Dynamic Balance and Gait Metrics for Robotic Bipeds

by

Brandon J. DeHart

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Wael Suleiman Assoc. Professor, Electrical and Computer Engineering Dept. Université de Sherbrooke
Supervisors:	Dana Kulić Professor, Dept. of Electrical and Computer Systems Engineering, Cross Appoint., Dept. of Mechanical and Aerospace Engineering, Monash University Assoc. Professor, Dept. of Electrical & Computer Engineering, University of Waterloo
	Rob Gorbet Assoc. Professor, Dept. of Knowledge Integration, Cross Appoint., Dept. of Electrical & Computer Engineering, University of Waterloo
Internal Members:	Dan Miller Professor, Dept. of Electrical & Computer Engineering, University of Waterloo
	Christopher Nielsen Assoc. Professor, Dept. of Electrical & Computer Engineering, University of Waterloo
Internal-External Member:	John McPhee Professor, Dept. of Systems Design Engineering, University of Waterloo

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

For legged robots to be useful in the real world, they must be able to balance and walk reliably. Both of these abilities improve when a system is more effective at moving itself around relative to its contacts (i.e., its feet). Achieving this type of movement depends both on the controller used to perform the motion and the physical properties of the system. Although much work has been done on the development of dynamic controllers for balance and gait, only limited research exists on how to quantify a system's physical balance capabilities or how to modify the system to improve those capabilities.

From the control perspective, there are three strategies for maintaining balance in bipeds: flexing, leaning, and stepping. Both stepping and leaning strategies typically depend on balance points (critical points used for maintaining or regaining balance) to determine whether or not a step is needed, and if so, where to step. Although several balance point estimators exist, the majority of these methods make undesirable assumptions (e.g., ignoring the impact dynamics, assuming massless legs, planar motion, etc.).

From the physical design perspective, one promising approach for analyzing system performance is a set of dynamic ratios called velocity and momentum gains, which are dependent only on the (scale-invariant) dynamic parameters and instantaneous configuration of a system, enabling entire classes of mechanisms to be analyzed at the same time.

This thesis makes four key contributions towards improving biped balancing capabilities. First, a dynamic bipedal controller is proposed which uses a 3D balance point estimator both to respond to disturbances and produce reliable stepping. Second, a novel balance point estimator is proposed that facilitates stepping while combining and expanding the features of existing 2D and 3D estimators to produce a generalized 3D formulation.

Third, the momentum gain formulation is extended to general 2D and 3D systems, then both gains are compared to centroidal momentum via a spatial formulation and incorporated into a generalized gain definition. Finally, the gains are used as a metric in an optimization framework to design parameterized balancing mechanisms within a given configuration space. Effectively, this enables an optimization of how well a system could balance without the need to pre-specify or co-generate controllers and/or trajectories.

To validate the control contributions, simulated bipeds are subjected to external disturbances while standing still and walking. For the gain contributions, the framework is used to compare gain-optimized mechanisms to those based on the cost of transport metric.

Through the combination of gain-based physical design optimization and the use of predictive, real-time balance point estimators within dynamic controllers, bipeds and other legged systems will soon be able to achieve reliable balance and gait in the real world.

Acknowledgements

First, I would like to thank my supervisors, Prof. Dana Kulić and Prof. Rob Gorbet, for everything they have done over the years to support my academic pursuits. It has been a bit of a rollercoaster ride with all the ups and downs we've been through to get to this point and you have both stayed true to your roles despite it all. I look forward to sitting down with you both once the ride is over to reminisce as friends and colleagues.

I would also like to thank the members of my committee, Prof. John McPhee, Prof. Chris Nielsen, and Prof. Dan Miller, for their insightful questions and constructive feedback throughout the PhD process which have dramatically improved this final product.

I have benefited from being a part of a great cohort of grad students and post-docs in the Adaptive Systems Lab who were always willing to lend an ear and share some common sense feedback. In particular, I'd like to thank Jonathan Lin and Vladimir Joukov for their patience when I would show up in their offices and ramble about my research.

As part of my time at the University of Waterloo, I have also been lucky enough to teach and work on campus with some fantastic colleagues, some of whom I've known since I arrived to start my first degree here in 2004. Specifically, I would like to thank Prof. Sanjeev Bedi and Prof. William Melek for their confidence in my abilities and for making sure I consider the big picture throughout my studies and into my professional life.

This thesis would never have happened if it weren't for the continuous support and love that I receive regularly from my family and friends, despite most of them living on the other side of the country! This is especially true for my parents, Jim and Darlene, who have always been there for us when it was needed most with unconditional love and generosity. Thank you all, I appreciate everything you do and have done to help us thrive, and I look forward to celebrating the end of my student status with you.

Finally, I would like to thank my wife, Sarah, for her love, her unending support, her encouragement... really, for everything. Everything she has done, has created, has given up, has put on hold, has taken care of, has contributed, has endured... (to name a few) while I was working my way through this degree (and the last one, for that matter). I never would have been able to finish this PhD without you by my side.

Thank you!

Dedication

This thesis is dedicated to my kids, Malcolm and Maisie, who have both spent their whole lives up to this point with a father who is "working on his PhD".

Table of Contents

List of Tables

List of Figures

Nomenclature

This list provides a reference for the main symbols used in this thesis. Modifications and/or exceptions to this list will be clearly stated where they happen in the text.

a, A	scalar
$\hat{A}$	point (in 2D or 3D space)
$\vec{a}, \vec{A}$	common vector (2D or 3D)
$\hat{a}$	common unit vector (2D or 3D)
$\tilde{a}$	skew symmetric matrix (3×3) generated using the 3D vector $\vec{a}$ (see A.17)
$\tilde{A}$	common matrix (3×3)
$\vec{\boldsymbol{a}}, \vec{\boldsymbol{A}}$	spatial vector (6D)
$\hat{\boldsymbol{a}}$	spatial unit vector (6D)
$\tilde{\boldsymbol{a}}$	block matrix (6×6) generated using the 6D spatial vector $\vec{\boldsymbol{a}}$ (see A.18)
$\tilde{\boldsymbol{A}}$	spatial matrix (6×6)
$\boldsymbol{a}$	general vector
$\boldsymbol{A}$	general matrix

Chapter 1

Introduction

For two-legged robots, or bipeds, to be useful in the real world, they must be able to safely walk and work in human environments, even when faced with external disturbances. This requires bipeds to move reliably over real terrain and recover from many environmental challenges, such as rocks underfoot, carrying shifting loads, or being jostled on the sidewalk.

For bipeds and other legged systems, both balance and gait performance improve when the system is more effective at moving its Center of Mass (COM) relative to its contacts (i.e., its feet). Whether it is standing still or walking, external disturbances and unknown environments may also require real-time adjustments to this COM motion. The capacity of a legged system to move its COM relative to its contacts depends on both the controller used to perform the motion and the system's physical design.

Controllers used to maintain or recover balance typically use disturbance compensation methods called balance strategies, which mainly fall into three categories: *flexing*, by pushing against the ground and other contacts; *leaning*, by changing the system's momentum or posture; and *stepping*, by stepping out to change the contact surface(s).

Flexing strategies allow a robot to absorb small disturbances through joint compliance, either passively as part of the robot's mechanical design or actively as part of the controller [1–4]. Flexing is only capable of countering small disturbances, and generally introduces complexity into both the robot's mechatronic design and low-level controller.

Leaning applies changes in momentum (e.g., by varying posture or using momentum actuators such as reaction wheels) to recover from disturbances, often by using momentum-based control [5–10]. However, leaning strategies can only handle small to moderate disturbances, and momentum-based controllers are typically dependent on having an accurate model of the biped's dynamics.

Stepping strategies use balance point estimators to find a point or region where the biped must step to regain balance after a large disturbance [11–14]. These strategies all consist of lifting and then placing a foot somewhere specific (a region or point) on the ground, often repeatedly, to regain balance. In an unknown environment, this usually requires estimates of safe places to step, leading most balance point estimators to assume a smooth, horizontal walking surface.

Several different balance point estimators have been developed, both in 2D and 3D, to determine where a biped should step to maintain or regain its balance [14–20]. However, the majority of these methods are not predictive, assuming that the biped is capable of placing a foot at a range of desired locations instantaneously[1] [17–20]. Those that are predictive often make undesirable assumptions, such as constraining the biped's motion artificially, ignoring impact dynamics, or ignoring angular momentum [14–16].

Although much work has been done to date on developing controllers to achieve balance and gait for legged systems, only limited research has investigated the quantification of a system's innate physical capacity for balance, or how to adjust its design to improve this capacity. The most promising measures of a mechanism's inherent balance abilities are dynamic COM manipulability [21–24] and velocity and momentum gains [25–27].

Recently, Azad *et al.* proposed an updated formulation of dynamic COM manipulability [23], which defines an ellipsoid in 3D space which outlines the system's physical COM acceleration limits. However, this metric depends on the use of a normalizing weighting matrix and the ellipsoid must be projected into lower dimensions to be used for balance. Further work analyzing the impact of the weighting matrices on the dynamic manipulability of these systems has also recently been published [24].

Featherstone developed dynamic ratios called velocity gains [27], which quantify how effectively an articulating system balancing on a passive contact can move its COM. These gains are invariant to a scaling of the system's total mass, and the angular velocity gain is also invariant to a scaling of length, allowing the balance abilities of an entire class of mechanisms to be quantified with one metric [27]. They are also independent of the controller used, as they are functions of the physical properties and configuration of the mechanism, and define a limit on how well any controller could balance the mechanism[2].

[1] In addition to considering the time it takes to lift and move a foot to the desired stepping point, for a balance point estimator to be considered predictive in this work it must also take into account the effects of having finite-length legs and the effects of the robot's existing contacts. Incorporating these additional effects will change the dynamics of the system, and by extension the desired stepping point, which in some cases may mean that a step is no longer required or that multiple steps are needed to avoid falling.

[2] In this context, balance is assumed to be primarily a function of COM motion (i.e., to move the COM above the support polygon). Angular momentum about the COM is assumed to be regulated.

1.1 Contributions

In this thesis, the challenge of biped balance is addressed from two perspectives: how to generate and use continuously updated balance points to maximize balance and disturbance recovery abilities; and the definition and application of physical metrics to both quantify a system's balance capabilities and improve them via mechanical design optimization.

Specifically, this thesis makes the four following contributions to the state of the art in humanoid balance and dynamic gait:

C1. Dynamic Balance and Gait Strategy

A dynamic balance and gait control strategy based on an existing 3D balance point estimator and a continuously updated impact plane, which uses both leaning and stepping balance strategies to respond to external disturbances while standing and walking.

This dynamic balance and gait strategy generates gait trajectories by purposely moving into an unbalanced state, then recovering using stepping. The novelty of this contribution is in the use of a 3D balance point estimator which has only been used in the biomechanics literature in tandem with a continuously updated impact plane, which hasn't been done to date. The goal of this contribution is to demonstrate that this approach to balance and gait is capable of compensating for external, unknown disturbances as part of a gait cycle.

A balance point estimator is used to allow the robot to respond to external disturbances while standing still or walking. One of the key benefits of this strategy is that recovery from external disturbances is a subset of the elements required for dynamic gait. To walk, the biped needs only push itself into an unstable state in the desired direction of motion and allow the recovery behavior to avoid a fall by stepping. To validate the proposed approach, simulations were run to evaluate and compare the two most promising balance point estimators for use as the key metric in the resulting stepping and leaning strategy.

C2. Spherical Foot Placement Estimator

A novel balance point estimator which combines and extends the desirable properties of existing estimators while removing or reducing the majority of their drawbacks.

Building on the simulations in Contribution C1 and the foundational work in [14, 18, 20], a novel 3D balance point estimator called the Spherical Foot Placement Estimator

(SFPE) was developed. The SFPE maps the biped's current kinematics and dynamics to a simplified model to continuously predict whether or not a step is needed, and if so where to step, in order to maintain or restore balance in 3D.

This novel estimator has been developed to address the drawbacks of existing balance point estimators: it is predictive, considers impact, includes rotational inertia, allows ankle torques and 3D Center of Mass (COM) motion, does not require flat ground or heuristic parameters, and works for a biped at rest or in motion.

An example of a controller using the SFPE was also developed and validated in simulation, which augments an existing optimal controller with both leaning and stepping strategies for disturbance rejection: SFPE-based feedback is used to generate a desired momentum for momentum-based leaning while the SFPE point is used as a control reference for stepping. This new estimator outperforms existing balance criteria by providing both recovery step location prediction and momentum objectives with smooth dynamics.

C3. Generalized 2D and 3D Gains

Generalized definitions of momentum gain for 2D and 3D systems, the definition of a novel spatial gain based on changes in centroidal momentum, and the formulation of a generalized gain equation which includes all existing gains as special cases.

This contribution builds on Featherstone's foundational work on the development of velocity and momentum gains [27] by:

- Extending Featherstone's momentum gain formulation from only being defined for planar 2-link inverted pendulums to general 2D and 3D models;
- Defining two different methods for calculating momentum gains for these general 2D and 3D models, including via an additional novel gain defined using spatial notation;
- Relating the definitions of velocity and momentum gains to the system's centroidal momentum via the newly defined spatial gain of the system; and
- Formulating a generalized gain equation which includes all existing gains, the newly defined general momentum gains, and spatial gain as special cases and/or subsets.

The generalized momentum gains inherit the corresponding velocity gains' independence from the applied control scheme and invariance properties, in addition to including a consideration of the inertial properties of the system. These properties make the momentum gains a good choice for analyzing the balance capabilities of a given mechanism, or as a criterion for optimizing the design of balancing mechanisms.

C4. Generalized Optimization Framework

A framework using generalized gains which enables the optimization of parameterized balancing mechanisms without the need to define or generate controllers or trajectories.

An optimization framework has been developed which uses the generalized gains described in Contribution C3 as the core element of its objective functions. This enables the framework to optimize parameterized mechanisms for a given set of behaviors within a specified motion space based purely on the physical properties of the system. The optimization is therefore independent of the controller and/or trajectories used to achieve those behaviors, invariant to gravitational or velocity product dynamics, and (when balancing on a point or line contact and using angular gains) independent of the contact angle.

The generalized gain formulation also enables the application of weighted matrix norms in the objective function, which allows the optimization problem to incorporate domain knowledge. To demonstrate the usefulness of both this framework and the newly developed gains, planar velocity and momentum gains were compared when used as criteria for the design of simple 2- and 3-link planar balancing systems using this framework.

Based on the results of this gain comparison, angular momentum gain was used as the objective in optimizing the design of a 2D 5-link biped. These new results were then compared to optimizing the same mechanism using a cost of transport based objective function[3]. Finally, the use of different norms and weighting matrices in the objective function was analyzed through the optimization of a 3D 5-link biped. This was also used to validate the generalized gain formulations from Contribution C3 on a 3D mechanism.

1.2 Outline

The remainder of this thesis is organized as follows:

In Chapter 2, the necessary background concepts and related work are discussed to position this research relative to the current state of the art in humanoid robotics, including a comparison of the majority of existing balance point estimators.

The stepping strategy used for generating balanced dynamic gait described in Contribution C1 is discussed in Chapter 3, while the novel balance point estimator outlined in Contribution C2 and its use in a leaning and stepping strategy are described in Chapter 4.

[3]The cost of transport is used as the comparative metric for this optimization as it is the most common method of comparing between legged designs, as well as for comparing gait controllers (see Section 2.4).

Chapter 5 derives the gain generalizations of Contribution C3 and shows that the generalized gain equations can be used to formulate both the existing and novel gains described in this work. The generalized gain formulation is also used to show that the differences between velocity and momentum gains are entirely a function of scaling.

Chapter 5 concludes with the introduction of the generalized optimization framework of Contribution C4, along with its associated gain-based objective functions. The framework and these novel objective functions are then applied and validated, along with a discussion of the results, in Chapter 6 via three sets of simulations:

- Analyzing the results of using gains in the objective function by optimizing a set of simple planar mechanisms: 2- and 3-link pendulums and a 3-link biped model;
- Comparing the results of an objective function based on angular momentum gain to one using cost of transport in the optimization of a 2D 5-link biped model; and
- Applying and comparing the use of 1-, 2-, and ∞-norms along with 2 different types of weighting matrices in the optimization of a 3D 5-link biped model.

Finally, Chapter 7 summarizes the results of the thesis, outlines the key contributions, and suggests future directions based on this work.

Chapter 2

Background

This chapter presents the current state of the art in approaches to humanoid balance, as it relates to both control strategies and mechanism design. First, an overview of bipedal dynamics is given in Section 2.1, including definitions and notation for system properties and reference points which will be used throughout this thesis. A set of simplified models are also described, which make various assumptions about the biped's dynamics to facilitate the development, and reduce the complexity of, bipedal algorithms and controllers.

Section 2.2 examines the use of balance point estimators for foot placement. In Section 2.3, existing applications of balance strategies are categorized and discussed at a high level, followed by a focused discussion about the existing methods of combining these strategies. Finally, Section 2.4 outlines existing methods for optimized mechanism design, including two potentially useful objective function criteria used to quantify a system's physical balance capabilities: dynamic COM manipulability and velocity gains.

2.1 Biped Basics

Humanoid robots with rigid links can be modelled and analyzed as a multibody system: a set of rigid bodies, or links, which are connected by joints into a tree-like structure. These joints impose constraints on the relative movements between the connected links, such as a rotary joint preventing any motion which is not a rotation about the joint axis.

For systems which are not attached to a fixed inertial reference frame, such as a humanoid robot, an additional fictitious joint is used which defines the unconstrained motion of one of the links of the robot, called the floating base, relative to the fixed inertial frame.

In humanoid robots, the torso or pelvis link is typically chosen as the floating base link, as many multibody algorithms can be formulated to take advantage of this choice.

There are four standard sets of multibody algorithms, used to translate between relative joint variables and absolute (with respect to a fixed inertial reference frame) link variables:

Forward Kinematics uses the joint positions (joint angles, for rotary joints) to determine the absolute positions and orientations of the reference frames of the system's links;

Inverse Kinematics does the opposite, using the absolute positions and/or orientations of one or more links to determine the joint positions;

Forward Dynamics uses the joint forces (joint torques, for rotary joints) and the system's inertial properties to determine the resulting link accelerations; and

Inverse Dynamics (often labelled the "equations of motion") uses the accelerations and inertial properties of the links to find the joint forces.

