. Wiley India Pvt. Limited, 2008.

[60] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[61] B. J. Stephens and C. G. Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 52–59, 2010.

[62] N. G. Tsagarakis, S. Morfey, G. Medrano Cerda, L. Zhibin, and D. G. Caldwell. Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. In L. Parker, editor, *IEEE International Conference on Robotics and Automation (ICRA)*, pages 673–678, 2013.

[63] L. Villani and J. De Schutter. Force control. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 161–185. Springer, 2008.

[64] C. J. Walsh, K. Endo, and H. Herr. A quasi-passive leg exoskeleton for load-carrying augmentation. *International Journal of Humanoid Robotics*, 04(03):487–506, 2007.

[65] E. R. Westervelt. *Toward a coherent framework for the control of planar biped locomotion*. PhD thesis, University of Michigan, 2003.

[66] E. R. Westervelt, G. Buche, and J. W. Grizzle. Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds. *The International Journal of Robotics Research*, 23(6):559–582, 2004.

[67] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.

[68] E. C. Whitman. *Coordination of Multiple Dynamic Programming Policies for Control of Bipedal Walking*. PhD thesis, Carnegie Mellon University, 2013.

[69] Wikipedia. Fukushima daiichi nuclear disaster. Web, 2014. `http://en.wikipedia.org/wiki/Fukushima_Daiichi_nuclear_disaster`. Accessed October 4, 2014.

[70] M. Wisse, A. L. Schwab, and F. C. van der Helm. Passive dynamic walking model with upper body. *Robotica*, 22(06):681–688, 2004.

[71] S. Yadukumar. Bipedal robotic walking on flat-ground, up-slope and rough terrain with human-inspired hybrid zero dynamics. Master's thesis, Texas A&M University, 2012.

[72] S. N. Yadukumar, M. Pasupuleti, and A. D. Ames. Human-inspired underactuated bipedal robotic walking with amber on flat-ground, up-slope and uneven terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Sys-*

tems (IROS), pages 2478–2483, 2012.

[73] S. N. Yadukumar, M. Pasupuleti, and A. D. Ames. From formal methods to algorithmic implementation of human inspired control on bipedal robots. In E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, editors, *Algorithmic Foundations of Robotics X*, pages 511–526. Springer, 2013.

[74] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 368–374, 1999.

[75] H. Zhao, W. Ma, M. B. Zeagler, and A. D. Ames. Human-inspired multi-contact locomotion with amber2. In *International Conference on Cyber-Physical Systems*, 2014. (submitted).

APPENDIX A

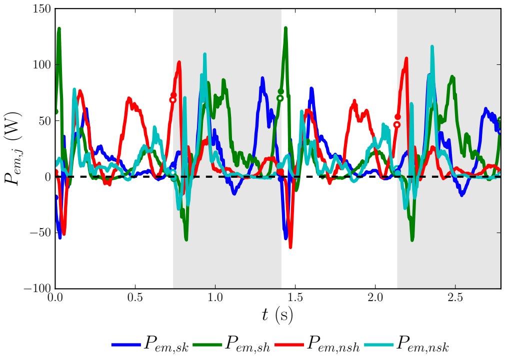
ADDITIONAL RESULTS

For both configurations, Light Torso and Heavy Torso, plots are shown for electrical motor power consumption, per joint and overall, in Fig. A.1, Fig. A.2 and Fig. A.8, Fig. A.9. As a means of reference to the mean, the electrical costs of transport for individual step shown in these plots are shown in Table A.1.

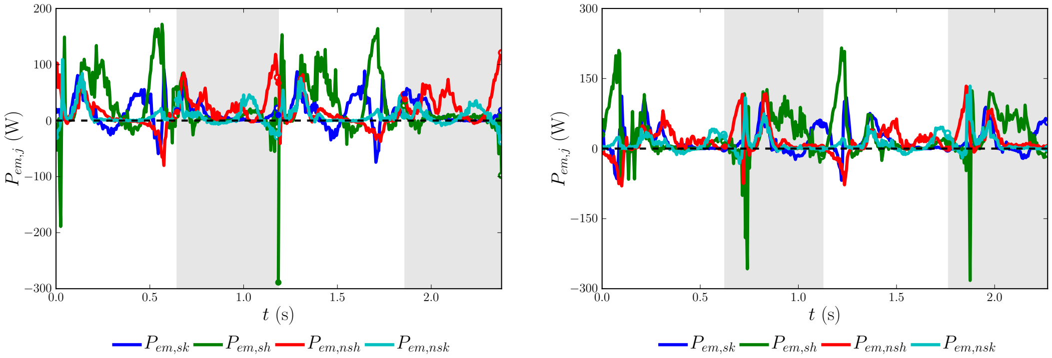
In addition, plots are shown for torques in Fig. A.3 and Fig. A.10, phase variables in Fig. A.4 and Fig. A.11, and Lyapunov function values, V_ε and $\dot{V}_\varepsilon$, in Fig. A.5 and Fig. A.12.

Table A.1: The average cost of electrical transport for a given trial, in addition to the individual costs of transport for each step shown in the plots presented.

Method	$\bar{c}_{et}$	$c_{et,i}$			
Light Torso					
HZD	1.1	1.06	1.1	1.04	1.2
IO	0.86	0.87	0.85	0.88	0.83
CLF-QP	0.85	0.82	0.8	0.87	0.82
Heavy Torso					
HZD	0.73	0.74	0.69	0.81	0.69
IO	0.68	0.6	0.67	0.63	0.64
CLF-QP	0.63	0.62	0.58	0.65	0.59



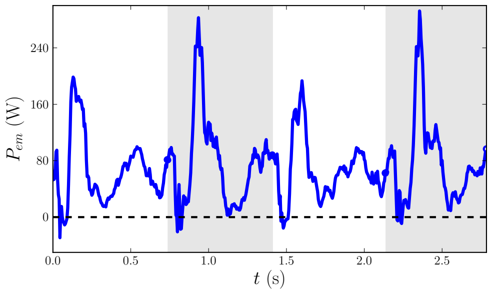
(a) Experiment, HZD



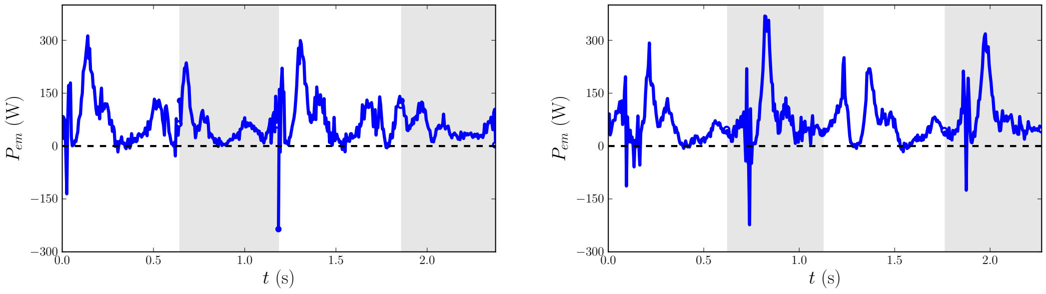
(b) Experiment, IO

(c) Experiment, CLF

Figure A.1: Comparison of electrical power for individual motors, $P_{em,j}$, for the Light Torso configuration.