The standard equations of motion for a fully actuated multibody system with n degrees of freedom (DOFs) can be written as $\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\vec{\boldsymbol{f}}_{ext})=\boldsymbol{\tau}$, where $\vec{\boldsymbol{f}}_{ext}$ is an aggregated 6D spatial vector of all external forces affecting the system and $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ and $\boldsymbol{\tau}$ are the n dimensional vectors of joint positions, velocities, accelerations, and forces, respectively. The $\boldsymbol{H}$ term is the $n\times n$ symmetric, positive-definite joint-space inertia matrix, while $\boldsymbol{c}$ is the n dimensional vector of generalized bias forces, which includes the forces due to Coriolis, centrifugal, and gravity effects (as well as the external forces in $\vec{\boldsymbol{f}}_{ext}$).

For the same fully actuated multibody system with a floating base, the equations of motion (using the spatial vector notation from [28, 29]) are given by

$$\begin{bmatrix} \tilde{\boldsymbol{I}}_0^c(\boldsymbol{q}) & \boldsymbol{F}(\boldsymbol{q}) \\ \boldsymbol{F}^T(\boldsymbol{q}) & \boldsymbol{H}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \vec{\boldsymbol{a}}_0 \\ \ddot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \vec{\boldsymbol{p}}_0^c(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\vec{\boldsymbol{f}}_{ext}) \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{\tau} \end{bmatrix} \tag{2.1}$$

where $\tilde{\boldsymbol{I}}_0^c(\boldsymbol{q})$ is the 6×6 Composite Rigid Body Inertia (CRBI) matrix, $\vec{\boldsymbol{a}}_0$ and $\vec{\boldsymbol{p}}_0^c(\boldsymbol{q},\dot{\boldsymbol{q}})$ are the 6 dimensional vectors of spatial acceleration and spatial bias forces for the floating base, respectively, and $\boldsymbol{F}(\boldsymbol{q})$ is a $6\times n$ matrix whose columns are the spatial forces required at the floating base to support unit accelerations about each of the actuated joints [28].

Efficient methods for determining $\boldsymbol{H}$ and $\boldsymbol{c}$ are readily available in the literature (including in [28]). Definitions of the floating base variables ($\tilde{\boldsymbol{I}}_0^c$, $\vec{\boldsymbol{p}}_0^c$, and $\vec{\boldsymbol{a}}_0$) and $\boldsymbol{F}$ can be found in Subsection 2.1.1, which also includes a description and definition of a number of other useful matrices and system properties which will be used throughout this work.

Following these definitions, a number of common reference points used in humanoid robotics are described in Subsection 2.1.2. Using these properties and points, a variety of simplified models are introduced in Subsection 2.1.3 which are frequently used both in the development of bipedal controllers and in the design of physical robots.

2.1.1 System Properties

There are a number of overall system properties in multibody dynamics which are useful for generating models, formulating algorithms, and designing controllers. The definitions below, and the notation used, are largely based on the book "Rigid Body Dynamics Algorithms" by Featherstone [28]. Further details on the spatial vector notation used can be found in Appendix A, based mainly on two introductory tutorials on the topic [30, 31].

The relevant properties used in this work are:

- the Center of Mass (COM), a weighted average of all of the links' COM positions;
- centroidal properties, measured relative to the COM (e.g., average spatial velocity);
- the kinetic energy of the system, in terms of joint and centroidal properties; and
- the inertia, Jacobian, and momentum matrices used to calculate these properties.

A fundamental property of any multibody system, which is particularly useful for those with a floating base such as a humanoid robot, is its Center of Mass (COM). This point is also sometimes referred to as its Center of Gravity (COG), although in this work the more common definition of the COG as the ground projection of the COM will be used.

Three types of reference frames will be used in this work: local link reference frames, a centroidal reference frame, and a fixed inertial reference frame at the origin ($\hat{O}$). The local frame for a given link is attached to the link, rotating and moving with the link, and is located at the center of the joint which connects the link to its parent (the link in the tree closer to the floating base). The centroidal reference frame has a fixed orientation, aligned with the fixed inertial frame, and is always instantaneously located at the COM ($\hat{C}$).

The location of $\hat{C}$ for a multibody system of N links (bodies) is given by the vector

$$\vec{c}(\boldsymbol{q}) = \frac{\sum_{i=1}^{N} m_i \vec{c}_i(\boldsymbol{q})}{\sum_{i=1}^{N} m_i} = \frac{\sum_{i=1}^{N} m_i \vec{c}_i(\boldsymbol{q})}{m} \tag{2.2}$$

where m_i is the mass of link i, $\vec{c}_i$ is the COM of link i, and m is the total mass of the system. The point $\hat{G}$, at the COG (the ground projection of the COM), is therefore defined as $\hat{G} := (x_C, y_C, h_G)$, where h_G is the height of the ground directly below the COM.

The 6D spatial velocity $\vec{\boldsymbol{v}}_i$ of link i can be expressed as the sum of the velocity $\vec{\boldsymbol{v}}_{p(i)}$ of its parent $p(i)$, and the velocity due to the movement of the n_i DOF joint connecting them: $\vec{\boldsymbol{v}}_i = \vec{\boldsymbol{v}}_{p(i)} + \boldsymbol{S}_i\dot{\boldsymbol{q}}_i$. Each joint's contribution is defined by $\dot{\boldsymbol{q}}_i$, a vector of $n_i \leq 6$ generalized joint velocities, and the motion subspace $\boldsymbol{S}_i$, a $6\times n_i$ matrix defining the joint axes.

A $6\times n$ spatial Link Jacobian $\boldsymbol{J}_i(\boldsymbol{q})$ can be defined which relates the ith link's spatial velocity $\vec{\boldsymbol{v}}_i$ to the system's ($n = \sum_{i=1}^{N} n_i$ dimensional) joint velocity vector $\dot{\boldsymbol{q}} = [\dot{\boldsymbol{q}}_1^T \ \ldots \ \dot{\boldsymbol{q}}_N^T]^T$

$$\vec{\boldsymbol{v}}_i = \vec{\boldsymbol{v}}_{p(i)} + \boldsymbol{S}_i\dot{\boldsymbol{q}}_i = \vec{\boldsymbol{v}}_{p(p(i))} + \boldsymbol{S}_{p(i)}\dot{\boldsymbol{q}}_{p(i)} + \boldsymbol{S}_i\dot{\boldsymbol{q}}_i = \boldsymbol{P}_i\boldsymbol{S}\dot{\boldsymbol{q}} =: \boldsymbol{J}_i\dot{\boldsymbol{q}} \tag{2.3}$$

where $\boldsymbol{S} = diag(\boldsymbol{S}_1, \ \ldots, \ \boldsymbol{S}_N)$ is the block diagonal $6N\times n$ System Motion Matrix and $\boldsymbol{P}_i = [\tilde{\boldsymbol{P}}_{i1} \ \ldots \ \tilde{\boldsymbol{P}}_{iN}]$ is a $6\times 6N$ selection matrix, where $\tilde{\boldsymbol{P}}_{ij}$ is a 6×6 identity matrix if $i = j$ or link j is above link i in the tree, and a 6×6 zero matrix otherwise.

By stacking all of the Link Jacobians together in order, the System Jacobian $\boldsymbol{J}$ can be constructed: $\boldsymbol{J} = [\boldsymbol{J}_1{}^T \ \ldots \ \boldsymbol{J}_N{}^T]^T$ [10]. The System Jacobian relates the system (link) velocity vector $\boldsymbol{v} = [\vec{\boldsymbol{v}}_1^T \ \ldots \ \vec{\boldsymbol{v}}_N^T]^T$ to $\dot{\boldsymbol{q}}$ for a particular multibody system: $\boldsymbol{v} = \boldsymbol{J}\dot{\boldsymbol{q}}$.

The spatial momentum of a given link is defined as $\vec{\boldsymbol{h}}_i = \tilde{\boldsymbol{I}}_i\vec{\boldsymbol{v}}_i$, where $\tilde{\boldsymbol{I}}_i$ is the link's spatial inertia (a 6×6 matrix which includes the link's mass m_i, 3D center of mass vector $\vec{c}_i$, and 3×3 rotational inertia matrix $\tilde{I}_i$). Using the spatial momentum equation, the block diagonal System Inertia matrix $\boldsymbol{I} = diag(\tilde{\boldsymbol{I}}_1, \ \ldots, \ \tilde{\boldsymbol{I}}_N)$ can also be defined, which relates the system momentum $\boldsymbol{h} = [\vec{\boldsymbol{h}}_1^T \ \ldots \ \vec{\boldsymbol{h}}_N^T]^T$ to the system velocity vector: $\boldsymbol{h} = \boldsymbol{I}\boldsymbol{v}$.

The Joint Space Inertia Matrix (JSIM) $\boldsymbol{H}$, from Equation (2.1), can be defined using the System Inertia and the System Jacobian matrices as $\boldsymbol{H} = \boldsymbol{J}^T\boldsymbol{I}\boldsymbol{J}$. This can quickly be shown using the two spatial definitions for the kinetic energy T of a multibody system:

$$T(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\sum_{i=1}^{N} \vec{\boldsymbol{v}}_i^T\tilde{\boldsymbol{I}}_i\vec{\boldsymbol{v}}_i = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{I}\boldsymbol{v} = \frac{1}{2}(\boldsymbol{J}\dot{\boldsymbol{q}})^T\boldsymbol{I}(\boldsymbol{J}\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{J}^T\boldsymbol{I}\boldsymbol{J}\dot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{H}\dot{\boldsymbol{q}} \tag{2.4}$$

Similar to its use relating joint accelerations to forces, the JSIM can also be used to relate the joint velocities to the joint momenta $\boldsymbol{h}_q$ (also called the generalized or canonical momenta [10]): $\boldsymbol{h}_q = \boldsymbol{H}\dot{\boldsymbol{q}}$. Although it has previously been used in other aspects of robotics, joint momentum has only recently been used in the control of humanoids [32].

An efficient method for calculating $\boldsymbol{H}$ has been developed, called the Composite Rigid Body Algorithm [28]. This algorithm makes use of a set of 6×6 matrices called the Composite Rigid Body Inertias (CRBIs), labelled $\tilde{\boldsymbol{I}}_i^c$. If two bodies are rigidly attached they will have an identical spatial velocity, therefore based on the definition of spatial momentum their combined spatial inertia is simply the sum of their spatial inertias.

Extending this thought, the CRBI for a given joint is defined as the sum of the spatial inertias of all of the links below the joint in the kinematic tree. This leads to the definition of the floating base CRBI ($\tilde{\boldsymbol{I}}_0^c$) from Equation (2.1) as $\tilde{\boldsymbol{I}}_0^c = \sum_{i=1}^{N} \tilde{\boldsymbol{I}}_i$ (note that this is the sum of the individual links' spatial inertias, not the sum of their CRBIs). It also allows us to define the matrix of spatial forces $\boldsymbol{F}$ which connects the fully actuated multibody system with the floating base joint dynamics in Equation (2.1): $\boldsymbol{F} = [\tilde{\boldsymbol{I}}_1^c \boldsymbol{S}_1 \ \ldots \ \tilde{\boldsymbol{I}}_N^c \boldsymbol{S}_N]$.

The total momentum of a closed system is a conserved quantity (it will remain constant), and therefore the total momentum $\vec{\boldsymbol{h}}$ of a system is simply the sum of individual link momenta: $\vec{\boldsymbol{h}} = \sum_{i=1}^{N} \vec{\boldsymbol{h}}_i$. In practice, the total momentum $\vec{\boldsymbol{h}}$ is typically found by summing coordinate vectors for each link's momentum referenced to a common frame. For example, by referencing each link's momentum to the centroidal reference frame, the centroidal momentum $\vec{\boldsymbol{h}}_C$ can be found: $\vec{\boldsymbol{h}}_C = \sum_{i=1}^{N} {}^C\vec{\boldsymbol{h}}_i$ [10, 33].

The Centroidal Inertia Matrix (CIM), also sometimes known as the Centroidal Composite Rigid Body Inertia (CCRBI) [34], is an Operational Space Inertia Matrix (OSIM) [10] with its end effector set to the system's COM $\hat{C}$. The CIM $\tilde{\boldsymbol{I}}_C$ relates the average spatial velocity $\vec{\boldsymbol{v}}_C$ of the system to the centroidal momentum: $\vec{\boldsymbol{h}}_C = \tilde{\boldsymbol{I}}_C \vec{\boldsymbol{v}}_C$.

Using the centroidal reference frame, the CIM can be defined using a 6×6 block diagonal matrix with two 3×3 blocks (where $\tilde{0}$ is the 3×3 zero matrix):

$$\tilde{\boldsymbol{I}}_C = \begin{bmatrix} \tilde{I}_C & \tilde{0} \\ \tilde{0} & m\tilde{1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} {}^C\tilde{I}_i & \tilde{0} \\ \tilde{0} & m\tilde{1} \end{bmatrix} \tag{2.5}$$

where ${}^C\tilde{I}_i$ is the standard 3×3 rotational inertia matrix of each link ($\tilde{I}_i$) projected to the centroidal frame (at $\hat{C}$), m is the total mass, and $\tilde{1}$ is a 3×3 identity matrix.

As with any spatial coordinate vector, $\vec{\boldsymbol{v}}_C$ can be separated into linear and angular components: $\vec{\boldsymbol{v}}_C = [\vec{\omega}_C^T \ \vec{v}_C^T]^T$. The vector $\vec{v}_C$ is the linear velocity of the COM, while $\vec{\omega}_C$ can be thought of as the average angular velocity of the entire system [10].

The system and centroidal momentum vectors can also be found using a pair of momentum matrices, which relate each momentum vector to the system joint velocity vector [10,33]. The System Momentum Matrix $\boldsymbol{A}$ relates the system momentum directly to the system's joint velocity vector: $\boldsymbol{h} = \boldsymbol{A}\dot{\boldsymbol{q}}$. Using the expressions for the System Inertia and System Jacobian matrices, it is easy to show that $\boldsymbol{A} = \boldsymbol{I}\boldsymbol{J}$ (and by extension $\boldsymbol{H} = \boldsymbol{J}^T\boldsymbol{A}$):

$$\boldsymbol{h} = \boldsymbol{I}\boldsymbol{v} = \boldsymbol{I}\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{A}\dot{\boldsymbol{q}} \tag{2.6}$$

The Centroidal Momentum Matrix (CMM) ${}^C\boldsymbol{A}$ relates the centroidal momentum $\vec{\boldsymbol{h}}_C$

to the generalized velocities of a system: $\vec{\boldsymbol{h}}_C = {}^C\boldsymbol{A}\dot{\boldsymbol{q}}$ [33]. The structure and properties of the CMM, as well as methods for its calculation, are discussed further in [10, 33].

The simplest method for defining the spatial momentum $\vec{\boldsymbol{h}}_i$ of a rigid body is to use the local reference frame to generate the coordinate vector $\vec{\boldsymbol{h}}_i = \tilde{\boldsymbol{I}}_i\vec{\boldsymbol{v}}_i$. These vectors can then be transformed using a force transformation matrix and summed to find $\vec{\boldsymbol{h}}_C$:

$$\vec{\boldsymbol{h}}_C = \sum_{i=1}^{N} {}^C\vec{\boldsymbol{h}}_i = \sum_{i=1}^{N} {}^C\tilde{\boldsymbol{X}}_i^*\vec{\boldsymbol{h}}_i = {}^C\boldsymbol{X}\boldsymbol{h} \tag{2.7}$$

where the centroidal transformation matrix is given by ${}^C\boldsymbol{X} = \begin{bmatrix} {}^C\tilde{\boldsymbol{X}}_1^* & \dots & {}^C\tilde{\boldsymbol{X}}_N^* \end{bmatrix}$.

Using this definition for $\vec{\boldsymbol{h}}_C$, it can be easily shown that ${}^C\boldsymbol{A} = {}^C\boldsymbol{X}\boldsymbol{A} = {}^C\boldsymbol{X}\boldsymbol{I}\boldsymbol{J}$:

$$\vec{\boldsymbol{h}}_C = {}^C\boldsymbol{X}\boldsymbol{h} = {}^C\boldsymbol{X}\boldsymbol{A}\dot{\boldsymbol{q}} = {}^C\boldsymbol{X}\boldsymbol{I}\boldsymbol{J}\dot{\boldsymbol{q}} = {}^C\boldsymbol{A}\dot{\boldsymbol{q}} \tag{2.8}$$

For further details on how these vectors and matrices are defined or the notation used, see Appendix A, based on [30, 31], or see [10, 28, 29, 33].

2.1.2 Reference Points

There are a number of useful state-dependent reference points which have been defined for bipedal robots, generally used for control and postural stability assessment. The classic reference points in bipedal research are the Center of Mass (COM), Center of Pressure (COP), the Zero Moment Point (ZMP) [35], and the Centroidal Moment Pivot (CMP) point [36]. Some of these points are also used to define key points for the simplified models discussed later in this paper, so they will be briefly introduced below.

The Centroidal Moment Pivot (CMP) point was originally proposed by two different groups at ICRA 2004: Popovic, Hofmann, and Herr defined it as the Zero Spin Center of Pressure [37], while Goswami and Kallem called it the Zero Rate of Angular Momentum point [38]. Following this parallel effort, the authors collaboratively defined the point as the CMP in [36], and compared it to both the COP and ZMP (shown in Figure 2.1).

A standard tool in the analysis of the external forces on a system of rigid bodies is to find a single equivalent resultant force which produces the same net effect on the system as the original set of forces. The CMP is defined as the point on the ground through which the resultant external force would have to act to produce no horizontal torque about the

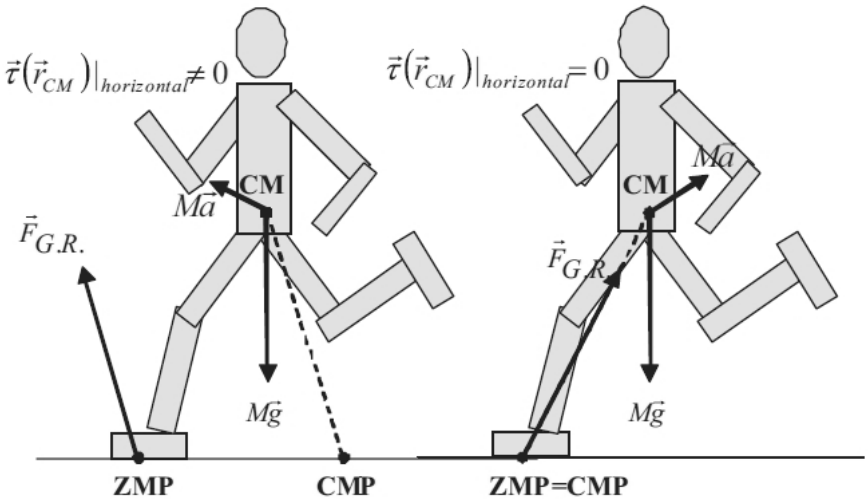


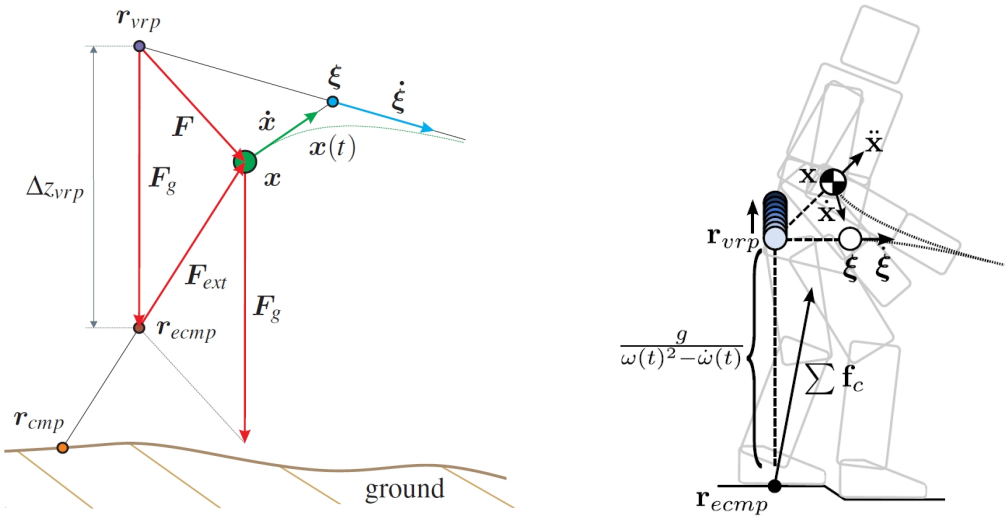
Figure 2.1: Illustrations of the Zero Moment Point (ZMP) and Centroidal Moment Pivot (CMP) point from [36]. In this figure, **CM** is the COM, M is the total mass, $\vec{F}_{G.R.}$ is the external ("Ground Reaction") force, $\vec{g}$ is the acceleration due to gravity, and $\vec{a}$ is the acceleration of the COM. Since these diagrams are on horizontal ground, the COP is identical to the ZMP, so is unlabeled. On the left, the line of action of the external force does not point directly at the COM, leading to a non-zero torque about the COM, so the CMP and the ZMP are in different locations. On the right, the CMP and ZMP are coincident, due to the external force acting directly towards the COM and therefore producing no torque about the COM.

COM, or equivalently as the projection of the COM onto the ground along a line parallel to the external force vector's line of action. This is illustrated in Figure 2.1.

For legged robots, the resultant external force vector always passes through the support polygon, and is often called the ground reaction force as it is typically due to the unilateral (pushing) contacts of the feet with the ground. The support polygon (or volume) of a legged robot is defined as the shape enclosed by the convex hull connecting every contact point between a robot and its environment [39]. The intersection of the external force vector with the support polygon is known as the Center of Pressure (COP).

One of the most well known reference points is the Zero Moment Point (ZMP), defined as the point on the ground at which the resultant external force would produce no horizontal torque [35]. If a biped is walking with flat feet on horizontal ground and the feet are the only contacts, the ZMP is equivalent to the COP [36, 40] (see Figure 2.1).

When these restrictions are not met, based on their common definition as a point through which the external resultant force acts, both the ZMP and the COP must be located along the line of action of the resultant external force. Since the COP is, by definition, always located within the support polygon, the ZMP is effectively the projection



(a) DCM with Constant Δz [11]

(b) DCM with Varying Δz [13]

Figure 2.2: Illustrations of the Divergent Component of Motion (DCM). In (a), the vertical offset between the eCMP and the VRP is constant, while in (b) it varies with time. In these figures, x is the position of the COM and ξ is the position of the DCM. Dotted variables represent derivatives with respect to time.

of the COP onto the ground along the resultant external force vector [36].

The final reference point used in this thesis, known as the Divergent Component of Motion (DCM) [41] (or Extrapolated COM [42]), has been used to analyze the dynamics of bipeds moving in 3D on non-horizontal ground [11, 13, 43].

The DCM ($\hat{\xi}$) is defined as $\hat{\xi} = \hat{C} + b\vec{v}_C$, as shown in Figure 2.2. In [11], b is defined as the time constant of the DCM dynamics, given by $b = \sqrt{\Delta z_{vrp}/g}$. The value Δz_{vrp} is heuristically defined as the (estimated) average height of the COM above the ground, and defines the vertical offset between two more new reference points: the Enhanced CMP (eCMP) and the Virtual Repellent Point (VRP).

Mathematically, the VRP and eCMP are defined by the total and external (contact) forces on a biped (see Figure 2.2a), respectively, using a simple force-to-point translation based on b [11]. However, they can both also be defined using a more intuitive approach by comparing the eCMP to the COP and the VRP to the DCM.

Similar to how the ZMP is the projection of the COP onto the ground along the external force vector, the CMP is the projection of the eCMP onto the ground along a line passing through the COM which is parallel to the external force vector. Further, the intuitive definition of the DCM as a point which the COM is constantly moving towards is echoed

for the VRP, by observing that the COM will always be accelerating away from the VRP and that the DCM will therefore always move directly away from the VRP.

Englsberger *et al.* used these points to generate desired DCM trajectories from an initial footstep plan, by assuming a constant angular momentum and ignoring impact. These DCM trajectories were used as control references for a DCM tracking controller, along with a heuristic balance point estimator which determines an updated step position based on the existing footstep plan, predicted DCM location at impact, and the nearby terrain [11]. Recently, Englsberger *et al.* used their DCM and CMP based trajectory generation with an analytical footstep adjustment for recovery from strong disturbances [44]. They have also developed a gait generator based on the DCM which is capable of generating dynamic walking across moving support surfaces [45].

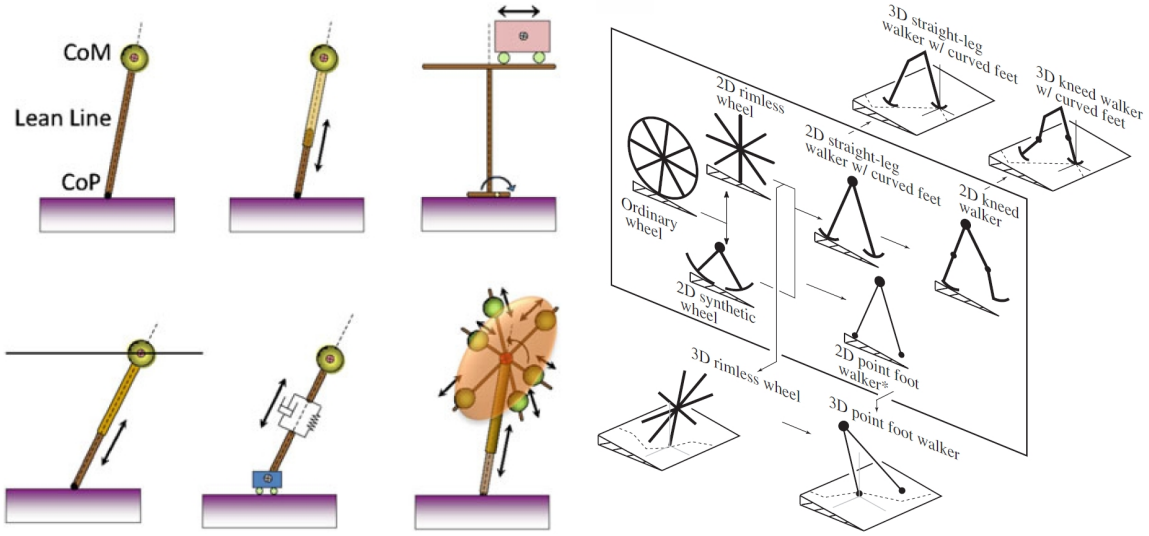
Hopkins *et al.* have developed an extension of the DCM, by defining the natural frequency $\omega = 1/b$ of the DCM dynamics as a function of time (see Figure 2.2b). Since the rest of the formulation consists only of constants, this effectively allows the use of a dynamic vertical offset between the eCMP and the VRP. In [13,43], they use this extension in tandem with centroidal momentum to generate a whole-body controller.

2.1.3 Simplified Models

Although there have been developments in the application of full-body dynamic models for balancing (for examples, see Section 2.3 and [29]), there is still value in working with simplified models. In particular, several of these simpler models enable an intuitive understanding of balance and gait for bipeds which facilitates both their mathematical expression and the qualitative description of the associated algorithms and controllers.

The simplest dynamic model of a humanoid robot consists of an inverted pendulum (IP), with a point mass located at the COM, a point foot located at the COP, and a massless rigid leg between the COM and foot (see Figure 2.3a). A common extension of this model is the telescoping IP, where a prismatic joint is added to the leg to allow changes in the distance between the COM and the COP. Due to the way the COP is defined, the aggregate of the external contact forces can be replaced with a single linear force acting along the leg connecting the COP to the COM and a moment about the point foot. As with any pendulum, the dynamics of these IP models are inherently non-linear.

Another model is the rimless spoked wheel (see Figure 2.3b), which is effectively an IP with a number of massless legs, extending at regular intervals from the common point mass within a plane [47, 48]. This is also a non-linear model, but has the advantage that it can predict and/or control future step locations by manipulating the angle between the



(a) Inverted Pendulum Models [10] (b) Passive Biped Models [46]

Figure 2.3: Illustrations of common simplified biped models. The models shown in (a) are (from left to right, top to bottom): Rigid Inverted Pendulum (IP), Telescoping IP, Cart Table, Linear IP, Variable Impedance IP, and the Reaction Mass Pendulum [10].

spokes (legs). A similar model is the compass gait biped, consisting of two massless legs connected together with a rotary joint at the common COM point mass [49]. This model has the benefits of the spoked wheel model combined with a clear method for manipulating the angle between the legs, without the need for any additional legs.

The cart table model is an alternative to the above IP-based models, where the COM is modelled as a frictionless cart running along a horizontal table and the "foot" of the table has a non-zero contact area [50]. One of the main advantages of this model is that it is inherently linear, leading to simpler equations and controllers. The cart table model is often used when discussing the ZMP of a humanoid, as the size of the foot and the location of the ZMP within the foot is a critical aspect of most ZMP controllers.

A simplification of the telescoping IP model is the Linear IP model (often called the 3D LIPM, or just LIPM), which has both the advantages of a simple IP structure and linear dynamics [51, 52]. Similar to the cart table model, the linearity is imposed due to a restriction placed on the COM that it remains on a specific plane, usually a horizontal one as this means that the COM of the biped model remains at a fixed height. Although it is typically used with a restriction that the COM must remain at a fixed height, the model itself only confines the COM to a plane, which allows movement over stairs and

other sloped surfaces. To model the LIPM's dynamics, the leg force required to maintain the COM on the given plane can be represented using a preloaded non-linear spring [17].

Another form of IP model which restricts the COM motion is the Prismatic IP model (PIPM), where the lateral and saggital single contact dynamics are independent ordinary differential equations [53]. Similar to how the LIPM restricts COM movement to a plane, the COM in the PIPM is restricted to motion along a non-planar, piecewise linear surface.

The Angular Momentum inducing IP model (AMPM) [54, 55], a LIPM model with a rotational inertia, enables an explicit representation of the angular momentum of the biped. This extension allowed much better control over the overall motion of the biped, as it could inherently encode the linear and angular motions of the overall system.

Building on the benefits of the AMPM, the Reaction Mass Pendulum (RMP) model was introduced by Lee and Goswami [34], compared to the compass gait biped in [49], and was examined thoroughly from a dynamics and stability perspective in [56]. This model consists of a modification of the telescoping IP model, by breaking the point mass into three pairs of equal proof masses which can rotate about the COM position. This enables the model to take into account both the angular momentum and changes in rotational inertia, which is required to allow full control of the momentum and other inertial effects. This modification could potentially be applied to any of the above models which use a point mass at the COM, alongside the existing assumptions.

All of these models can also be extended with additional features, to create double pendulums or multibody systems with distributed masses, which is sometimes used in the generation of balance strategies (see Figure 2.3). However, the additional links and masses remove many of the benefits of these simplified models, existing on a continuum between the above simplified models and the full dynamics of the biped.

Finally, a newer model which is being used in most state of the art controllers is the Centroidal Dynamic model [10]. With this model, shown in Figure 2.4, the various centroidal properties defined in Subsection 2.1.1 are used directly as a model of the overall dynamics of the system. Effectively, the biped is assumed to be a single rigid body, with its inertia equal to the centroidal inertia, its velocity equal to the average spatial velocity, and all external forces projected to the centroidal reference frame.

2.2 Balance Point Estimators

There are two high-level categories of balance point estimation methods: those which consider the loss of energy due to the impact of the landing foot with the ground, and those

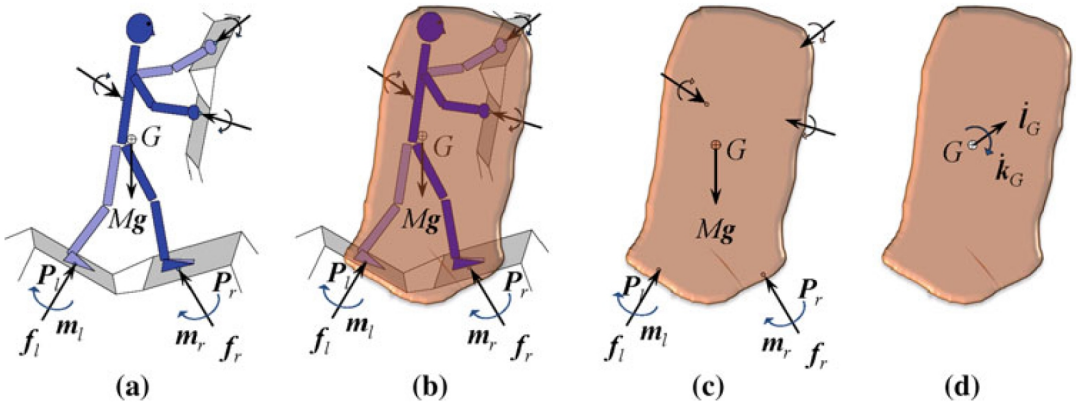


Figure 2.4: Illustration of the Centroidal Dynamic model from [10]. In this figure, G is the COM.

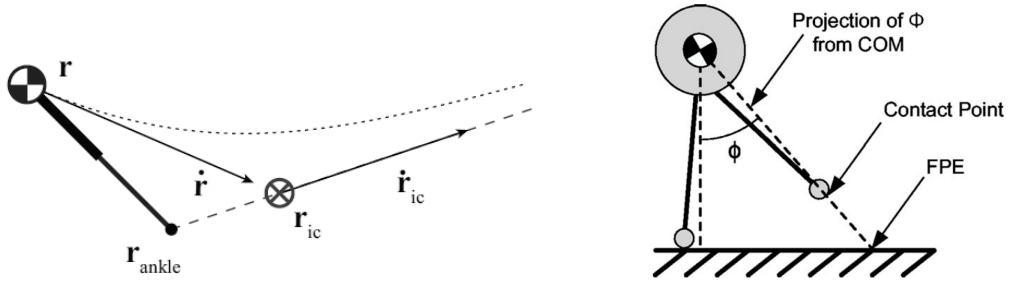
that do not. In the majority of methods that do not consider impact losses, the LIPM is used so vertical motion of the COM is also ignored.

The methods which account for impact-based energy loss follow a common formulation, although the specifics and model used to find the balance point can vary [14, 16, 18–20]. Given the (possibly predicted) model state at the time just prior to impact, conservation of angular momentum about the point of impact and conservation of energy after impact are used to determine where to step with a rigid leg. Energy is lost at impact, based on an instantaneous reduction in COM velocity due to the conservation of angular momentum, and then converted from kinetic energy to potential energy after impact via the rigid leg.

Similarly, methods which ignore the loss of energy at impact also have a common formulation, again with varying specifics and models [15–17, 57]. Given the (possibly predicted) model state at the time just prior to stepping, conservation of energy and constraints on the COM motion are used to determine where to step. In most of these methods, the COM is constrained to a fixed height and the orbital energy is conserved via a horizontal force which is proportional to the horizontal distance between the COM and the COP.

The basic idea behind the two approaches can be examined by comparing an exemplar method from each category (see Figure 2.5): the Instantaneous Capture Point method [17], for the impact-free group; and the Foot Placement Estimator [18], for the impact group. Since they are both balance point estimators, both methods have identical goals: determining where to step so that the biped can come to rest with the COM directly above the balance point. They are also both only defined for horizontal, flat ground, and use simplified models of the biped to determine their different balance points.

The Instantaneous Capture Point (ICP) method [17] uses a LIPM and the conservation



(a) Illustration of the ICP [15].

(b) Illustration of the FPE [18].