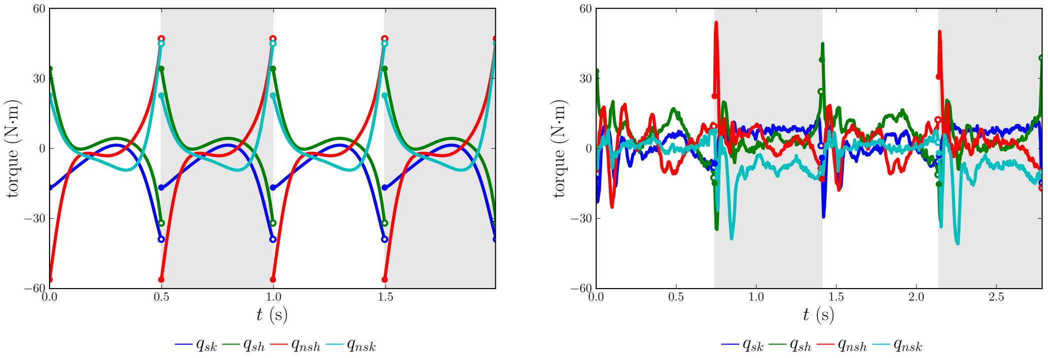
(a) Experiment, HZD



(b) Experiment, IO

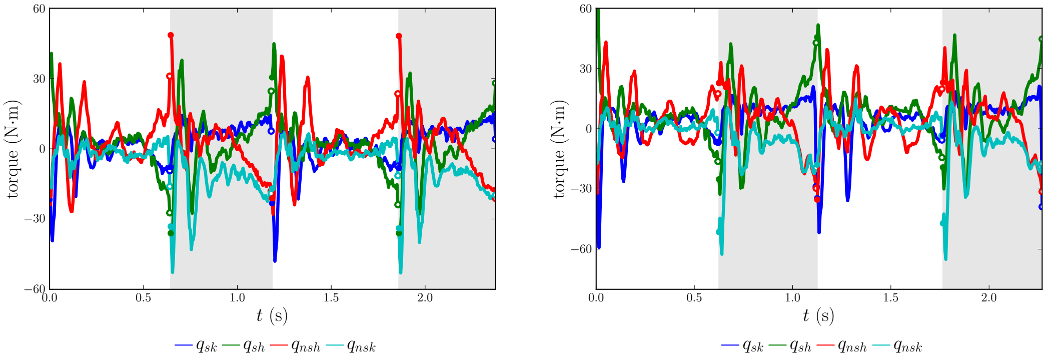
(c) Experiment, CLF

Figure A.2: Comparison of electrical power for all motors, P_{em}, for the Light Torso configuration.



(a) Simulation, IO

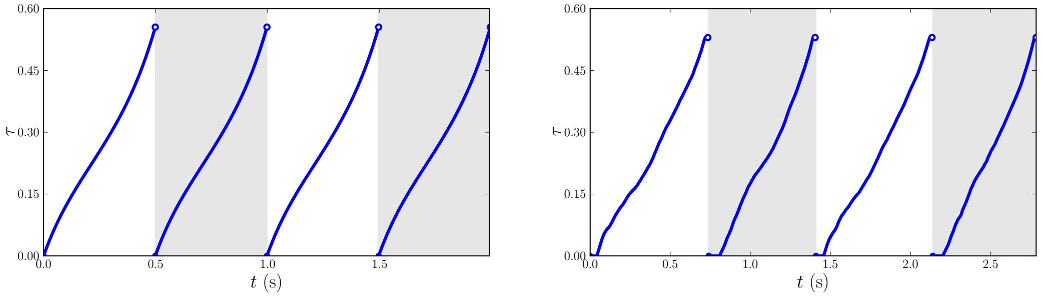
(b) Experiment, HZD



(c) Experiment, IO

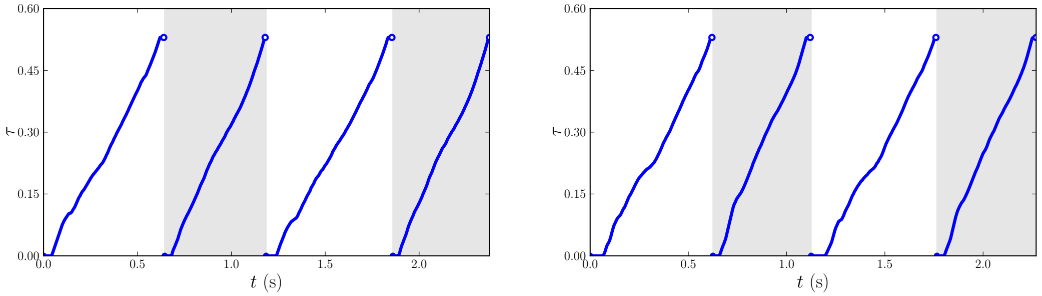
(d) Experiment, CLF

Figure A.3: Comparison of torques measured on the system for the Light Torso configuration.



(a) Simulation, IO

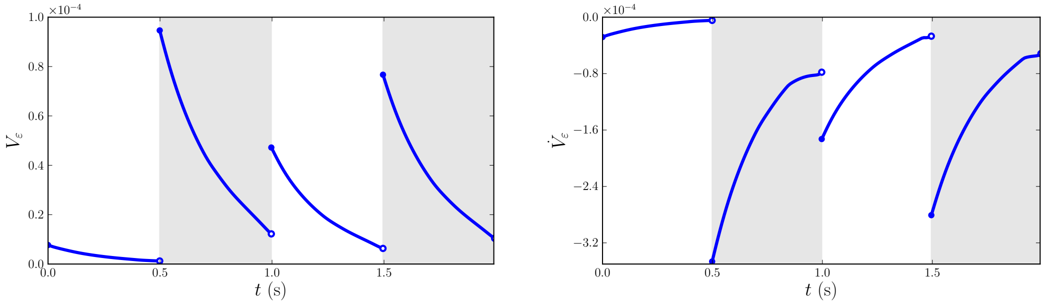
(b) Experiment, HZD



(c) Experiment, IO

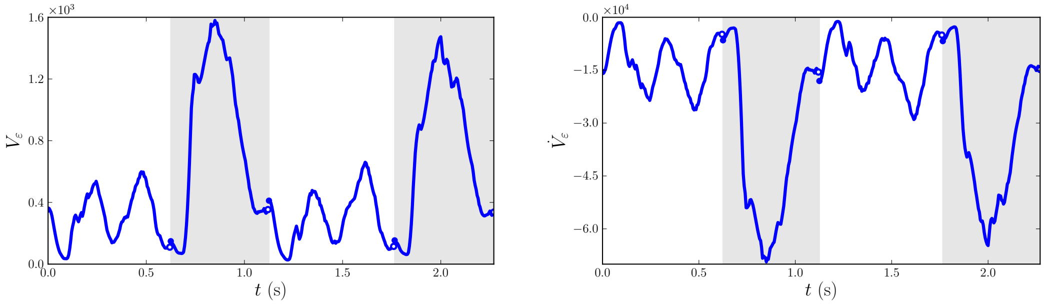
(d) Experiment, CLF

Figure A.4: Comparison of phase variable for the Light Torso configuration.