Figure 2.5: Illustrations of the Instantaneous Capture Point (ICP) and the Foot Placement Estimator (FPE). The ICP in (a) is drawn in the horizontal plane, while the FPE in (b) is drawn in a vertical plane. In (a), $\mathbf{r}$ is the COM, $\mathbf{r}_{\text{ankle}}$ is the COP, and $\mathbf{r}_{\text{ic}}$ is the ICP.

of orbital energy to determine where to instantaneously step to come to rest (i.e., the balance point). Although the original definition of the ICP in [17] augmented the LIPM with a flywheel to account for rotational inertia, the vast majority of later work ignores the inertia and just uses the standard LIPM (with a point mass, as shown in Figure 2.5a).

Using the standard LIPM, the location of the ICP (labelled $\hat{\xi}_{xy}$) is simply defined as

$$\hat{\xi}_{xy} = \hat{G} + \sqrt{h/g}\ \begin{bmatrix}\dot{x}_C & \dot{y}_C & 0\end{bmatrix}^T \tag{2.9}$$

where $h = z_C - z_G$ is the current height of the COM above the ground.

The Foot Placement Estimator (FPE) [18], the original impact-based method, was developed based on a non-linear analysis of a planar biped model with a fixed rotational inertia and massless, fixed length legs (as shown in Figure 2.5b). However, the actual formulation of the FPE point removes one of the legs (producing a reaction wheel pendulum) and relaxes the model assumptions by allowing for the leg length to be defined using the angle ϕ and the height of the COM. It uses conservation of angular momentum about the balance point to determine the energy lost during impact, then conservation of the post-impact energy to determine where to step.

Trigonometry can be applied in the vertical plane to find the FPE point $\hat{F}$ on the ground, using the angle ϕ at which the model must instantaneously place its point foot: $\hat{F} := \hat{G} + h\tan(\phi)$. Therefore, the angle ϕ must be found, using conservation of energy and momentum and by defining the length l of the model's leg using $l = h/\cos\phi$:

$$\frac{\Big[mh\cos\phi(\dot{h}\sin\phi + v\cos\phi) + I_C\omega\cos^2\phi\Big]^2}{2mgh\cos\phi(1-\cos\phi)(mh^2 + I_C\cos^2\phi)} = 0 \tag{2.10}$$

where h is the height of the COM, $\dot{h}$, v, and ω are the vertical, horizontal, and angular (about $\hat{G}$) velocity of the COM, respectively, and I_C is the planar rotational inertia.

The loss of energy at impact due to the FPE model's rigid leg, compared to the loss of energy due to compression of the non-linear spring in the LIPM leg used in the ICP, is the key differentiator between these methods. This can be used to show that, when both models' assumptions hold, $\hat{F}$ is the closest point in a continuous linear set of balance points, which includes $\hat{\xi}_{xy}$, where a point's position in the set depends mainly on the model's leg stiffness [20]. Alternatively, since the rigid leg of the FPE model causes the COM to move in a vertical arc and the LIPM causes the COM to move in a horizontal line, the actual balance point for the typical flattened arc of bipedal gait would likely fall somewhere between the balance points found using the FPE and ICP methods [58].

The next two subsections describe a number of extensions to the ICP and FPE methods for estimating balance points. These extensions mainly aim to reduce the somewhat limiting assumptions made by the ICP and FPE formulations: planar COM motion (horizontal plane for the ICP, vertical plane for the FPE) and instantaneous swing foot movement.

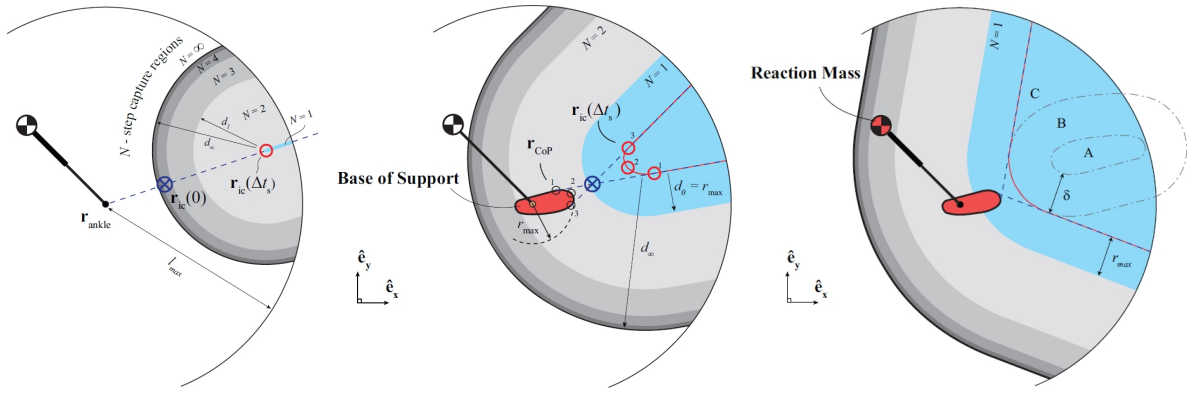
2.2.1 ICP Extensions

The most often-used extension of the ICP is Capturability, which defines being N-Step Capturable as the ability of a biped to come to rest in N steps [15]. By this definition, if a biped is 0-Step Capturable then it does not need to take a step, and can use other balancing strategies (e.g., movement of the COP/CMP within the existing support polygon) to come to rest. Otherwise, at least one step is needed, where the number of steps needed depends on where the first step is placed and the minimum time required to take a step.

It is easy to show that for the classic LIPM, with a point foot on the ground and a point mass at the COM, the point foot must be placed at the ICP for the model to be 0-Step Capturable. However, in [15] the LIPM model is also progressively augmented, first with a finite-sized foot and then both a finite foot and a fixed rotational inertia.

The finite sized foot allows the model to be 0-Step Capturable provided the ICP is anywhere within the finite contact area of the foot, as it assumes that the COP can be moved freely within the foot. The reaction mass (a fixed rotational inertia) adds a lunging capability, which allows the ICP to be brought back inside the contact area if it is just outside the foot's edge, increasing the size of the 0-Step Capture Region.

If the biped is not 0-Step Capturable, then a set of Capture Regions can be found on the ground, where stepping into the N-Step Capture Region will cause the biped to become



(a) LIPM (b) LIPM with Foot (c) LIPM with Foot and Inertia

Figure 2.6: Illustrations of the N-Step Capture Regions [15]. The 1-Step Capture Region in each figure is blue, while the remaining N-Step Capture Regions are progressively darker shades of grey. Models shown are (a) a LIPM, (b) a LIPM with a finite sized foot, and (c) a LIPM with a finite sized foot and a fixed rotational inertia. All figures are drawn in the horizontal plane.

(N-1)-Step Capturable (see Figure 2.6). For the classic LIPM model, the 1-Step Capture Region is simply a line pointing away from the model's point foot, starting at the predicted ICP location after a minimum swing time has passed (see Figure 2.6a).

For the other models, the 1-Step Capture Region is found using the size of the foot and the possible predicted ICP locations, which appear as a scaled, mirrored image of the possible COP locations in the foot projected through the ICP. For all models, each N-Step Capture Region starting at N=2 is then recursively defined as the region within d_{N-1} of the (N-1) Capture Region, with d_N defined as a function of d_{N-1} with exponential decay.

A planar extension to the ICP was developed by Ramos and Hauser [57], in which the LIPM constraints are loosened to allow for sloped linear motion or parabolic motion (labelled a "Nonlinear Inverted Pendulum" model) and the terrain is defined as a set of piecewise linear segments. Since this eliminates the ICP method's closed-form equations, a shooting approach using bisection is applied to successively define equations to find: a balance point on the (possibly uneven) terrain given a parabolic COM path, coefficients of a parabolic COM path given a balance point location, and finally an enumeration of candidate pairs of COM paths and balance points distributed across the given terrain.

Although this method expands the possible applications of the original ICP approach, it still limits the COM motion and has only been developed in a planar environment. The use of a shooting approach also requires repeated calculations for every location and COM path, leading to the potential for massive computational requirements if this approach were

extended to 3D. Since push recovery in particular demands very fast reactions to external disturbances, this computational load may be too high for many bipeds to be able to find an acceptable balance point in time to regain balance before falling.

Finally, the DCM has been described as the 3D analog of the ICP: When the ICP assumptions are satisfied (horizontal ground and COM motion), the ground projection of the DCM is equal to the ICP [11]. The DCM also incorporates only state-based information and makes fewer assumptions about the model or the environment.

2.2.2 FPE Extensions

There are a number of extensions to the FPE, both for planar robots [16, 19] and in 3D [14, 20, 59, 60]. Currently, all of the 3D extensions to the FPE consist of choosing a vertical plane, projecting the dynamics of the biped into the chosen plane, then applying the FPE equations to find a balance point [14, 20, 59, 60].

The first planar extension of the FPE was the Foot Placement Indicator (FPI), which replaced the reaction wheel model of the FPE with a planar multilink model of the robot with distributed masses before calculating the restorative balance point [19]. By using a more realistic model of the distributed masses of the robot, particularly in the swing leg, the FPI gains accuracy in exchange for a loss in computational efficiency.

Li *et al.* compared the ICP (in a vertical plane) to a predictive version of the ICP (which we have labeled the Predictive Capture Point, or PCP) and to a predictive balance point estimator using a simple inverted pendulum (IP) model for one-step balance recovery [16]. Prediction for both balance point estimators was based on conservation of energy leading up to stepping and a heuristically chosen desired stance leg angle at step time. With some rearrangement of the equations, the IP-based estimator is effectively a predictive form of the FPE, with the stipulation that the stance and swing legs can have different lengths, so will be labeled the Predictive FPE (PFPE).

Although originally designed and applied in a planar environment, the FPE has been used directly in a 3D environment by projecting the robot's dynamics onto either a constant [59,60] or changing [14,20] vertical plane. By projecting the biped's dynamics into a specific vertical plane (typically passing through the COM), and applying the FPE equations, a balance point can be found along the intersection of the plane and the ground which balances the robot in the chosen plane. Both the FPI and the PFPE are defined in a vertical plane, similar to the original FPE, and could also benefit from projection into 3D.

In both [59] and [60], the vertical projection plane is chosen manually and constant, which keeps the computational cost down (in exchange for requiring separate out-of-plane

balance in [60]). However, for the Generalized FPE (GFPE) in [14] and the 3D FPE in [20], the vertical projection plane always passes through the COM, but its orientation is calculated online in an attempt to include as much of the robot's 3D dynamics as possible.

The 3D FPE [20] applies the original FPE in a vertical plane (containing the COM) perpendicular to the horizontal components of $\vec{k}_G$, the biped's angular momentum about the COG. When using the 3D FPE, the biped's full 3D dynamics are projected into the assumed (vertical) impact plane to determine its equivalent planar rotational inertia and average angular velocity. By basing the vertical plane's orientation on the angular momentum about the COG, it directly takes into account both the linear velocity of the COM and its centroidal angular momentum.

This method was developed using the "3D Euler Pendulum" model, a monoped with a disc foot which is equivalent to the reaction wheel biped model from the original FPE paper when its motion is restricted to a vertical plane [20]. Similar to the original FPE, the 3D FPE formulation relaxes the fixed leg length assumption, so the biped need not be at rest nor in contact with the ground before impact. However, as a direct extension of the FPE into 3D, it is only defined for horizontal ground surfaces and is not predictive, requiring continuous calculation until impact occurs.

The GFPE [14], on the other hand, uses a 2D point foot walker (which they have labelled a rimless wheel) to predict a recovery step location, based on the velocity of the COM after a biped is pushed while at rest. The model is embedded in a vertical plane (containing the COM) parallel to the COM velocity ($\vec{v}_C$) and uses a modified version of the original planar FPE equations for step prediction.

The model's stance foot location is placed at the orthogonal projection of the Center of Pressure (COP) into this vertical plane, which for predictive purposes is assumed to be a point about which the biped is in pure planar rotation until impact. The GFPE can also be applied to piecewise planar ground surfaces, using an elevation angle between the stance foot and the desired landing foot location. However, this approach neglects the rotational inertia of the biped, and assumes that the biped is at rest before a disturbance.

2.2.3 Comparison

To support both balance and gait, the ideal balance point estimator would predict a location to step in 3D to achieve a desired final state based on the current state of the biped, without using any heuristic parameters, and should include the effects of impact, a finite sized foot, centroidal inertia, and realistic 3D COM motion over 3D terrain. Table 2.1 gives an overview of the most well known existing balance point estimators.

Table 2.1: Comparison between Balance Points

The 3D column indicates whether the balance point is defined in 3D and allows 3D (non-flat) terrain. In this table, the following variables are used: $\boldsymbol{\tau}_a$ for ankle torque, $\tilde{I}_C$ for centroidal inertia, $\dot{z}_C$ for vertical COM velocity, θ_- for the pre-impact angle of the pendulum model, t_s for the minimum swing time of the swing leg, and Δz for the height of the VRP above the eCMP. An asterix is used to indicate the following special cases: when using the ICP, $\tilde{I}_C$ is typically ignored although it was part of the original formulation; the 3D FPE model is defined in 3D, but the 3D FPE itself is only defined for flat terrain; adding $\tilde{I}_C$ to the GFPE formulation is trivial; and the GFPE is defined for piecewise flat terrain, not 3D terrain.

Balance Point	**Ref.**	**Predictive**	$\boldsymbol{\tau}_a \neq 0$	$\tilde{I}_C \neq 0$	$\dot{z}_C \neq 0$	**Impact**	**3D**	**Notes**
ICP	[17]	✗	N/A	✓*	✗	✗	✗	
FPE	[18]	✗	N/A	✓	✓	✓	✗	
FPI	[19]	✗	N/A	✓	✓	✓	✗	Distributed mass
3D FPE	[20]	✗	N/A	✓	✓	✓	✗*	
Predictive CP	[16]	✓	✗	✗	✗	✗	✗	Needs θ_-
Predictive FPE	[16]	✓	✗	✗	✓	✓	✗	Needs θ_-
Capturability	[15]	✓	✗	✗	✗	✗	✗	Needs t_s
w/ Finite Foot	[15]	✓	✓	✗	✗	✗	✗	Needs t_s
w/ Reaction Mass	[15]	✓	✓	✓	✗	✗	✗	Needs t_s
DCM-based	[11]	✓	✓	✓	✓	✗	✓	Needs Δz
GFPE	[14]	✓	✓	✗*	✓	✓	✓*	

The most successful step-based balance strategies use balance point estimators which include prediction, such as Capturability [15]. The ability to predict a stepping location allows the controller to optimize its control efforts towards achieving useful stepping behaviors. Without prediction, very fast control loops are required to track an instantaneously moving point and there are no guarantees that the desired stepping point will end up within the feasible motion range of the swinging leg's foot.

The inclusion of ankle torques, or equivalently a finite sized foot, allows the COP to be shifted within the support polygon, in some cases obviating the need to take a step. For instantaneous balance points, this is not typically used as they do not make any assumptions as to the future behavior of the COP, so can decide whether to step based on if the generated balance point is within the existing support polygon. However, with predictive methods the ability to move the COP around within the support polygon must be considered, and therefore ankle torques should be included.

Allowing the COM to move freely in 3D over 3D terrain and including centroidal inertia are useful features, as they relax the common assumptions of horizontal COM motion over flat ground and centroidal angular momentum regulation. When these assumptions are used, they simplify the equations needed to find balance points, but real bipeds and terrain often violate these assumptions. In addition, the ability to modify the COM height and the centroidal inertia for the purposes of balance are effective strategies for compensating for some disturbances, so should not be neglected in any part of a balancing system.

Since the majority of the balance point estimators depend on the conservation of energy, considering the effects of impact is a critical feature, which is often ignored to allow the use of simpler linear models of bipedal motion. Not only does the energy lost to impact directly effect the location of the generated balance point, it can also be used to determine the required energy input to maintain a constant desired gait velocity.

In fact, although many balance points exist, none were found which could fully satisfy these requirements, as shown in Table 2.1. Since the various existing balance point estimators make different assumptions but use similar approaches, there is the potential that a more general method can be found which encapsulates and generalizes existing methods.

This is partially inspired by [20], which determined that the stiffness of the chosen model's leg was the critical difference between the FPE point and the ICP, and that a continuum of suitable balance points existed along a line between the two named points. Therefore, a novel balance point estimator is needed that satisfies all of the requirements of this research (including the features listed in Table 2.1), and which generalizes and encapsulates most of the existing 2D and 3D balance points.

2.3 Balance for Bipeds

The three basic humanoid balance strategies, as described in [5], are: ankle strategy, hip strategy, and stepping out. The ankle strategy, also called COP balancing, is used to keep the COP within the support polygon (SP) by applying ankle torques. The hip strategy, also called CMP balancing, uses centroidal torques to move the CMP relative to the COP, while also applying COP balancing to keep the COP in the SP. In [61], the ankle and hip strategies were extended by adding a knee joint to the model used in [5], effectively changing the LIPM model back into a non-linear telescoping IP.

More generally, there are three main types of balance strategies which correspond to the basic strategies described in [5], labelled as: flexing, leaning, and stepping. Flexing strategies make use of joint compliance to deal with small disturbances, leaning strategies

use joint movements to deal with moderate disturbances, and stepping strategies use one or more steps to deal with large disturbances. Each of these types of strategies is discussed at a high level in the next three subsections, with an emphasis on describing the benefits and drawbacks of each type of strategy when applied independently.

Briefly, flexing is relatively simple and fast, but its effectiveness is limited by the strength and duration of the disturbance and requires at least minor mechatronic design changes to position-controlled joints. Leaning uses body movements (e.g., torso leaning, swinging arms, etc) to manipulate the system's centroidal dynamics, which allows it to handle moderate disturbances but is slower than flexing. Stepping can handle large disturbances, but requires swinging a leg to a desired point on the ground (sometimes more than once), and thus is highly dependent on the ground surface and the leg swing speed.

Knowing which specific balance strategy, or set of strategies, to apply for a given disturbance is still somewhat unclear. This is particularly true when the disturbance is small, and could be compensated successfully (if not quickly or efficiently) by any one of the different strategies. The speed of response, simplicity of application, and possible interference with ongoing tasks can all also affect which strategy (or strategies) should be used to maintain or recover balance. The final part of this section is dedicated to discussing existing research on combinations of these different types of strategies and/or multiple strategies of the same type (in Subsection 2.3.4).

2.3.1 Flexing

Flexing strategies can be broadly defined as using joint compliance to enable a robot to absorb small disturbances automatically (i.e., without high-level controller intervention). This generally happens either passively, by including compliant components in the mechatronic design, or actively, as part of the joint controller [1–4, 62–67].

The main difference between these two approaches is in the complexity of the controller and the design: with passive compliance, the joint itself is complex while the joint controller is relatively simple, while controlled compliance requires some form of high-speed force measurement and control, but can use fairly standard rigid joint designs. There is also an added benefit of the use of rigid joints in the controlled compliance cases, as it is sometimes useful to combine force and position control in a hybrid control strategy [4].

Although flexing is a very useful form of compensation when the disturbance is small, it is less effective in compensating other forms of disturbances. Some research has also demonstrated that having high levels of compliance in certain joints can actually increase the difficulty of performing dynamic motions, such as walking or running [64].

2.3.2 Leaning

The goal of leaning strategies is to change the biped's momentum in response to a disturbance (e.g., by varying posture or using momentum actuators such as reaction wheels) without modifying the state of any of the robot's contact points. This includes swinging the arms or legs, twisting the torso, or bending at the waist, similar to the way humans react to external disturbances while standing and walking. In general, leaning is performed based on measuring either the disturbance itself, or the effects of the disturbance, and countering the changes by accelerating the robot's links in a suitable direction.

A number of different unique leaning strategies for modifying the posture of a robot have been introduced in the literature [5,61]. However, the majority of these strategies are based on a heavily simplified model of the robot in the sagittal plane, which can lead to difficult translation onto a real system. Yoshida *et al.* investigated the leaning and stepping strategies employed by humans in the lateral plane in [68].

The most effective leaning approaches for real bipeds generally control the overall momentum of the robot to achieve a desired whole-body behavior [8–10, 32, 38, 69–71]. These methods are primarily used to map a low DOF desired momentum onto a set of high DOF joint level commands, enabling the controller to automatically and smoothly apply a number of different leaning strategies in parallel (e.g., arm swing and bending).

Momentum-based control started with two main groups: Kajita *et al.* with their concept of resolved momentum control [69], and Goswami and Kallem who used the rate of change of angular momentum [38]. Machietto *et al.* developed similar methods for using momentum in the control of animated bipeds [70], while Hofmann *et al.* used momentum-based controllers to improve independent COM and COP control of bipeds [71].

Lee, Orin, and Goswami have used the CMM to develop momentum-based controllers to reject disturbances while standing still on non-level and non-stationary ground in [8–10]. In their work, the centroidal momentum is controlled using the CMM, but achieving the desired linear momentum is prioritized, causing the robot to lean in response to external disturbances. CMM-based control with a non-zero desired angular momentum was used in [72], to successfully generate whole-body motions for kicking and jumping tasks.

Building on this successful dynamic kicking controller, Wensing and Orin showed that specialized algorithms are not required to calculate the CMM and discussed methods of applying centroidal momentum control strategies via the CMM in [73]. Centroidal momentum and centroidal dynamics have also been used to enable whole-body planning approaches for full-size bipeds, including dynamic contacts for both the feet and hands, in [74].

In the humanoid balance controller defined in [8,9], the desired Ground Reaction Force (GRF) and COP of each foot are actively controlled by directly relating them to the desired rate of change of linear and angular momentum, respectively. When the desired GRF and COP are not reachable, prioritized optimal solutions for reachable locations are found at the expense of errors between the desired and actual momentum.

By using weighting matrices to prioritize between the two types of momentum, the disturbance compensation strategy can be varied between fully respecting either linear or angular momentum. Effectively, when only the linear momentum is respected the motion of the COM is actively controlled, so the angular momentum must be used to balance. Similarly, when only the angular momentum is respected the centroidal rotation is actively controlled, so the linear momentum (i.e., motion of the COM) must be used to balance.

In [32], Moro *et al.* propose Whole Body Motion Control (WBMC) based on the use of a control element called 'attractors', to enable a humanoid to perform several tasks at once. This work was validated on COMAN, a torque controlled compliant humanoid, and appears to be the first use of joint momentum for the purposes of control [10].

Leaning strategies are by far the most popular of the various balance strategies, since they do not require the robot to change any of its contact points or include compliance in either the mechanical system or the controller. Unfortunately, leaning strategies can only generally handle low to moderate levels of disturbances, and are usually highly dependent on accurately measuring or calculating the effects of the disturbance.

Also, although leaning is generally capable of maintaining balance when subjected to a disturbance while standing still, additional challenges arise when there are multiple competing objectives for the robot. To deal with these issues, there is typically a requirement to define several heuristic parameters to prevent the robot from entering a state which contradicts its objectives. Subsection 2.3.4 discusses solutions to some of these issues.

2.3.3 Stepping

Stepping is the key balance strategy for dealing with moderate to large disturbances, particularly when a biped is in motion when disturbed. In its most basic form, a stepping strategy consists of finding a location on the ground to place a swinging foot to counteract a given disturbance, either wholly or partially, and then actively stepping to that location. If the disturbance has only been partially compensated, further steps or other balance strategies will then be necessary to return the robot to its desired normal behavior.

Finding the location to step requires the use of balance point estimators, which determine where a biped should step to achieve some desired outcome. As discussed in Section

2.2, most balance point estimators find a single point where the biped must step instantaneously to cause the COM to come to rest above the newly placed foot, although some are predictive, use multiple steps, or allow alternate desired outcomes. The most common balance point estimators were discussed in detail in Section 2.2, so here the focus of the discussion is on how they are applied for balance.

There are three general methods which are used to apply a stepping strategy, based largely on the balance point. The first method is applied with the majority of the instantaneous balance points, and involves moving the swinging foot as quickly as possible towards the calculated balance point (e.g., [18, 60]), which is continuously recalculated during the step. Since the balance point is typically moving away from the foot, the swinging foot is effectively trying to catch up to the point before it leaves the achievable stepping range.

The second method requires a predictive balance point estimator, using the predicted point as a static control reference for some form of task-based control of the swinging foot. In [14] this is the approach taken, where the GFPE is used once to find a desired balance point immediately after a push. This point is then provided as a desired footstep to a standard footstep-based stepping controller, which takes care of moving the swing foot to the point and brings the stance foot up to step beside the new swing foot location.

If the biped can alter its dynamics to exert some control on the balance point's dynamics during the step, generally through modification of external forces, then a third method can be used: moving both the swinging foot and the balance point towards a common location. An example of this is the work on using Capturability to control a biped, where the dynamics of the ICP are controlled by shifting the COP within the stance foot [12].

Stepping strategies rely on the ability to place a swinging foot somewhere specific on the ground (a region or point) to regain balance, which is usually dependent on having smooth, flat walking surfaces or a good models of the ground. Since stepping requires much more energy to achieve than other strategies, and can be computationally expensive, it should typically not be used to deal with moderate or small disturbances. Stepping also takes longer to execute, so is generally used as a last resort to prevent falling.

Although the movements associated with stepping will generally include some amount of leaning, it is typically a byproduct of moving the robot's swing foot, and does not necessarily help to counter the disturbance. Finally, the majority of these approaches ignore the effects of impact and rely on the inertia-free LIPM introduced in Subsection 2.1.3, which has inherent limitations in terms of the need for a COM with a constant height and no method of modelling either rotational inertia or angular momentum.

2.3.4 Combined Approaches

Work has been done in combining these different approaches to balance, as well as different strategies within a given approach, as discussed below. However, in most cases only one strategy is selected to compensate for a given disturbance, and the majority of solutions have been designed solely for when the robot is at rest, with no discussion of how the response would change for a moving robot. This poses several questions about how the different methods might compensate for disturbances while walking, and how they would react to successive disturbances. A selection of combined balance approaches are discussed below, with emphasis on the different types of combinations found in the literature.

In [75], flexing based on controlled compliance is used to maintain a desired waist orientation, combined with an omnidirectional stepping strategy for a robot at rest. The robot maintains its existing configuration until a suitably large disturbance is measured in the COP which will likely cause the robot to fall. When this occurs, the controller causes the robot to take one or two quick steps at set inter-foot distances, dependent on the direction of the potential fall. Regardless of whether the robot is stepping or standing, a compliant waist orientation controller is used to smooth the motion of the robot before inverse kinematics are applied to generate desired joint angles. This sequential combination of flexing and stepping makes the robot reasonably robust to disturbances.

In [76], disturbance suppression is applied via state feedback to lean towards external disturbances, then a reactive step is taken if the disturbance has not been fully compensated. This minimizes the difference between the desired and actual COM and ZMP trajectories using state feedback to suppress disturbances. If this is not enough, or if the robot is in a configuration which makes this difficult, the desired COM and ZMP trajectories are modified for the next step to compensate. Although this was developed for a robot in motion and has been verified on a physical robot, it relies on the LIPM to represent the robot's dynamics and does not take into consideration its posture or momentum.

In [77], a combination of ankle flexing and leaning at the hips is used to compensate for disturbances, using a double IP model built on the LIPM for a robot at rest. These two strategies are used in parallel to generate desired torques based on joint feedback, and are designed to mimic the human response to disturbances. There is no mechanism in this work for determining if the ankle and hip strategies are combined constructively or not, as their resulting torques are simply added together within the controller.

In [78], a reinforcement learning method is used to determine how best to apply these two strategies, as well as ICP-based stepping. This method uses the estimated disturbance to decide what combination of the three strategies should be applied, and uses them to

modify an online ZMP-based walking controller. Various points in the gait cycle are simulated and then applied on a small physical robot with relatively large feet, which show that the learning system is able to compensate for a slightly larger set of disturbances than the online ZMP-based walking controller alone. However, since the scale of each of the different strategies is dependent on learned behavior, and is built on top of an online ZMP-based walk, there are no guarantees that the strategies combine constructively.

A Model Predictive Control scheme was used in [79, 80] to combine the ankle strategy with stepping, which was later augmented to include the hip strategy and an optimized step duration in [81]. This approach is promising, as it avoids the saturation or thresholding which is typically used to transition between strategies, as in [82], as well as allowing online modification of the step duration, often a fixed or estimated parameter. However, it has only been applied for a simple simulated robot at rest, and requires both a good model and the resources to run an optimization over a finite window at every control cycle.

One of the major drawbacks of several of these methods is that they are only used while at rest, with no discussion of their use while walking or otherwise in motion. All of the methods which do discuss walking are based on the LIPM, which has several assumptions that must be compensated for in the controller. Further, for all of these methods, either only one strategy is used for a given disturbance, or a set of strategies are used in parallel with no consideration for whether they are combining in a constructive way.

An alternative method to these combinations of simpler strategies is to generate whole-body behaviors directly, which can automatically apply any number of simpler strategies by design. For example, the CMM-based leaning controller in [8–10] for a robot at rest was augmented to include both a GFPE-based stepping strategy, in [14], and a damage-prevention strategy when falling, in [83, 84]. A high level controller is used to choose which of the strategies (leaning, stepping, or safely falling) is applied at any time, based on the GFPE point's location immediately after a push.

The controller used in [14, 83, 84] also switches between different heuristically chosen weighting matrices based on the chosen strategy. This weighting matrix is used to prioritize between the linear and angular momentum: where the original leaning strategy prioritized respecting linear momentum, the stepping strategy prioritizes minimizing the angular momentum. In the controllers with stepping, the desired step location was also generated only once after a push and used as a reference point in a standard step planner, which left dealing with swing leg dynamics and any further disturbances to a feedback controller.

There have also been several examples in recent years of the application of Model Predictive Control (MPC) and/or optimization-based controllers in combination with incorporating the full dynamics of a robot into the planning and control stages to generate balance

and gait [85–91]. Many of these examples were developed (partially or completely) during the concentrated efforts towards completing the DARPA Robotics Challenge (DRC) and its virtual predecessor [85, 87, 88, 91]. For a thorough discussion of these and other approaches to the control of legged robots, see [39].

In [85], a combination of high-level instructions from human operators and fast computation of the full dynamics of the system is used as part of a full-body MPC method. This controller was applied to several tasks in the virtual DRC, but was determined to be too slow for application on a real robot due to a desire for simple and abstract costs resulting in complex motions. Although easier than designing control laws directly, this approach still requires fine-tuning to achieve the performance required.

The first successful use of a whole-body MPC on a physical robot in real time is described in [86]. This successful application of an MPC to the control of a real robot showed that this is a viable method for generating a feasible control strategy which can respond in real-time to environmental changes. A real-time nonlinear MPC was later implemented on a position controlled humanoid robot and found to improve its performance [89].

The most successful applications of MPC for the control of humanoids have incorporated reference points and balance points [90, 91]. Namely, the DCM was combined with MPC in [90] to achieve a robust walking controller which can be implemented on robots with different control methods in a variety of environments. In [91], a controller for balance recovery of the real-world robot DRC-HUBO+ was developed by combining the ICP with a ZMP-based MPC, achieving push recovery and balanced walking on uneven terrain.

Optimization based planning, estimation, and control approaches have also shown some promise when combined with full-body dynamics [87, 88]. In [87], an inverse dynamics-based full-body controller is augmented with a COM trajectory planner and a receding-horizon optimization which generates the next desired foot placement location. Finally, the MIT DRC team outlined the various optimization-based algorithms they used in their planning, control, and estimation work that was applied to great success at the DRC in [88].

2.4 Optimized Mechanism Design

Several prior works investigate the possibility of generating dynamic biped parameters using optimization [92–98]. In these works, either the cost of transport or the number of steps is used as the optimization metric, with a dual objective of both gait generation and physical property selection. One of the first examples of this application was the use of

genetic algorithms and evolutionary computing in [92, 93] to select control parameters (or, equivalently, a trajectory) and generate dynamic parameters in parallel.

Later, Haberland *et al.* developed a generic framework for extracting design principles from observed biological systems [94, 95] . Using the biological observations, a non-dimensionalized design space is populated with principles and then sampled. Optimal control is then used to test whether the principle is valid for the given design space.

In [96], an optimization and associated objective function were developed using a single design parameter, the spring constant between the model's legs, to optimize the design parameters and generate gait. The system is then controlled using trajectory tracking by reducing the biped to a 1 DOF system using hybrid zero dynamics [99].

Another framework [97] used simulation to co-optimize the control and design of bipeds. Comparing this to the original methods found in [92,93], the key differences are the addition of full 2D and 3D dynamics in a simulated environment and the possibility of non-periodic gait. Building on this approach, [98] fixed the robot's dynamic parameters and optimized the spring constants and trajectories (state and input) to design compliant gait trajectories for bipeds. In this thesis, a passive rotational joint will also be used in place of the stance ankle and the instantaneous dynamics of stepping will be ignored, as proposed in [98].

The above works all use a controller, either pre-existing or co-optimized, as part of the formulation of their dynamic parameter optimization. Therefore, the mechanisms which are found are optimized only for the given controller. The objective functions of these optimizations are also all based on the cost of transport, with several using torque norms for comparison. These measures are both dependent on the trajectory and controller used.

Although this approach has shown successful results, it limits the prospective applications of the biped to those which were conceived as part of the optimization and can suffer from overfitting (to the controller, trajectory, or both). It also assumes that a controller must be developed (or at least parameterized) either prior to or as part of the overall parameterization of the biped.

This limits these methods to a specific controller, or subset of controllers, which prevents general applicability of the biped itself for other use cases. To avoid these issues, some researchers have focused on how to quantify a mechanism's inherent balancing capabilities, or how to modify the mechanism to improve these capabilities, without depending on (and potentially overfitting to) a particular controller and/or trajectory formulation.

Building on earlier work ([21,22]), Azad *et al.* proposed a formulation of dynamic COM manipulability in [23] which defines a system's physical COM acceleration limits as an ellipsoid. Dynamic COM manipulability has also been successfully to optimize a postural task for the iCub robot as part of the CoDyCo project [100].

However, dynamic COM manipulability depends on the specification of a weighting matrix for normalization and the ellipsoid that is generated must be projected into lower dimensions to be used for balance. Further work analyzing the impact of the weighting matrices on the dynamic manipulability of these systems has recently been published [24].

An alternative metric was proposed by Featherstone in [25] and extended in [26, 27]: A set of linear and angular dynamic ratios he called velocity gains, which quantify how effectively an articulated system balancing on a passive contact can move its COM. These gains are independent of the control scheme used, as they are functions only of the properties and configuration of the mechanism, and provide an upper bound on how well any controller could balance the given system[1]. These gains have also been used in the development of effective planar balance controllers [101–103].

The gains are invariant to a scaling of the total mass of the system, and the angular velocity gain is also invariant to a scaling of total length, allowing the balancing capabilities of an entire class of mechanisms to be quantified with a single metric [27].

2.4.1 Velocity Gains

Linear velocity gain [27] is defined as a ratio of the change in horizontal COM velocity relative to an impulsive change in the velocity of the model's actuated joint(s), assuming a single passive (rolling or point) contact with the environment.

Similarly, angular velocity gain [27] is defined as a ratio of the change in angular COM velocity about the (instantaneous, if rolling) contact point relative to an impulsive change in the velocity of the model's actuated joint(s).