(a) Simulation, V_ε

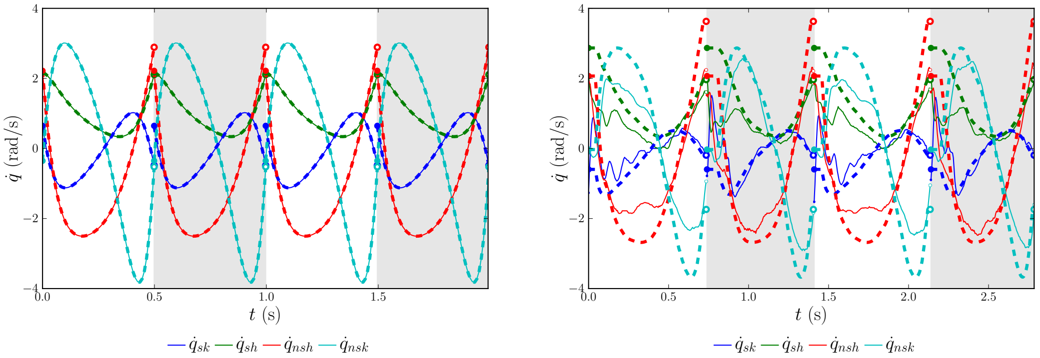
(b) Simulation, $\dot{V}_\varepsilon$



(c) Experiment, V_ε

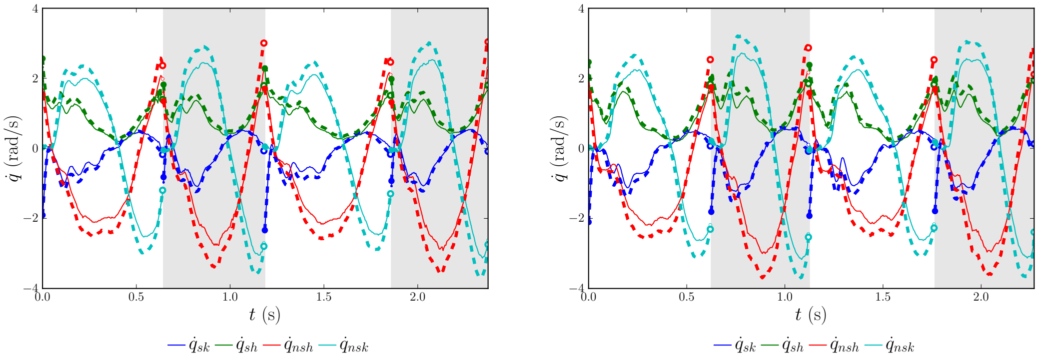
(d) Experiment, $\dot{V}_\varepsilon$

Figure A.5: Comparison of Lyapunov function evolution through 4 steps for the Light Torso configuration.



(a) Simulation, IO

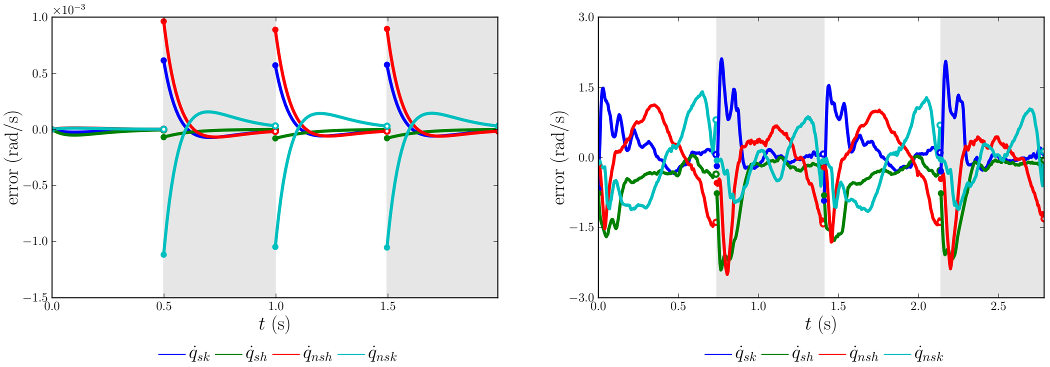
(b) Experiment, HZD



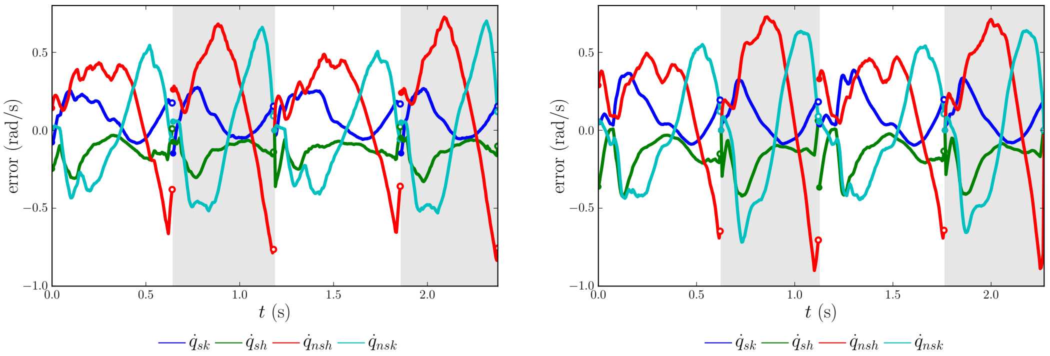
(c) Experiment, IO

(d) Experiment, CLF

Figure A.6: Comparison of velocity tracking for the Light Torso configuration where the solid lines are the actual positions and the dotted lines are the desired positions.

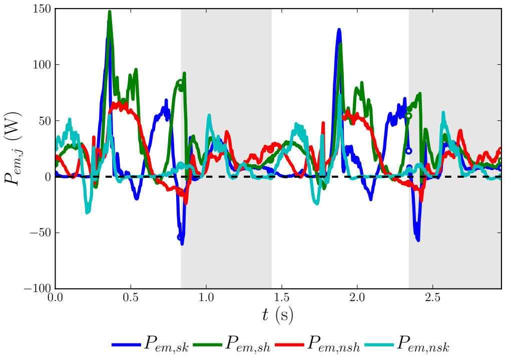


(a) Simulation, IO (b) Experiment, HZD

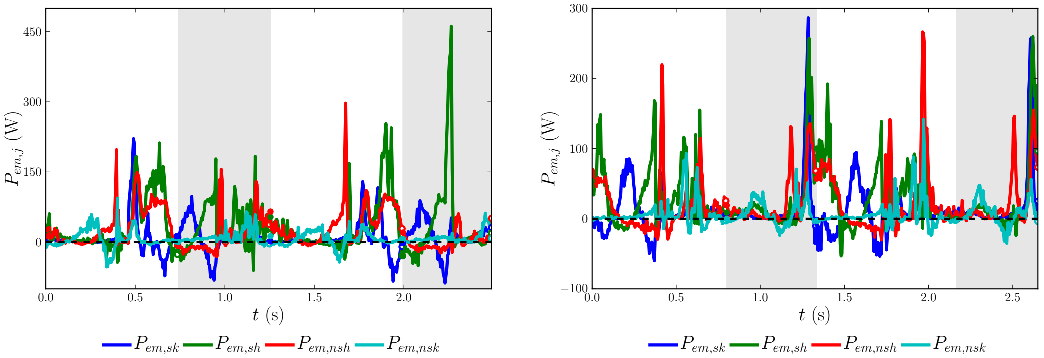


(c) Experiment, IO (d) Experiment, CLF

Figure A.7: Comparison of velocity errors for the Light Torso configuration.



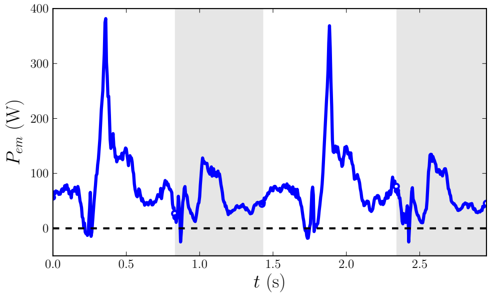
(a) Experiment, HZD



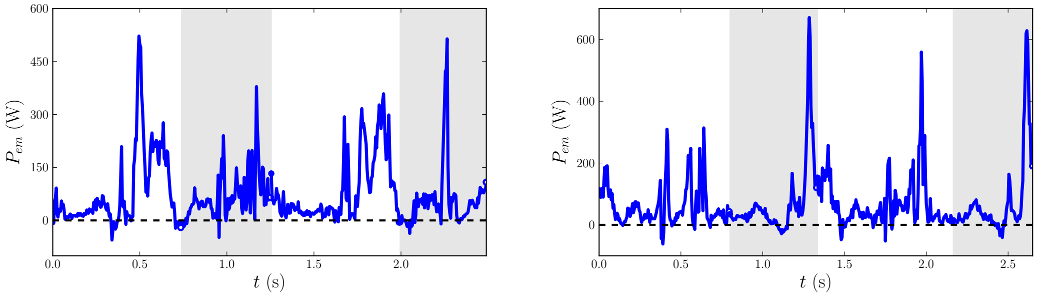
(b) Experiment, IO

(c) Experiment, CLF

Figure A.8: Comparison of electrical power for individual motors, $P_{em,j}$, for the Heavy Torso configuration.



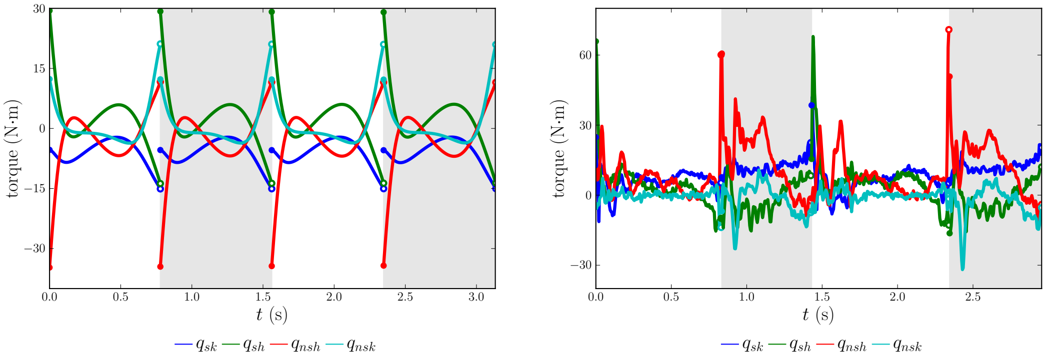
(a) Experiment, HZD



(b) Experiment, IO

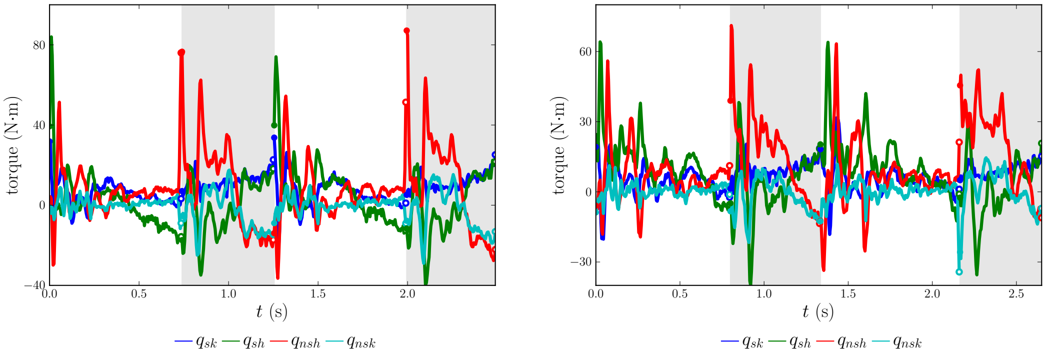
(c) Experiment, CLF

Figure A.9: Comparison of electrical power for all motors, P_{em}, for the Heavy Torso configuration.



(a) Simulation, IO

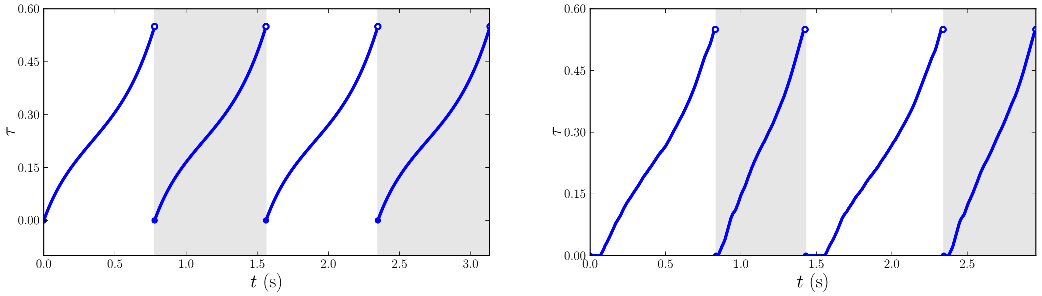
(b) Experiment, HZD



(c) Experiment, IO

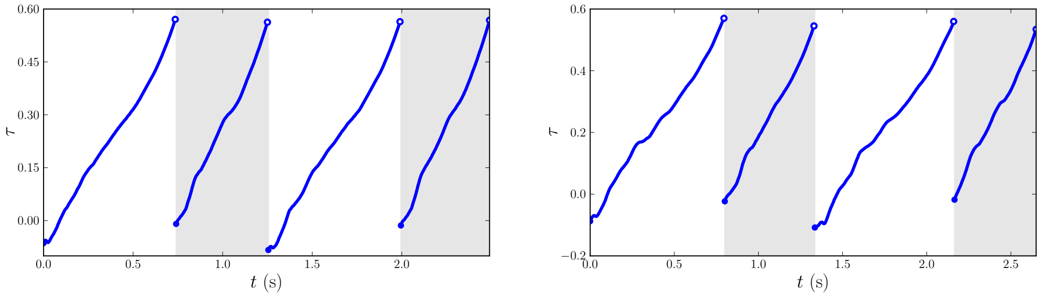
(d) Experiment, CLF

Figure A.10: Comparison of torques measured on the system for the Heavy Torso configuration.



(a) Simulation, IO

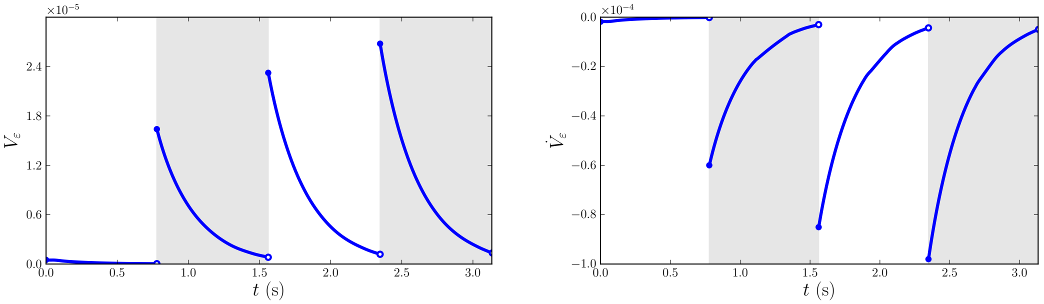
(b) Experiment, HZD



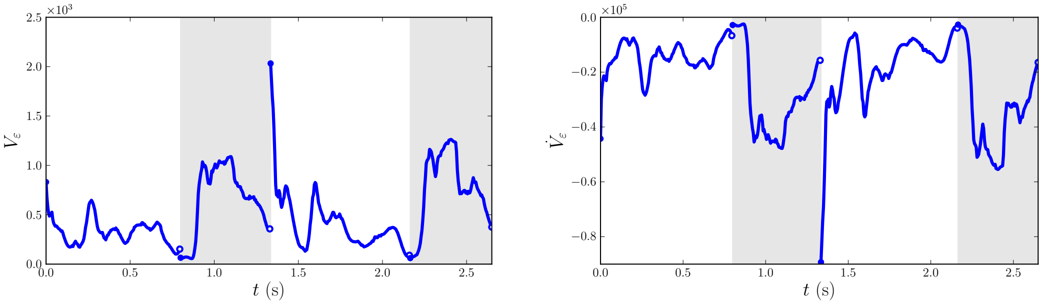
(c) Experiment, IO

(d) Experiment, CLF

Figure A.11: Comparison of phase variable for the Heavy Torso configuration.