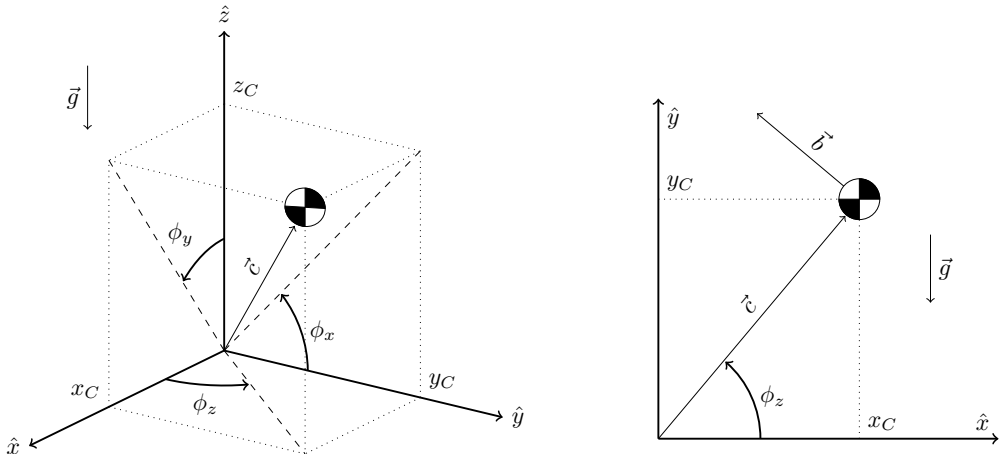
For 3D models, the linear velocity gain $\vec{G}_v$ includes both horizontal directions of motion, while the angular velocity gain $\vec{G}_\omega$ includes all 3 rotations[2] about the contact point:

$$\vec{G}_v(\Delta\dot{\boldsymbol{q}}_a) = \begin{bmatrix} \Delta\dot{x}_C \\ \Delta\dot{y}_C \end{bmatrix} \qquad \vec{G}_\omega(\Delta\dot{\boldsymbol{q}}_a) = \Delta\dot{\vec{\phi}} = \frac{\vec{c}\times\Delta\vec{v}_C}{|\vec{c}|^2} \tag{2.11}$$

where $\vec{c} = [x_C \; y_C \; z_C]^T$ is a vector from the contact point to the COM (see Figure 2.7a) with length $|\vec{c}| = \|\vec{c}\|_2$, the change in COM velocity is $\Delta\vec{v}_C = [\Delta\dot{x}_C \; \Delta\dot{y}_C \; \Delta\dot{z}_C]^T$, the change in

[1] In this context, balance is assumed to be primarily a function of COM motion (i.e., to move the COM above the support polygon). Angular momentum about the COM is assumed to be regulated.

[2] Although the $\vec{G}_\omega$ component about the vertical axis does not contribute to balance, it was included in the original definition of $\vec{G}_\omega$ along with a brief discussion of how it could be used in spinning motions [27].



(a) 3D Coordinates (b) 2D Coordinates

Figure 2.7: In general, $\vec{c} = [x_C \; y_C \; z_C]^T$ is a vector from the contact point to the COM and the angles $\vec{\phi} = [\phi_x \; \phi_y \; \phi_z]^T$ are measured from the reference frame to $\vec{c}$. For 2D, we assume gravity acts in the $-\mathbf{y}$ direction and $z_C = 0$, which means $\phi_x = 0$ and $\phi_y = \pi/2$.

angular COM velocity about the contact point is $\Delta\dot{\vec{\phi}} = [\Delta\dot{\phi}_x \; \Delta\dot{\phi}_y \; \Delta\dot{\phi}_z]^T = (\vec{c} \times \Delta\vec{v}_C)/c^2$, and the change in actuated joint velocity is $\Delta\dot{\boldsymbol{q}}_a$. Note that in these equations, it is assumed that $\Delta\dot{\boldsymbol{q}}_a$ is a unit velocity step (i.e., $||\Delta\dot{\boldsymbol{q}}_a|| = 1$) [27]. Both gains are divided by the velocity step magnitude, which gives $\vec{G}_v$ units of length and makes $\vec{G}_\omega$ dimensionless.

For a planar model, we assume that gravity acts in the $-\hat{y}$ direction and set $z_C = 0$, resulting in $\vec{\phi} = [0 \; \pi/2 \; \phi_z]^T$ (see Figure 2.7b). Using this to simplify Equation (2.11), we can extract the scalar velocity gains defined in [27]: $G_v(\Delta\dot{\boldsymbol{q}}_a) = \Delta\dot{x}_C$ and $G_\omega(\Delta\dot{\boldsymbol{q}}_a) = \Delta\dot{\phi}_z$.

Three methods of calculating the velocity gains are given in [27]: Direct, COM Jacobian, and Augmented Inertia Matrix. When using the Augmented Inertia Matrix method, virtual immobile prismatic joints (labeled $0 = \{x, y, z\}$) are inserted between the passive contact and the inertial reference frame.

The impulsive equations, with 3 virtual prismatic joints (labeled 0), 3 passive rotational joints (labeled p) at the contact, and the actuated joints (labeled a), are therefore:

$$\boldsymbol{H}'\Delta\dot{\boldsymbol{q}}' = \begin{bmatrix} \boldsymbol{H}_{00} & \boldsymbol{H}_{0p} & \boldsymbol{H}_{0a} \\ \boldsymbol{H}_{p0} & \boldsymbol{H}_{pp} & \boldsymbol{H}_{pa} \\ \boldsymbol{H}_{a0} & \boldsymbol{H}_{ap} & \boldsymbol{H}_{aa} \end{bmatrix} \begin{bmatrix} 0 \\ \Delta\dot{\boldsymbol{q}}_p \\ \Delta\dot{\boldsymbol{q}}_a \end{bmatrix} = \begin{bmatrix} \vec{\iota}_0 \\ 0 \\ \boldsymbol{\iota}_a \end{bmatrix} \tag{2.12}$$

where $\vec{\iota}_0$ and $\boldsymbol{\iota}_a$ are virtual joint and actuated joint impulses, respectively, and $\boldsymbol{H}'$ is the

Augmented Joint Space Inertia Matrix[3]. This enables the direct calculation of $\Delta\vec{v}_C$ since the virtual impulse $\vec{\iota}_0 = m\Delta\vec{v}_C$, where m is the total mass [27].

Since $\boldsymbol{H}_{pp}$ is a 3x3 symmetric positive definite matrix, we can use the middle row to get $\Delta\dot{\boldsymbol{q}}_p = -\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\Delta\dot{\boldsymbol{q}}_a$ as discussed in [27], and the equation for $\Delta\vec{v}_C$ becomes:

$$\Delta\vec{v}_C = \frac{1}{m}\left(\boldsymbol{H}_{0a} - \boldsymbol{H}_{0p}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\right)\Delta\dot{\boldsymbol{q}}_a \tag{2.13}$$

In 2D, with one passive contact joint ($p = \{1\}$), $\boldsymbol{H}_{pp}$ is scalar and $\boldsymbol{H}_{pp}^{-1} = H_{11}^{-1} = 1/H_{11}$.

In any given configuration, the matrix $\boldsymbol{H}'$ and vector $\vec{c}$ are fixed. Since the velocity gain equations are linear with respect to $\Delta\dot{\boldsymbol{q}}_a$, we can define a pair of gain matrices, $\boldsymbol{G}_{va}$ and $\boldsymbol{G}_{\omega a}$, such that $\vec{G}_v(\Delta\dot{\boldsymbol{q}}_a) = \boldsymbol{G}_{va}\Delta\dot{\boldsymbol{q}}_a$ and $\vec{G}_\omega(\Delta\dot{\boldsymbol{q}}_a) = \boldsymbol{G}_{\omega a}\Delta\dot{\boldsymbol{q}}_a$ [27].

These gain matrices are defined as:

$$\boldsymbol{G}_{va} = \frac{1}{m}\begin{bmatrix}\boldsymbol{H}_{xa} - \boldsymbol{H}_{xp}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\\ \boldsymbol{H}_{ya} - \boldsymbol{H}_{yp}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\end{bmatrix} \qquad \boldsymbol{G}_{\omega a} = \frac{\tilde{c}}{m|\vec{c}|^2}\left(\boldsymbol{H}_{0a} - \boldsymbol{H}_{0p}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\right) \tag{2.14}$$

where $\tilde{c}$ represents the 3x3 skew symmetric matrix.

For 2D systems, the (horizontal) gain vectors from [27] can be extracted from these general gain matrices: the first row of $\boldsymbol{G}_{va}$ is the linear velocity gain vector, and the third (final) row of $\boldsymbol{G}_{\omega a}$ is the angular gain vector.

Since the only restriction placed on $\Delta\dot{\boldsymbol{q}}_a$ for these gains is that $||\Delta\dot{\boldsymbol{q}}_a|| = 1$, the relative values of its elements can be selected to achieve a desired system behavior. As an example, to use a specific joint for balance, the $\Delta\dot{\boldsymbol{q}}_a$ element for that joint can be set to 1 and all other elements can be set to 0.

2.4.2 2-Link Planar Momentum Gains

A 2-link planar inverted pendulum, the simplest balancing mechanism, has only one actuated joint ($a = \{2\}$) so its scalar velocity gains can be directly defined as ratios [27]:

$$G_v = \frac{\Delta\dot{x}_C}{\Delta\dot{q}_2} \qquad G_\omega = \frac{\Delta\dot{\phi}_z}{\Delta\dot{q}_2} \tag{2.15}$$

Featherstone also defined momentum gains for the 2-link planar model [27]:

[3]The standard Joint Space Inertia Matrix, $\boldsymbol{H}$, that was defined in Section 2.1.1 is the remaining lower right submatrix of $\boldsymbol{H}'$ after all of the elements of $\boldsymbol{H}'$ with a 0 in either subscript have been removed. In other words, $\boldsymbol{H}$ is the 2×2 block matrix consisting of $\boldsymbol{H}_{pp}$, $\boldsymbol{H}_{pa}$, $\boldsymbol{H}_{ap}$, and $\boldsymbol{H}_{aa}$.

- Linear momentum gain (G_m) is a measure of the change in horizontal linear COM momentum due to an impulse at the actuated joint:

$$G_m = \frac{m\Delta\dot{x}_C}{\iota_2} = mG_v\frac{\Delta\dot{q}_2}{\iota_2} \tag{2.16}$$

- Angular momentum gain (G_o) is a measure of the change in the moment of momentum of the COM about the contact due to an impulse at the actuated joint:

$$G_o = \frac{mc^2\Delta\dot{\phi}_z}{\iota_2} = mc^2G_\omega\frac{\Delta\dot{q}_2}{\iota_2} \tag{2.17}$$

Note that G_o is defined using the change in the moment of momentum about the contact (effectively, the angular momentum about the contact due to COM motion). This is not the same as the change in total angular momentum about the contact, which is always 0 for a passive rotary joint (which, by definition, cannot apply any torques) [27].

As shown above, momentum gains are directly related to velocity gains. For the 2-link planar model, they are both strictly positive multiples of their respective velocity gains, since $\boldsymbol{H}$ is positive definite and $\Delta\dot{q}_2/\iota_2 = H_{11}/\det(\boldsymbol{H})$ [27]. In light of this, Featherstone concluded that there was no objective reason to use momentum gains.

2.5 Summary

Balance points are an excellent tool for maintaining and regaining balance, as they are easy to compute, use simplified models, and can generate dynamic footstep locations online. However, due to the assumptions of the simplified models and a lack of existing dynamic gait controllers which use these methods, few of the balance point estimators can be directly used in real world applications. Therefore, a new balance point estimator which encapsulates and generalizes the existing methods, while reducing or eliminating their shortcomings, would be a critical asset for dynamic balance and gait control.

There is also a lack of existing research on how to both measure and improve the fundamental physical limits of peak balancing performance for a given mechanism. Although the few approaches that have been developed can handle some situations, a generalized method is needed to enable these two key outcomes. In light of this, a generalized metric which can measure how well a given system can balance, for a variety of desired balancing behaviors, should be developed. Further, to make use of this metric for the purposes of mechanism design, an optimization framework built on the new metric would be valuable.

In Chapter 3, the vertical planes used to find the GFPE [14] and the 3D FPE [20] are compared when used in a push recovery task. The results of this comparison show that the vertical plane used to find the GFPE is in fact a special case of the 3D FPE plane, where there is no relative motion or rotation of the robot's links. The 3D FPE is then used, along with a continuously updated impact plane, to generate a reference point for a dynamic balance and gait controller which can produce corrective stepping responses in real time. However, the performance of this controller is limited due to a lack of a balance point estimator which includes all of the desirable features discussed in Section 2.2.3.

Chapter 4 describes the development of a novel predictive 3D balance point estimator, called the Spherical Foot Placement Estimator (SFPE), which extends the FPE [18] and its existing 3D extensions (the GFPE and 3D FPE [14, 20]). This novel estimator has been developed to address the drawbacks of existing approaches: it is predictive, considers impact, includes rotational inertia, allows ankle torques and 3D Center of Mass (COM) motion, does not require flat ground or heuristic parameters, and works for a biped at rest or in motion. The SFPE is compared to existing estimators using a simple COM feedback controller, which is then modified to use the SFPE to improve a biped's dynamic balance. This results in an augmented optimal controller with both leaning and stepping.

Building on the velocity gains and the planar 2-link momentum gains described in Sections 2.4.1 and 2.4.2 [27], simplified notation is developed in Chapter 5 to facilitate the definition of these types of impulsive ratios. Using this notation, the momentum gains are extended and expanded to general 2D and 3D systems and two methods for calculating these new general momentum gains are described, along with an additional novel gain based on Featherstone's spatial notation which incorporates both the angular and linear gains. Using this spatial gain and its associated gain matrix, the relationship between the velocity and momentum gains and the centroidal momentum of a system is discussed.

Bringing all of these definitions together, a generalized gain formulation is then developed which includes all of the pre-existing and novel gains as special cases and/or subsets. These generalized gains are then used to define a set of objective functions in a generalized optimization framework. Since the gains are functions only of the configuration and physical properties of a given system, the framework is therefore able to optimize the COM motion properties of a mechanism independent of any controller and/or trajectory. The gains are demonstrated and compared as objective metrics, using this generalized framework for parameterized mechanism optimization, in Chapter 6.

Chapter 3

Dynamic Gait: Fall, Catch, Repeat[1]

In this chapter, the 3D FPE [20] is used within a high-level control strategy to achieve a dynamic gait capable of handling external disturbances. A key benefit of this approach is that the robot is able to respond in real time to external disturbances, regardless of whether it is at rest or in motion. This strategy is implemented in simulation to control a 14-DOF lower-body humanoid robot subjected to unknown external forces, both when at rest and while walking, and shown to generate successful stabilizing stepping actions.

The 3D FPE [20] (see discussion in Section 2.2.2) is used to generate a control reference to inform both a high-level state machine and its associated task-level trajectory generator. The task-level trajectories are used as control inputs to a prioritized Jacobian-based feedback loop [60] and a simple low-level PD joint controller to drive a simulated robot.

The goal of this overall control strategy is to allow the robot to respond directly to external disturbances by stepping onto the 3D FPE point, either while standing still or walking. One of the key benefits of this strategy is that recovery from external disturbances is simply a subset of the elements required for dynamic gait. To walk, the robot needs only to push itself into an unstable state in the desired direction of motion, then allow the step-based recovery mechanism to prevent it from falling.

Both [18] and [60] have investigated earlier versions of this form of state-machine-based control strategy. However, the first deals only with planar bipeds with point feet, while the second uses a constant plane of motion, requires separate lateral stabilization, and has very limited discussion of external disturbances. This work uses a continuously updating plane, similar to [20] and [14], which allows it to respond in 3D to unknown external

[1] An earlier version of this chapter was previously published in [104].

disturbances. It also includes dynamic gait generation, unlike [14], and is used directly as part of a control method, as opposed to a measurement method such as in [20].

The main novelty of this controller is the use of a continuously calculated 3D FPE point to achieve step-based balance, both while standing and in motion. The 3D FPE has only been used until now in the biomechanics literature, and no existing work has been done in 3D using a continuously updating FPE plane. The results demonstrate that this approach is capable of compensating for external, unknown disturbances as part of a gait cycle.

3.1 Formulation

The proposed method of handling recovery from external disturbances, and by extension dynamic walking, builds on the previous FPE research discussed in Section 2.2.2. Here, the 3D FPE will be used to generate a control input for a high level state-machine based controller with online task-level trajectory generation and a prioritized Jacobian-based feedback loop, as described in Section 3.2. First, this section explains the differences between the GFPE and 3D FPE, to justify the use of the 3D FPE method in this work.

In [14], the vertical GFPE plane is chosen parallel to $\vec{v}_C$, sampled immediately after an external disturbance. In their formulation, the robot is assumed to be at rest before any disturbance occurs, and therefore the COM should only move in the direction of $\vec{v}_C$ until action is taken. This assumption, and therefore the plane itself, is reasonably valid if the robot is standing still when a disturbance takes place, but is less appropriate if the robot is already moving when disturbed (such as being bumped while walking).

In [20], the vertical 3D FPE plane is chosen to be perpendicular to $\vec{k}_G$, the total angular momentum of the robot taken about the COM ground projection. A system's centroidal angular momentum, $\vec{k}_C$, is a product of the average angular velocity, $\vec{\omega}_C$, and its centroidal rotational inertia, as shown in equation (2.5). Therefore, by using the angular momentum calculated about the COG, $\vec{k}_G$, the 3D FPE directly takes into account both the linear velocity of the COM, $\vec{v}_C$, and its centroidal angular momentum, $\vec{k}_C$.

This means that the GFPE plane defined in [14] can be considered a special case of the 3D FPE plane, where none of the links are rotating or moving relative to one another (i.e., $\vec{k}_C = 0$). Since this special case is only valid when the entire robot is at rest or translating purely linearly (difficult to achieve in the real world), the plane from the 3D FPE work is used in the remainder of this chapter. This allows the robot to respond to disturbances not only in the restricted cases of the GFPE plane, but also when it is already moving.

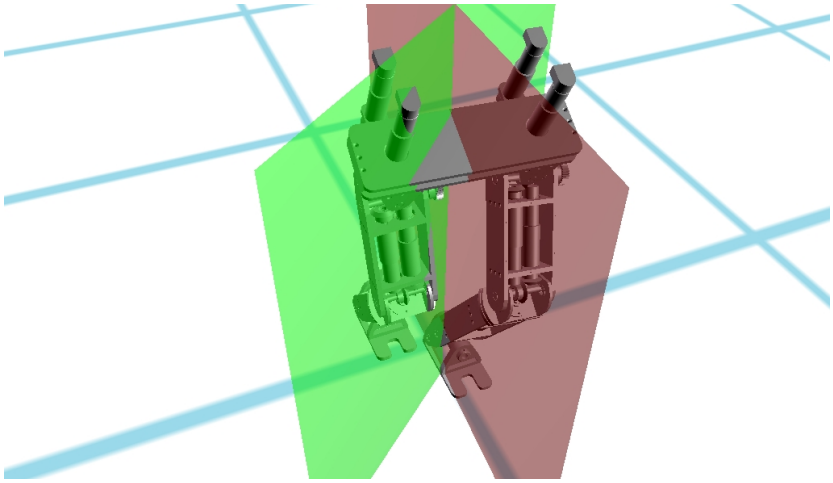


Figure 3.1: Image showing the Generalized FPE and 3D FPE planes, as defined in [14] and [20]. The robot's right foot (on the left side of the image) has just lifted during walking. The green plane is the 3D FPE plane, while the red plane is the GFPE plane.