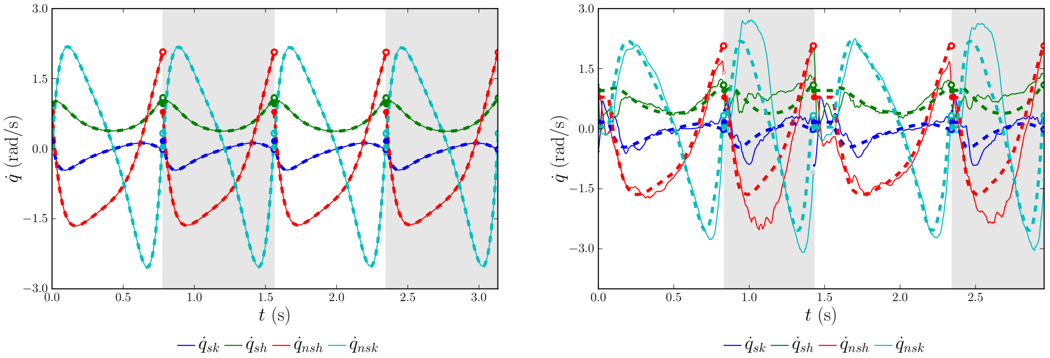


(a) Simulation, V_ε (b) Simulation, $\dot{V}_\varepsilon$

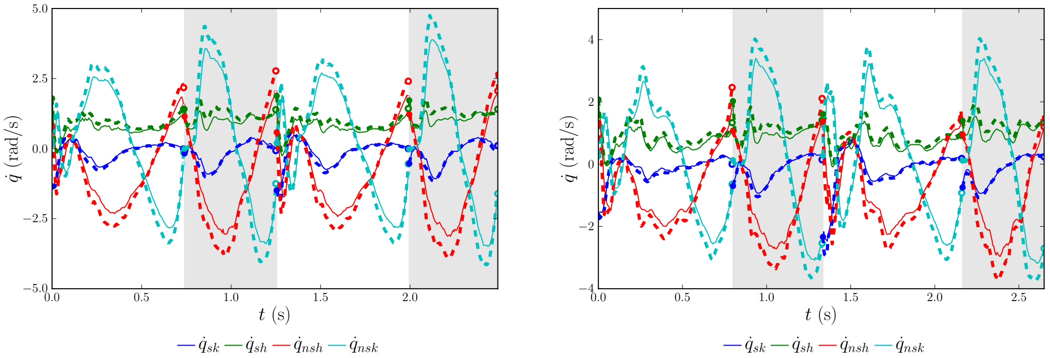


(c) Experiment, V_ε (d) Experiment, $\dot{V}_\varepsilon$

Figure A.12: Comparison of Lyapunov function evolution through 4 steps for the Heavy Torso configuration.

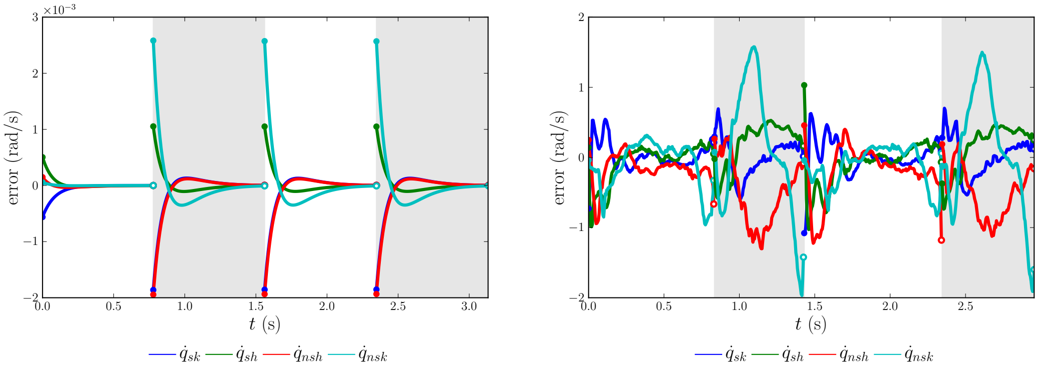


(a) Simulation, IO (b) Experiment, HZD

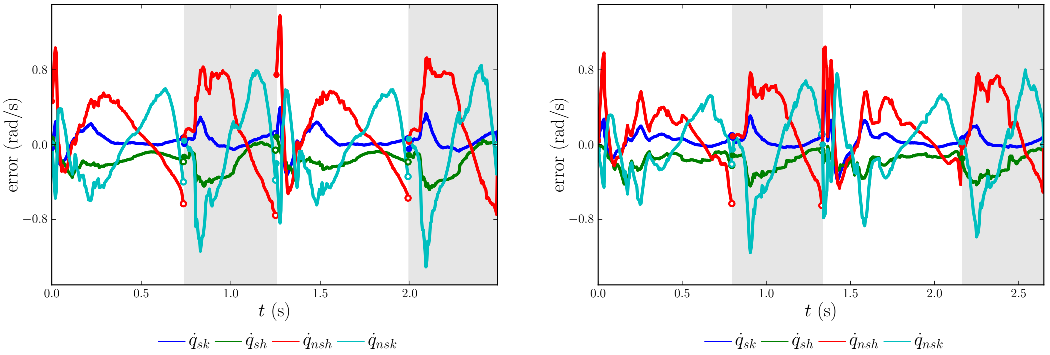


(c) Experiment, IO (d) Experiment, CLF

Figure A.13: Comparison of velocity tracking for the Heavy Torso configuration where the solid lines are the actual positions and the dotted lines are the desired positions.



(a) Simulation, IO (b) Experiment, HZD



(c) Experiment, IO (d) Experiment, CLF

Figure A.14: Comparison of velocity errors for the Heavy Torso configuration.

APPENDIX B

KINEMATIC MODELING

This section reviews basic definitions of kinematic modeling as a means to distinguish different conventions used in literature. A *coordinate frame* is defined as an *origin* and an orientation. A *Cartesian coordinate* for a point $\boldsymbol{p}$ represented in coordinate frame A, in the scope of relevant literature, is a vector in either a two- or three-dimensional space, with the common ordering of values specified as (x, y, z), which are the distances of point $\boldsymbol{p}$ from the origin of A along each of the axes defined as $[\boldsymbol{e}_{Ax}\ \boldsymbol{e}_{Ay}\ \boldsymbol{e}_{Az}]$, which are orthogonal unit vectors. These axes together define a special orthonormal group, $SO(2)$ for two-dimensional and $SO(3)$ for three-dimensional.

As shown in robotics literature [59, 42, 15], a *rotation matrix* can be defined as a linear transformation that projects coordinates from the orientation of frame A to the orientation of frame B:

$$^{B}\boldsymbol{R}_{A} = \begin{bmatrix} ^{B}\boldsymbol{e}_{Ax} & ^{B}\boldsymbol{e}_{Ay} & ^{B}\boldsymbol{e}_{Az} \end{bmatrix} \tag{B.1}$$

where $^{B}\boldsymbol{e}_{Ai}$ is the axis i of frame A expressed in the coordinates of B, following the notation of Featherstone [15].[1] Note that these rotation matrices all necessarily belong to $SO(3)$.

In order to simplify analysis, authors in literature have introduced the notion of *homogeneous transformations*, which allows linear algebra to be used for performing what normally would be affine transformations in Cartesian space when translation is involved. When transforming the coordinates of a point, $\boldsymbol{p}$, from frame A to B,

---

[1] Spong [59] uses the notation $\boldsymbol{R}_{A}^{B}$, and Murray [42] uses the notation $\boldsymbol{R}_{BA}$.

we may define the homogeneous transformation:

$$
{}^{B}\boldsymbol{T}_{A} = \begin{bmatrix} {}^{B}\boldsymbol{R}_{A} & {}^{B}\boldsymbol{p}_{A} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{B.2}
$$

where ${}^{B}\boldsymbol{R}_{A}$ transforms the orientation of $\boldsymbol{p}$, and ${}^{B}\boldsymbol{p}_{A}$ translates the point which takes the value of frame A's position in B's coordinates. If ${}^{A}\boldsymbol{p}$ and ${}^{B}\boldsymbol{p}$ are the coordinates of point in frames A and B, respectively, we have the relationship ${}^{B}\boldsymbol{p} = {}^{B}\boldsymbol{T}_{A}{}^{A}\boldsymbol{p}$.

For a rigid body manipulator, multiple techniques may be used to define the chain of transformations necessary to compute the *forward kinematics*, or the Cartesian coordinates, of a manipulator. These techniques range from the Denavit-Hartenberg convention [59] to exponential twists [42]. Since there was existing code from others in the lab for defining the system using exponential twists, this convention was employed. For a frame i driven by a degree of freedom i and an offset ${}^{i}\boldsymbol{o}$ representing a point $\boldsymbol{p}_i$, the coordinates in the “root” frame 0 are represented as:

$$
{}^{0}\boldsymbol{p}_{i}(\boldsymbol{q}) = \left(\prod_{k \in \kappa(i)} {}^{\lambda(k)}\boldsymbol{T}_{k} \right) \cdot {}^{i}\boldsymbol{o} \tag{B.3}
$$

where $\kappa(i)$ is the sorted chain of parents, including joint i, and $\lambda(i)$ is the immediate parent of joint i, as defined in [15, p. 72].

Since rotation is primary concept in the modeling of rigid bodies, the rotational velocity is also important. Given a rotation matrix, $\boldsymbol{R}(\boldsymbol{q}) \in SO(3)$, it's time-derivative, shown in [59, 15] is

$$
\dot{\boldsymbol{R}} = \boldsymbol{\omega} \times \boldsymbol{R} \tag{B.4}
$$

where $\boldsymbol{\omega}$ is the angular velocity of the body whose orientation is defined by $\boldsymbol{R}$.

APPENDIX C

CONSTRAINED AND PINNED DYNAMICS

C.1 Constrained Dynamics

Dynamics with a *holonomic constraint*, a function of only configuration or time which requires no work, represented as $\boldsymbol{h}_c(\boldsymbol{q})$, where $\boldsymbol{h}_c : \mathcal{Q} \rightarrow \mathbb{R}^{n_c}$, is modeled using Lagrange multipliers such that $\dot{\boldsymbol{h}}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{0}_{n_c}$, and by relation, $\ddot{\boldsymbol{h}}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{0}_{n_c}$. First, the dynamics are modified resulting in:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{J}_c^T \boldsymbol{\lambda}, \tag{C.1}$$

which combined with the acceleration-based constraint

$$\ddot{\boldsymbol{h}}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{J}_c\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{C.2}$$

where $\boldsymbol{\lambda}$ are the multipliers, or constraint forces, which are solved for by substituting the expression for $\ddot{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u})$, which yields the dynamics as in [54]:

$$\begin{aligned}
\boldsymbol{D}\ddot{\boldsymbol{q}} + \bar{\boldsymbol{N}}_c\boldsymbol{H} &= \bar{\boldsymbol{N}}_c\boldsymbol{B}\boldsymbol{u} - \boldsymbol{J}_c^T\bar{\boldsymbol{D}}_c\dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} && \text{(C.3)}\\
\bar{\boldsymbol{D}}_c(\boldsymbol{q}) &= (\boldsymbol{J}_c\boldsymbol{D}^{-1}\boldsymbol{J}_c^T)^{-1} && \text{(C.4)}\\
\bar{\boldsymbol{N}}_c(\boldsymbol{q}) &= \boldsymbol{I}_{n\times n} - \boldsymbol{J}_c^T\bar{\boldsymbol{D}}_c\boldsymbol{J}_c\boldsymbol{D}^{-1} && \text{(C.5)}\\
\boldsymbol{\lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}) &= -\bar{\boldsymbol{D}}_c\left[\dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} + \boldsymbol{J}_c\boldsymbol{D}^{-1}\left(\boldsymbol{B}\boldsymbol{u} - \boldsymbol{H}\right)\right] && \text{(C.6)}
\end{aligned}$$

where $\bar{\boldsymbol{D}}_c$ is the constrained inertia matrix, and $\bar{\boldsymbol{N}}_c$ is the dynamically consistent null space of the constraints. In the context of control, the following cases are used for defining the cases of actuation, with $n_u = n - n_c$ is the number of unconstrained degrees of freedom [56, 35]:

1. $m = n_u$: the system is *fully actuated*;
2. $m > n_u$: the system is *over actuated*; and
3. $m < n_u$: the system is *underactuated*.

C.2 Constraining Configuration Variables and State Space Reduction

For the author's satisfaction, the following steps were taken to mathematically ensure identical dynamics when constraining multiple degrees of freedom versus eliminating the coordinates from the equations of motion.

Assume that the coordinates are (re)arranged such that $\boldsymbol{q} = (\boldsymbol{q}_c, \boldsymbol{q}_u)^T$, with $\boldsymbol{q}_c \in \mathbb{R}^{n_c}$ and $\boldsymbol{q}_u \in \mathbb{R}^{n_u}$, and the constraint be defined as $\boldsymbol{h}_c(\boldsymbol{q}) = \boldsymbol{q}_c$, such that

$$\boldsymbol{J}_c = \begin{bmatrix} \boldsymbol{I}_{n_u \times n_u} & \boldsymbol{0}_{n_c \times n_c}, \end{bmatrix} \tag{C.7}$$

and subdivide the inertia tensor, $\boldsymbol{D}(\boldsymbol{q})$, as:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_{cc} & \boldsymbol{D}_{cu} \\ \boldsymbol{D}_{uc} & \boldsymbol{D}_{uu} \end{bmatrix}, \tag{C.8}$$

where $\boldsymbol{D}_{uc} = \boldsymbol{D}_{cu}^T$ due to the fact that $\boldsymbol{D} = \boldsymbol{D}^T$.