The difference between the two planes is shown in Figure 3.1, just after the robot has lifted its right foot (on the left, in the figure) while dynamically walking: the 3D FPE plane has already responded to this change by rotating towards the lifted foot, but the GFPE plane remains aligned with the current direction of linear translation. Once the robot starts to fall towards the lifted foot, the GFPE plane approaches the 3D FPE plane, as the COM velocity rotates to point in that direction.

Once a plane is selected, the centroidal dynamics are projected into the plane to determine the position of the 3D FPE point. The projection uses a horizontal unit vector normal to the plane, $\hat{n} = \vec{n}/||\vec{n}||$, and $\vec{n}$ in the case of the 3D FPE is given by $\vec{n} = \vec{k}_G - (\vec{k}_G \cdot \hat{k})\hat{k}$, where $\hat{k}$ is a vertical unit vector in the direction opposite gravity.

First, the average angular velocity, $\vec{\omega}_C$, is projected to find the angular velocity of the model in the plane: $\omega = \vec{\omega}_C \cdot \hat{n}$. Second, the centroidal rotational inertia defined in equation (2.5), $\tilde{I}_C$, is projected to find the rotational inertia of the model in the plane: $I_C = \hat{n}^T \cdot \tilde{I}_C \cdot \hat{n}$. Third, the linear velocity of the COM, $\vec{v}_C$, is projected into the plane using $\hat{k}$ to get the vertical speed of the COM, $\dot{h} = \vec{v}_C \cdot \hat{k}$, and using $\hat{u}$, a horizontal unit vector parallel to the plane (i.e., $\hat{u} = \hat{k} \times \hat{n}$), to get its horizontal speed: $v = \vec{v}_C \cdot \hat{u}$.

Using these planar properties, the angle ϕ between a vertical axis and the line between the COM and the 3D FPE point can be found using the original 2D FPE angle equation (2.10). Once the angle ϕ is found, the location of the 3D FPE point $\hat{F}$ can be found (using trigonometry) as: $\hat{F} = \hat{G} + h\tan(\phi)\hat{u}$.

This balance point is used as a reference position for the high-level state machine and the task-level trajectory generation algorithms described in the next section. More details on the development of the 2D FPE and the 3D FPE extension can be found in [18] and [20].

3.2 Gait Controller

A gait controller is defined in this section which enables a biped to recover from external disturbances, both at rest and while walking, using the 2D gait controller from [18] as a starting point. Although a 3D extension of the original controller was made in [60], it used a constant FPE plane in the desired direction of motion, requiring the use of ZMP-based stable gait generation for the majority of the gait cycle to compensate for the out-of-plane 3D dynamics. The existing controller was augmented to improve the walking performance, both by defining functions at a higher level and including states for leaning forward (to instigate walking) and settling the feet on the ground.

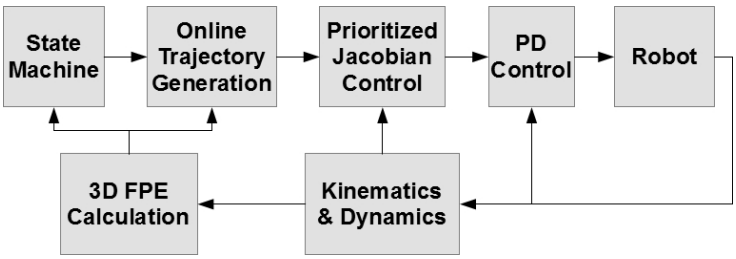


Figure 3.2: Diagram of the high-level controller.

The high-level controller (shown in Figure 3.2) is built around the use of a symmetric state-machine which defines the high level state of the robot at any particular time, as shown in Figure 3.3. A key feature of the controller is that recovery from external disturbances while at rest is a direct subset of the functionality required to walk. To walk, the high-level controller simply needs to cause the robot to move its COM until it begins to fall in the desired direction of motion, then enter push-recovery mode. It should be noted that the ground projection of the COM, $\hat{G}$, moving outside the robot's support polygon is only indirectly related to this mode: If the angular momentum is large enough, the stepping strategy will kick in before $\hat{G}$ exits the support polygon.

The state of the system, as determined by this state machine, is used to generate task-level trajectories, for the COM and both feet, to achieve the objectives of the given state. These trajectories are generated online, to avoid the need to calculate trajectories *a priori*

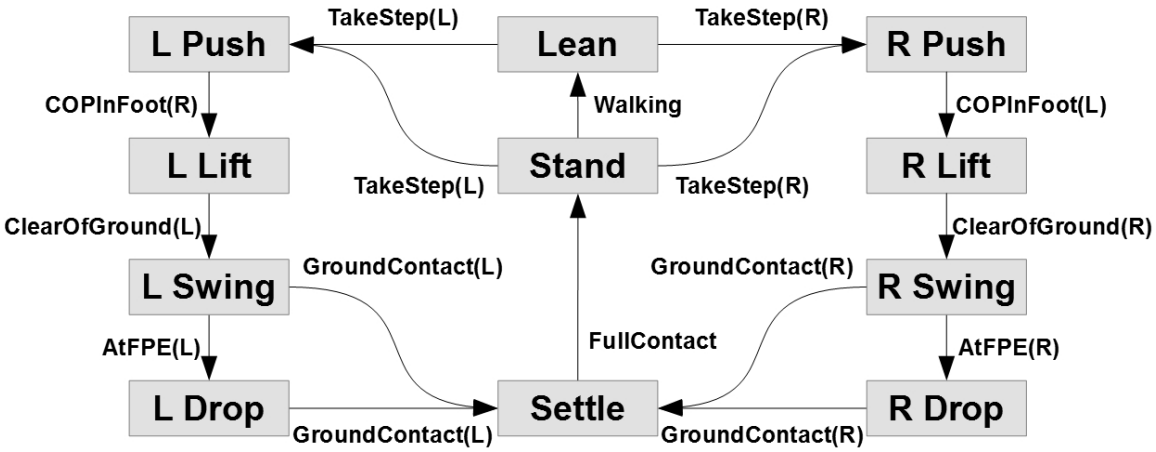


Figure 3.3: Diagram of the high-level state machine used in the controller.

and therefore allow maximum flexibility to respond to disturbances. Depending on the current state, different objectives exist for the desired COM and position and orientation of each foot. A set of goal positions and orientations are defined, based in part on the positions and orientations when each state starts. A trajectory is then generated using a quintic spline, to allow specification of the trajectory's initial and final derivatives.

To translate between task-level trajectories and the desired joint angles of the robot, a prioritized Jacobian-based feedback loop is used. This portion of the control method has been described previously in [60], so will only be briefly introduced here. In each state, task trajectories are prioritized into high and low categories. The high- and low-priority Jacobians are then built by stacking the task Jacobians in each priority category.

The overall Jacobian is found by projecting the low-priority Jacobian onto the null space of the high-priority Jacobian, and this overall Jacobian is inverted to translate between the task trajectories and desired joint trajectories. Finally, a simple PD control loop is used to calculate the torques required to move the robot to the desired joint angles.

3.2.1 State Machine

In the original state machine in [18], developed for a simple planar robot, the states defined joint trajectories directly. These definitions were specific to the particular biped geometry, and not easily generalizable to other configurations. In [60], the trajectories were defined in task space, in terms of Cartesian trajectories of the COM and swing foot,

providing a more general framework for arbitrary DOF legs. However, the trajectories defined in [60] were specific to the *a priori* chosen FPE plane and assumed level ground.

In the state machine described below, the states and the transitions have been defined with a further level of abstraction, to facilitate its use on a broad range of different biped robots and handle on-line changes to the orientation of the FPE plane. For example, during either of the 'Swing' states, the swinging foot is required to maintain "ground clearance" instead of "a height of at least h", to allow for variable terrain in the future.

States and Objectives

- *Stand*: The COM is moved to a set height to maximize its ability to respond to disturbances, centered between the positions of the two feet. A secondary objective is to maintain full ground contact with both feet.
- *Lean*: Move the COM at the desired COM velocity (where a velocity of 0 is equivalent to standing still), while maintaining full ground contact with both feet.
- *F Push*: Move the COP into the opposite foot, by augmenting the desired COM velocity, and maintain full ground contact with both feet.
- *F Lift*: Lift and rotate the given foot off the ground to achieve the desired ground clearance, while using remaining joints to maintain the desired COM velocity.
- *F Swing*: Move the given foot horizontally towards a point above the 3D FPE point, while both maintaining ground clearance of the foot and the desired COM velocity.
- *F Drop*: Lower the given foot onto the 3D FPE-based tracking point, while maintaining the desired COM velocity. The tracking point is offset from the 3D FPE point depending on the desired velocity. If the robot is walking, the tracking point is slightly behind the 3D FPE point, allowing the robot to maintain momentum in the desired gait direction. If the robot is standing and responding to a disturbance, or intends to stop walking, the tracking point is slightly in front of the 3D FPE point.
- *Settle*: Maximize the ground contact area of both feet.

Transition Functions

- *TakeStep(F)*: Checks if the given foot should take a step. The function is active if the 3D FPE point is further from the given foot than the opposite foot.

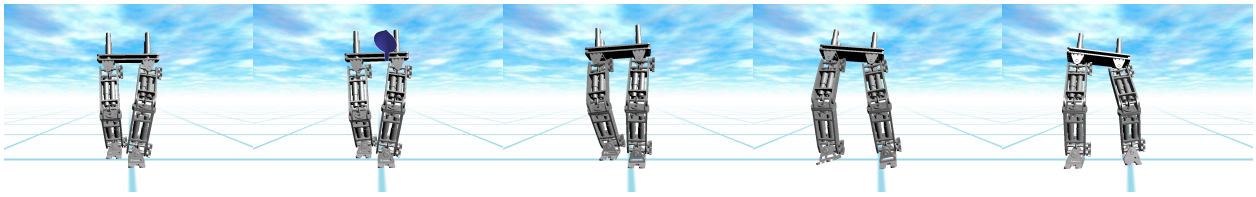
- *COPInFoot(F)*: Checks if the Center of Pressure is within the given foot. The function is active once the COP is at least a set distance within the foot's perimeter.
- *ClearOfGround(F)*: Checks if the given foot is clear of the ground. The function is active if the entire foot contact surface is a suitable distance away from the ground.
- *AtFPE(F)*: Checks if the given foot has reached the 3D FPE point. The function is active if the given foot is within a set horizontal distance of the 3D FPE point.
- *GroundContact(F)*: Checks if the given foot has touched the ground. The function is active if the foot's contact surface has touched the ground's contact surface.
- *FullContact*: Checks if both feet are in full contact with the ground. The function is active when both feet are flat on the ground.
- *Walking*: Checks if there is a desired COM velocity. If so, the function is active.

3.3 Results

The dynamic gait controller discussed in Subsection 3.2 was verified in simulation using a custom 14 DOF lower body humanoid (70cm tall, approx. 30kg) developed at the University of Waterloo [105]. The robot has 7 DOF in each leg, 3 at the hip, one at the knee, and 3 at the ankle. The CAD drawings of the robot were used to generate a SimMechanics dynamic model, including both the inertial and motor dynamics [106].

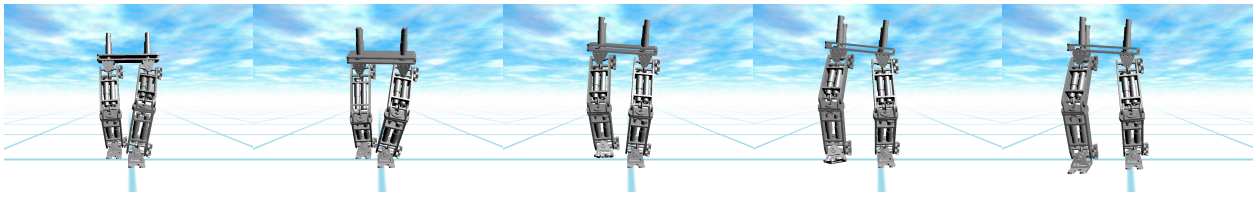
Various simulations were carried out where internal (walking) and external (pushes) disturbances were introduced, and the biped used the proposed controller to regain stability. These simulations consist of three classes of disturbance: being pushed while standing still, dynamic walking without external disturbances, and being pushed while walking. In all of these simulations, the robot was initialized to begin the simulation in a static staggered stance, with the robot's left foot forward and both feet flat on the ground.

The first simulation was used to verify that the controller was acting as expected in the presence of an external disturbance while the robot was stationary. A simulated force of 150 N was applied to the center of the torso in a direction out of the page for 0.1 seconds. The series of images in Figure 3.4 show the progression of the simulation at the transitions between the various states in the controller's state machine. At the end of the simulation, the robot is stationary with both feet on the ground.



(a) Initial Pose (b) After Push state (c) After Lift state (d) After Swing state (e) In Stand state

Figure 3.4: Simulation results when the robot is pushed from behind on the center of the torso with a force of 150 N for 0.1 seconds. The dark blue circle indicates the location of the disturbance force, the orientation is perpendicular to the plane shown, out of the page.



(a) Initial Pose (b) After Lean/Push (c) After Lift (d) After Swing (e) In Stand

Figure 3.5: Simulation results when the robot walks by leaning until it starts falling, then recovering.

The second simulation included a desired COM velocity, representing a desired speed and direction to walk. In this simulation, the robot is statically stable until it reaches the Swing state, as the Lift state is augmented to move the swinging foot in the direction of desired motion before switching to the Swing state. A series of images, in Figure 3.5, show the simulation at the transition points between states for one step of the walking cycle. The gait is such that the robot is statically stable at the end of each step in the cycle.

In the final set of simulations, the robot was subjected to external disturbances from various angles while dynamically walking using the strategy from the second simulation. For these simulations, the disturbance lasted for 0.1 seconds and acted on the center of the torso. The external force was applied during the Swing state, to demonstrate the robot is capable of responding to disturbances while in the dynamic portion of its walk cycle. The landing positions of the swinging foot in each of the various scenarios are shown in Figure 3.6, along with the foot paths from the point they start to differ.

In Figures 3.7a and 3.7b, the X and Y trajectories of the 3D FPE point in the three different push-while-walking scenarios are compared to the trajectory of the 3D FPE point in the normal walking scenario. Since the 3D FPE point is located on the ground, and the ground is assumed to be flat, the Z component is left out of the comparisons.

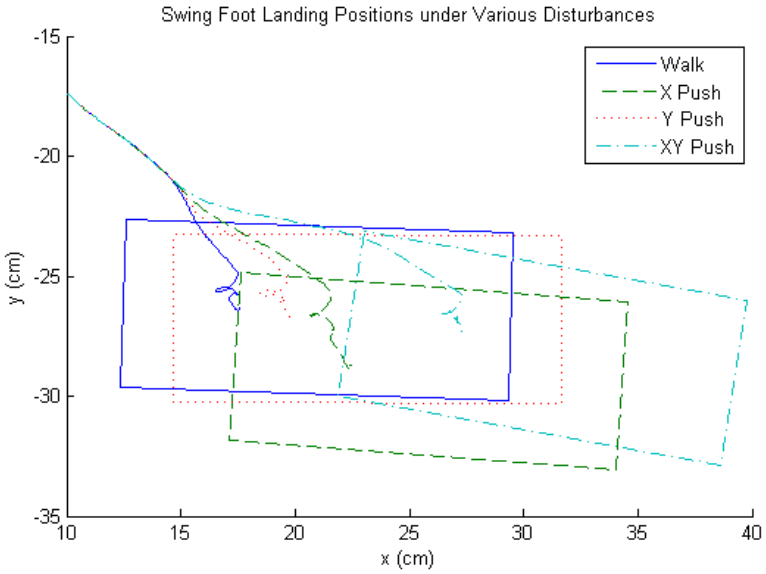


Figure 3.6: Landing positions of the swing foot in each of the four walking scenarios: normal walking (Walk), and being pushed in the X, Y, and XY directions while walking (* Push). The foot paths leading up to and after being pushed are also shown.

It is apparent from these graphs that at the time of the push (2 s), the 3D FPE point's location starts to diverge, depending on the different forces applied. As can be seen from the final robot poses in Figures 3.4 and 3.5, the controller performs very well in all applied scenarios. All of the final poses are statically stable, and in the case of the walking simulations the robot moves on to begin a step with the opposing foot.

In the specific implementation of the controller used in the simulations above, the Push and Lean states were combined to allow the robot to both move in the desired direction of motion and shift its COP into the chosen stance foot in parallel. This may have affected the timing of the initial states while walking, but since these states are during statically stable double support sections of the gait, this would not have affected the overall performance.

3.3.1 Discussion

Some potential issues with the current state machine used in the controller are:

- If both feet are an equal distance from the 3D FPE point, the chosen swing foot depends on the order of transition evaluation.