Given the constrained dynamics from (C.3), with $\dot{\boldsymbol{J}}_c = \boldsymbol{0}_{n_c \times n}$ incorporated, and rearranging the equations to yield:

$$\ddot{\boldsymbol{q}} = \begin{bmatrix} \ddot{\boldsymbol{q}}_c \\ \ddot{\boldsymbol{q}}_u \end{bmatrix} = \boldsymbol{D}^{-1}\bar{\boldsymbol{N}}_c\boldsymbol{b} \tag{C.9}$$

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_c \\ \boldsymbol{b}_u \end{bmatrix} = -\boldsymbol{H} + \boldsymbol{B}\boldsymbol{u}, \tag{C.10}$$

the goal is to show that constraining $\ddot{\boldsymbol{q}}_c = \boldsymbol{0}_{n_c}$ via Lagrange multipliers yields dynamics identical to:

$$\ddot{\boldsymbol{q}}_u = \boldsymbol{D}_{uu}^{-1}\boldsymbol{b}_u. \tag{C.11}$$

To begin, let $\boldsymbol{E} = \boldsymbol{D}^{-1}$, and subdivide it equivalently as:

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{E}_{cc} & \boldsymbol{E}_{cu} \\ \boldsymbol{E}_{uc} & \boldsymbol{E}_{uu} \end{bmatrix}. \tag{C.12}$$

Next, we incorporate the identity of the block-wise matrix inversion[1], which may be generically defined as:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \\ -(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & (\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \end{bmatrix}, \tag{C.13}$$

which will be used later.

---

[1] `http://en.wikipedia.org/wiki/Invertible_matrix#Blockwise_inversion`

First, we compute the constrained inertia matrix,

$$\begin{aligned}\bar{\boldsymbol{D}}_c^{-1} &= \boldsymbol{J}_c \boldsymbol{D}^{-1} \boldsymbol{J}_c^T = \boldsymbol{J}_c \boldsymbol{E} \boldsymbol{J}_c^T \\ &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_{cc} & \boldsymbol{E}_{cu} \\ \boldsymbol{E}_{uc} & \boldsymbol{E}_{uu} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{E}_{cc},\end{aligned}$$

followed by the dynamically consistent null-space:

$$\begin{aligned}\bar{\boldsymbol{N}}_c &= \boldsymbol{I}_{n\times n} - \boldsymbol{J}_c^T \bar{\boldsymbol{D}}_c \boldsymbol{J}_c \boldsymbol{D}^{-1} \\ &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} - \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{E}_{cc}^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_{cc} & \boldsymbol{E}_{cu} \\ \boldsymbol{E}_{uc} & \boldsymbol{E}_{uu} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{E}_{cc}^{-1} \boldsymbol{E}_{cu} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}.\end{aligned}$$

Incorporating this into (C.9) yields:

$$\begin{aligned}
\begin{bmatrix} \ddot{\boldsymbol{q}}_c \\ \ddot{\boldsymbol{q}}_u \end{bmatrix} &= \begin{bmatrix} \boldsymbol{E}_{cc} & \boldsymbol{E}_{cu} \\ \boldsymbol{E}_{uc} & \boldsymbol{E}_{uu} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{E}_{cc}^{-1}\boldsymbol{E}_{cu} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_c \\ \boldsymbol{b}_u \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{E}_{cc} & \boldsymbol{E}_{cu} \\ \boldsymbol{E}_{uc} & \boldsymbol{E}_{uu} \end{bmatrix} \begin{bmatrix} -\boldsymbol{E}_{cc}^{-1}\boldsymbol{E}_{cu}\boldsymbol{b}_u \\ \boldsymbol{b}_u \end{bmatrix} \\
&= \begin{bmatrix} -\boldsymbol{E}_{cc}^{-1}\boldsymbol{E}_{cc}\boldsymbol{E}_{cu}\boldsymbol{b}_u + \boldsymbol{E}_{cu}\boldsymbol{b}_u \\ -\boldsymbol{E}_{uc}\boldsymbol{E}_{cc}^{-1}\boldsymbol{E}_{cu}\boldsymbol{b}_u + \boldsymbol{E}_{uu}\boldsymbol{b}_u \end{bmatrix} \\
\begin{bmatrix} \ddot{\boldsymbol{q}}_c \\ \ddot{\boldsymbol{q}}_u \end{bmatrix} &= \begin{bmatrix} \boldsymbol{0} \\ (\boldsymbol{E}_{uu} - \boldsymbol{E}_{uc}\boldsymbol{E}_{cc}^{-1}\boldsymbol{E}_{cu})\boldsymbol{b}_u \end{bmatrix}.
\end{aligned}$$

Note that, according to (C.13),

$$\boldsymbol{D}_{uu}^{-1} = \boldsymbol{E}_{uu} - \boldsymbol{E}_{uc}\boldsymbol{E}_{cc}^{-1}\boldsymbol{E}_{cu}, \tag{C.14}$$

thus yielding the final results:

$$\begin{bmatrix} \ddot{\boldsymbol{q}}_c \\ \ddot{\boldsymbol{q}}_u \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{D}_{uu}^{-1}\boldsymbol{b}_u \end{bmatrix}. \tag{C.15}$$

APPENDIX D

REVIEW OF REDUCED ORDER ZERO DYNAMICS

Given the HZD reconstruction in Sec. 2.3.2, and repeating the zero dynamics coordinates from (2.66):[1]

$$\xi_1(\boldsymbol{q}) = \delta p^x_{hip}(\boldsymbol{q}) = \boldsymbol{J}^x_{hip}\boldsymbol{q} \tag{D.1}$$

$$\xi_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\gamma}_0(\boldsymbol{q})\dot{\boldsymbol{q}} := \gamma(\boldsymbol{x}), \tag{D.2}$$

the solution of $\boldsymbol{\xi}(t, \boldsymbol{\xi}_0)$ may be integrated using the zero dynamics $\boldsymbol{f}|_{\mathbf{Z}}(\boldsymbol{\xi})$ defined as:

$$\begin{aligned} \dot{\xi}_1 &= \kappa_1(\xi_1)\xi_2 && \text{(D.3)} \\ &= \boldsymbol{J}^x_{hip}\dot{\boldsymbol{q}}^r(\xi_1, \xi_2) = \boldsymbol{J}^x_{hip}\boldsymbol{\Psi}_{\mathbf{Z}}(\xi_1)\xi_2 \\ \dot{\xi}_2 &= \kappa_2(\xi_1) && \text{(D.4)} \\ &= -G_1(\boldsymbol{q}^r(\xi_1)) = -G_1(\boldsymbol{\Phi}_{\mathbf{Z}}(\xi_1)) \end{aligned}$$

where $G_1(\boldsymbol{q})$ is the first entry in the gravity vector $\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial \mathcal{V}}{\partial \boldsymbol{q}}(\boldsymbol{q})$. The dynamics of ξ_1 in (D.3) is easy to see as $\delta \dot{p}^x_{hip}(\boldsymbol{q}^r)$.

The dynamics of ξ_2 in (D.4) come from Remark 5.2 in the proof of Theorem 5.1 found in [67, p. 122], first noting that q_{sa} is a *cyclic coordinate*,[2] implying that the state itself does not affect the kinetic energy, $\frac{\partial \mathcal{T}}{\partial q_{sa}} = 0$.

---

[1] Note that units of ξ_1 is m (distance) and the units of ξ_2 is $\mathrm{N} \cdot \mathrm{s}$ (momentum).

[2] This is explained in depth in Proposition B.8 in [67, p. 426]. In essence, q_{sa} is a cyclic coordinate because it does not directly change the shape / inertia of the robot to affect the kinetic energy.

The direct derivative of ξ_2 is:

$$\dot{\xi}_2 = \frac{d}{dt}\left(\boldsymbol{\gamma}_0 \dot{\boldsymbol{q}}\right) = \boldsymbol{\gamma}_0 \ddot{\boldsymbol{q}} + \dot{\boldsymbol{\gamma}}_0 \dot{\boldsymbol{q}} = \boldsymbol{D}_{1,*} \ddot{\boldsymbol{q}} + \dot{\boldsymbol{D}}_{1,*} \dot{\boldsymbol{q}} = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_{sa}}, \tag{D.5}$$

noting that:

$$\xi_2 = \boldsymbol{\gamma}_0 \dot{\boldsymbol{q}} = \frac{\partial \mathcal{L}}{\partial \dot{q}_{sa}}, \tag{D.6}$$

and taking the Lagrangian of the dynamics of the non-actuated q_{sa} and incorporating the property of the cyclic coordinate yields:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_{sa}} - \frac{\partial \mathcal{L}}{\partial q_{sa}} = \frac{d}{dt}\frac{\partial \mathcal{T}}{\partial \dot{q}_{sa}} + \frac{\partial \mathcal{V}}{\partial q_{sa}} = 0,$$

which then simplifies to:

$$\dot{\xi}_2 = \frac{d}{dt}\frac{\partial \mathcal{T}}{\partial \dot{q}_{sa}} = -\frac{\partial \mathcal{V}}{\partial q_{sa}} = -G_1(\boldsymbol{q}^r).$$

Using these reduced coordinates, numeric constraints can be made that (a) ensure that the walking gait does not fall forward with increasing velocity, and that (b) that a nontrivial periodic orbit exists using energy methods. In order to consider these quantities, we consider the pre-impact fixed point $\boldsymbol{x}^- = (\boldsymbol{q}^-, \dot{\boldsymbol{q}}^-)^T$ and its resulting post-impact value, $\boldsymbol{x}^+ = (\boldsymbol{q}^+, \dot{\boldsymbol{q}}^+)^T$, and considering the reduced zero dynamics coordinates $\boldsymbol{\xi}^-(\boldsymbol{x}^-) = (\xi_1^-, \xi_2^-)^T$ and $\boldsymbol{\xi}^+(\boldsymbol{x}^+) = (\xi_1^+, \xi_2^+)^T$. Addressing (a) is straightforward, shown in [3, 67], that ξ_2^+ may be computed as:

$$\xi_2^+ = \Delta_{\xi_2}(\xi_1^-)\xi_2^- \tag{D.7}$$

$$\Delta_{\xi_2}(\xi_1^-) = \boldsymbol{\gamma}_0(\boldsymbol{q}^{r+}(\xi_1^-))\boldsymbol{\Delta}_{\dot{\boldsymbol{q}}}(\boldsymbol{q}^{r+}(\xi_1^-))\boldsymbol{\Psi}_{\mathbf{Z}}(\xi_1) \tag{D.8}$$

$$\boldsymbol{q}^{r+}(\xi_1^-) = \boldsymbol{\Delta}_{\boldsymbol{q}}\boldsymbol{\Phi}_{\mathbf{Z}}(\xi_1^-)) \tag{D.9}$$

$$\tag{D.10}$$

and knowing that the walking gait can be stable if the result velocity is forward moving and does not grow:

$$0 < \Delta_{\xi_2} < 1. \tag{D.11}$$

The more challenging construction is proving that there is a nontrivial periodic orbit for the reduced coordinates that implies a fixed point in the effective underactuated momentum defined by ξ_2. First, with the goal of finding a coordinate that is integrable in terms of solely ξ_1, first define the ratio:

$$\frac{\dot{\xi}_2}{\dot{\xi}_1}(\xi_1, \xi_2) = \frac{d\xi_2}{d\xi_1}(\xi_1, \xi_2) = \frac{\kappa_2(\xi_1)}{\kappa_1(\xi_1)\xi_2}, \tag{D.12}$$

and with the goal of eliminating ξ_2 on the denominator, define the "effective momentum" energy:[3]

$$E_2(\xi_2) = \frac{1}{2}(\xi_2)^2, \tag{D.13}$$

noting that

$$E_2^+ = \Delta_{\xi_2}^2 E_2^-,$$

---

[3] In literature [67, 3], this is normally defined as ζ_2, but the author chose this notation to emphasize its relation to energy.

where $\Delta_{\xi_2} = \Delta_{\xi_2}(\xi_1^-)$, and compute its derivative with respect to ξ_1 yielding:

$$\frac{dE_2}{d\xi_1}(\xi_1) = \frac{\kappa_2(\xi_1)}{\kappa_1(\xi_1)}, \tag{D.14}$$

and thus the integration to find $\hat{E}_2(\xi_1)$ may defined as:[4]

$$\hat{E}_2(\xi_1) = E_2^+ + \delta E_2(\xi_1), \tag{D.15}$$

where[5]

$$\delta E_2(\xi_1) = \int_{\xi_1^-}^{\xi_1} \frac{dE_2}{d\xi_1}(\hat{\xi}_1) d\hat{\xi}_1.$$

An important constraint mentioned by [67] is that $\hat{E}_2(\xi_1) > 0$ for all $\xi_1 \in [\xi_1^-, \xi_1^+]$, since the value itself must be positive. Thus we have the constraint:

$$\begin{aligned} \min_{\xi_1} \hat{E}_2(\xi_1) = E_2^+ + \delta E_2^{\min} > 0 \\ \delta E_2^{\min} = \min_{\xi_1} \delta E_2(\xi_1). \end{aligned} \tag{D.16}$$

Using this identity, we may then define the Poincaré return map of $E_2(\xi_1)$ at impact as:

$$\rho(E_2^-) = \hat{E}_2(\xi_1^-) = \Delta_{\xi_2}^2 E_2^- + \delta E_2(\xi_1^-), \tag{D.17}$$

which can the be used to find the fixed-point energy for a nontrivial periodic orbit, $\rho(E_2^{\bar{*}}) = E_2^{\bar{*}}$, which results in:

$$E_2^{\bar{*}} = \frac{1}{1 - \Delta_{\xi_2}^2} \delta E_2(\xi_1^-). \tag{D.18}$$

---

[4] Note that $E_2(\xi_2)$ is the coordinate transformation, and $\hat{E}_2(\xi_1)$ is the solution for E_2.

[5] Comparing to literature, $\delta E_2(\xi_1) = -\mathcal{V}_{\mathbf{Z}}(\xi_1)$.

For this fixed point to exist, it must be positive and be a member of the *zero dynamics energy domain*:

$$\mathcal{D}_{\mathbf{Z}}^{E_2} = \{E_2^- > 0 : E_2^{-\min} > 0\} \tag{D.19}$$

$$E_2^{-\min} = \Delta_{\xi_2}^2 E_2^- + \delta E_2^{\min} \tag{D.20}$$

which reinforces that the energy must always be positive. This results in the constraints:

$$E_2^{\bar{*}} > 0 \tag{D.21}$$

$$E_2^{\bar{*}\min} = \Delta_{\xi_2}^2 E_2^{\bar{*}} + \delta E_2^{\min} > 0 \Leftrightarrow E_2^{\bar{*}} \in \mathcal{D}_{\mathbf{Z}}^{E_2}. \tag{D.22}$$