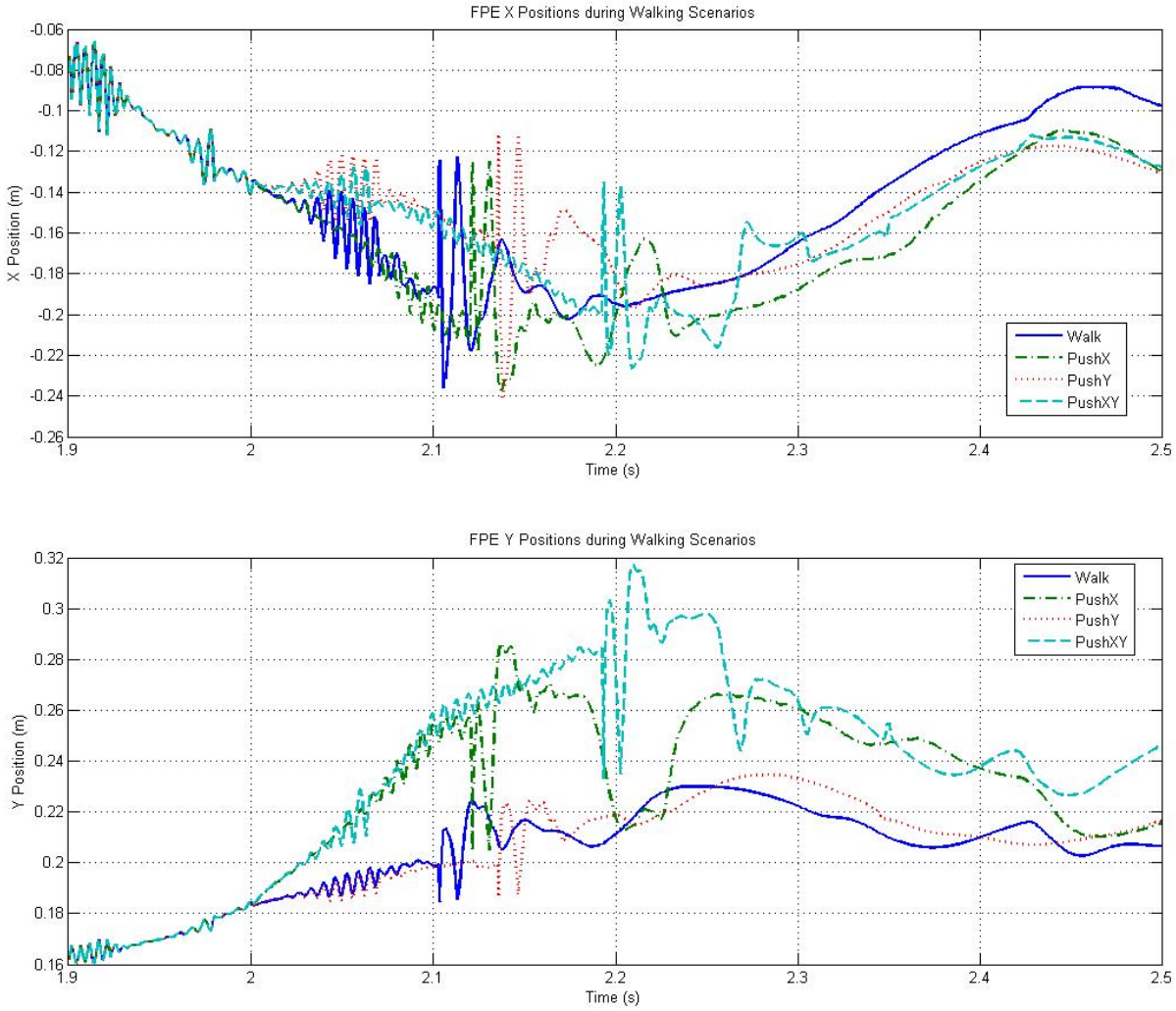


Figure 3.7: Comparison of the 3D FPE point trajectories for the four different walking scenarios (X on top, Y on bottom). The Swing state starts at 1.94 s, ending the initial oscillations caused by lifting the swing foot, while swinging the foot causes the smaller oscillations leading up to the push starting at 2 s. During the push, more oscillations are seen as the swinging foot causes directional changes in $\vec{k}_G$. Following the push, the robot enters the Drop state at different times in each scenario. Switching from Swing to Drop is the cause of the large oscillations between 2.1 and 2.25 seconds, when the controller switches from global to local reference frames to facilitate contact dynamics with the dropping foot. Note that although the 3D FPE point oscillates frequently due to changes in the direction of $\vec{k}_G$, these oscillations are filtered out before the balance point is used in the controller.

- If the robot is pushed in the *F Lift* or *F Swing* states, the wrong foot may be in the air. However, the continuous calculation of the 3D FPE point should allow the controller to adapt to many of these disturbances. For example, if the robot is pushed from the side opposite to the lifted foot, the raised foot will attempt to track the new 3D FPE point. If it is unable to do so, then it will make contact with the ground and a new round of push recovery will begin with the other foot.
- If the robot is pushed away from the 3D FPE point in the *F Drop* state, it is possible that the robot will fall. A possible solution is adding a transition to detect this and using it to switch back to the *F Swing* state.

3.4 Summary

This chapter outlined an approach for dynamic gait generation and control, capable of responding to unknown disturbances, using the 3D FPE balance point estimator to find a point at which to step to regain balance after a disturbance. This target stepping location is re-computed at each timestep, allowing the robot to adapt to disturbances on-line.

A simple state machine is used to generate a full gait cycle; the robot initiates lifting of the swing leg either when forward progress is desired or when a disturbance is observed, and tracks the target foot placement location to determine the swing leg placement. The proposed approach was tested in simulation and shown to generate stabilizing foot placements to disturbances from arbitrary directions, both while standing still and in motion.

Although the 3D FPE is useful for finding an instantaneous footstep location, a predicted future footstep location and the time available to swing the foot would improve the performance of this system. The heuristic approach developed in the state machine, to determine how to shift the COP into the desired stance foot, and then move the swing foot, would also benefit from a predictive balance point estimator. Finally, the use of switching prioritized Jacobians for controlling the robot results in control disturbances and oscillations each time the prioritization is switched, leading to a desire for a more robust method of controlling the robot as a whole.

A predictive balance point estimator is introduced in the next chapter which solves some of the issues with the 3D FPE. Along with an optimization-based whole body controller, it enables smooth transitions between motions.

Chapter 4

Spherical Foot Placement Estimator[1]

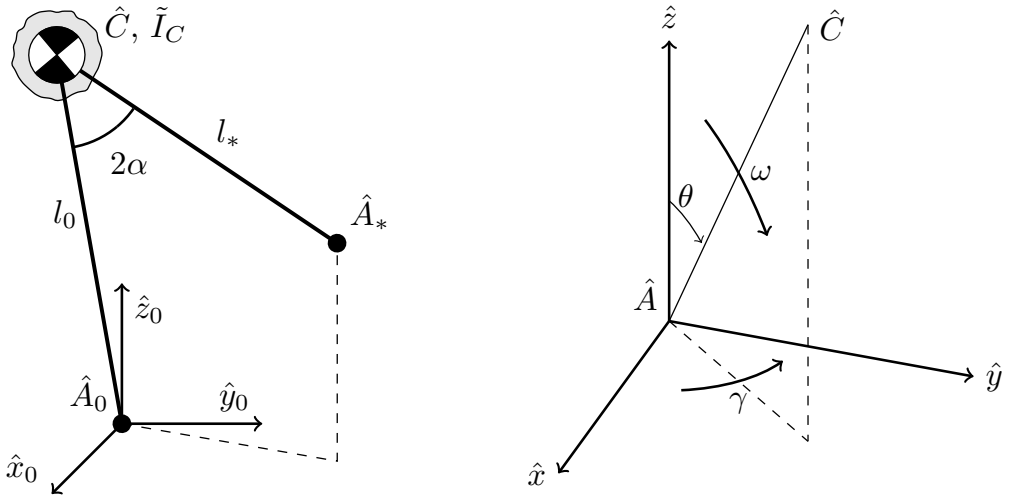
In Chapter 3, the existing 3D formulations of the FPE [18], the GFPE [14] and 3D FPE [20], were compared and the 3D FPE point was chosen as a decision variable and to generate desired stepping locations online. When using the 3D FPE, the full 3D dynamics of a biped are projected into the assumed 3D FPE impact plane, to determine the planar inertia and an average angular velocity.

As shown in the previous chapter, although the 3D FPE can be used as the foundation of a reasonably successful leaning and stepping strategy, there are several issues with this approach. In this chapter, a novel balance point estimator is introduced which projects a biped's full 3D dynamics into a vertical plane containing the COM and a fixed anchor point, a predicted vertical impact plane, and a horizontal plane.

The formulation of this new balance point estimator has many desirable features (e.g., those in Table 2.1): it is predictive, considers impact, includes rotational inertia, allows ankle torques and 3D COM motion, does not require flat ground or heuristic parameters, and works for a biped at rest or in motion. It also provides recovery step location prediction and momentum objectives with smooth dynamics.

This novel balance point estimator is called the Spherical Foot Placement Estimator (SFPE), as it uses spherical coordinates to extend the original (planar) Foot Placement Estimator (FPE) found in [18] and its existing 3D extensions: the Generalized FPE (GFPE) [14] and 3D FPE [20]. The SFPE has been developed to address the drawbacks of existing balance point estimators, and can be used to determine if a biped is going to lose its balance, how to compensate, and where to step if a step is needed.

[1] An earlier version of this chapter was previously published in [107].



(a) Diagram of the SFPE's 3D rimless spoked wheel model.

(b) Diagram of the spherical coordinates used for the SFPE.

Figure 4.1: Diagrams of the SFPE model and coordinates. The model in (a) shows that the COM, $\hat{C}$, is used as the attachment point of two legs with point feet. Note that the model used here includes a 3D rotational inertia about $\hat{C}$, labeled $\tilde{I}_C$. In (b), the spherical coordinates centered at the anchor point, $\hat{A}$, are used to describe the motion of the COM, $\hat{C}$. Note that θ is always measured in a vertical plane which contains both $\hat{C}$ and $\hat{A}$ and the distance between $\hat{C}$ and $\hat{A}$, the leg length l, is assumed to be constant.

4.1 Formulation

Similar to the GFPE [14], the SFPE uses a rimless wheel model, with two fixed length legs attached at the COM, $\hat{C}$, and a leg separation of 2α (see Figure 4.1a). It is assumed that only one of the model's point feet is in contact with the ground at any given time, allowing them to be labeled based on the state of the model: $\hat{A}_0$, the current anchor foot (on the ground), and $\hat{A}_*$, the future anchor foot. When a step is taken, $\hat{A}_*$ is assumed to land on the ground without slipping or bouncing at the same instant as $\hat{A}_0$ lifts off the ground, causing the status (and therefore labels) of the feet to switch instantaneously.

By assuming a fixed leg length between $\hat{C}$ and each anchor point, the motion of $\hat{C}$ can be described by a series of piecewise rotations about successive anchor points, replacing the common assumption of planar motion of the COM. This assumption produces a more realistic COM path, as typically the COM of a biped follows smoothed inverted pendular curves in 3D, not motion in successive (approximately horizontal) planes [108].

Thanks to this assumption of purely spherical motion of $\hat{C}$ during each step, spherical coordinates centered at each successive anchor point are used to define the model's dynamics in terms of three variables (see Figure 4.1b): the angle between the leg and the vertical axis, θ; its derivative, $\omega = \dot{\theta}$; and the angular velocity of $\hat{C}$ about the same vertical axis, γ. We assume no rotation about the stance leg axis.

These variables define the motion of the COM of the simplified model about the given anchor point, based on the linear velocity of the COM of the full multibody system. The simplified model includes only a single rigid body with its COM at the COM of the full system, which is constrained to spherical rotation about the anchor point.

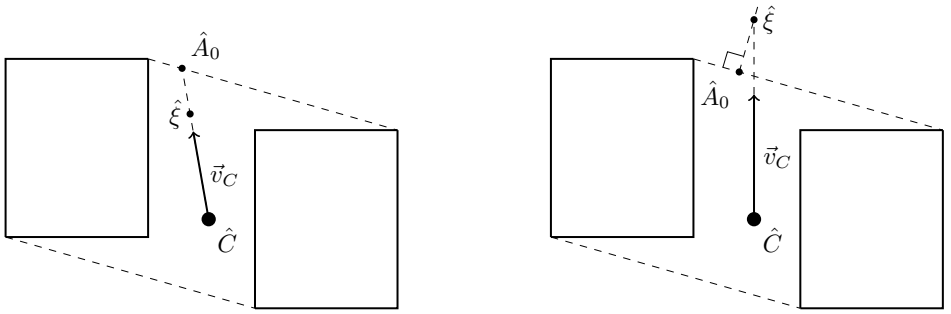
Unlike the GFPE, the model used in this chapter moves in 3D and includes rotational inertia about the COM, labeled $\tilde{I}_C$. This rotational inertia is the rotational submatrix of the biped's centroidal inertia matrix (the Composite Rigid Body matrix for the biped evaluated at the COM [10]). Multiplying the inverse of the centroidal inertia matrix by the centroidal momentum of the biped, the system's COM velocity $\vec{v}_C$ and average angular velocity $\vec{\omega}_C$ can also be calculated [10].

For the purposes of generating a predictive balance point, we select a pair of planes (one vertical and one horizontal) which approximate the motion of the 3D model. The planar inertia of the simple model in both of these planes is estimated using the $\tilde{I}_C$ matrix (similar to the projection used to generate the 3D FPE in [20] and Chapter 3).

The height of the horizontal plane and the location and rotation of the vertical plane require the definition of an appropriate anchor point for the simplified model. For the SFPE, the anchor point $\hat{A}_0$ is chosen to represent the location of maximum effectiveness for the Centroidal Moment Pivot point (CMP), as shown in Figure 4.2.

As discussed in [11], the dynamics of the Divergent Component of Motion (DCM) can be controlled using the CMP, as the DCM always moves directly away from the CMP. Therefore, placing the CMP on the edge of a biped's convex support polygon (SP) in the direction of $\vec{v}_C$ will produce maximum deceleration of the COM (if the DCM is above the SP). Similiarly, as described in [15], once the instantaneous Capture Point (ICP) has exited a biped's SP, maintaining the CMP as close as possible to the ICP is found to minimize the number of steps required to recover.

Unlike the COP or projected COP, which is used in the GFPE and may be discontinuous, this choice of anchor point will vary smoothly with changes in $\hat{C}$ and $\vec{v}_C$, and is more likely to be the point around which $\hat{C}$ will be purely rotating once a step is needed. When a step is required, this anchor point also provides the ability to correct for small disturbances (internal or external) without affecting the predicted SFPE point (assuming the stance foot is fixed), by moving the COP around within the support polygon.



(a) How to choose $\hat{A}_0$ when the DCM is above the SP.

(b) How to choose $\hat{A}_0$ when the DCM is not above the SP.

Figure 4.2: Illustrations of how to select the anchor point location, $\hat{A}_0$. When the DCM, $\hat{\xi}$, is above the biped's convex support polygon (SP), $\hat{A}_0$ is chosen as the (furthest) intersection of the perimeter of the SP with a ground-projected vector from $\hat{C}$ through $\hat{\xi}$, as shown in (a). If $\hat{\xi}$ is not above the biped's SP, $\hat{A}_0$ is chosen as the point on the edge of the SP closest to the ground projection of $\hat{\xi}$, as shown in (b).

In the development of the SFPE, several specific instants in the progression of the simplified model's state ($x = [\theta\ \omega\ \gamma]$) are used, and are labeled with the following subscripts:

0 The current state of the model,

$-$ Just before the landing foot impacts the ground,

$+$ Just after the landing foot impacts the ground, and

$*$ The final state of the model

As shown in Figure 4.3, there are several possible state transitions when using the SFPE. If a step is not required, an SFPE point is defined within the biped's SP which can be used to predict COM motion (as discussed in Section 4.1.1). Otherwise, finding the SFPE point requires a set of equations which relate the current state of the biped (x_0) to its final state (x_*), which are separated into three categories:

- Pre-impact equations, relating x_0 to x_-;
- Impact equations, relating x_- to x_+; and
- Post-impact equations, relating x_+ to x_*.

These three sets of equations are defined in Sections 4.1.2 to 4.1.4, using conservation of energy and/or momentum, as appropriate. The SFPE point $\hat{S}$ is then found as the point which satisfies these equations and results in the desired final state. In these examples, this desired final state is when the COM ($\hat{C}$) is at rest above the SFPE point ($\hat{S}$).

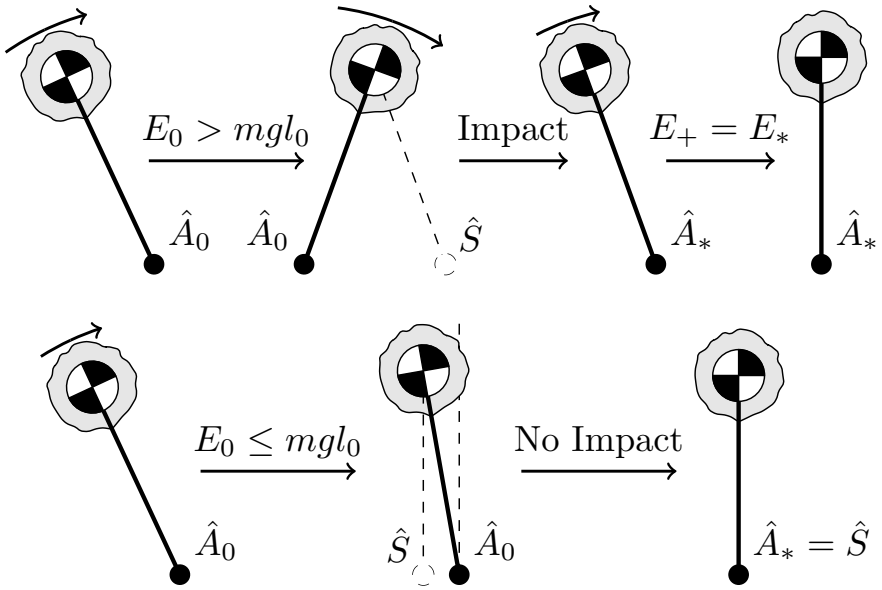


Figure 4.3: Illustration of the possible state transitions of the SFPE model. The top row illustrates the case where the initial energy is high enough to overcome the potential energy well created by the rigid leg, requiring a step. In this case, the legs are assumed to be equal in length and a balance point, $\hat{S}$, is determined which, when stepped on by a swinging foot, will result in $\hat{C}$ stopping above $\hat{S}$. If the energy is not high enough to require a step, the bottom row of state transitions applies. In this case, the swinging leg length is $l_* = E_0/mg$ and therefore is only equal to l_0 when $E_0 = mgl_0$.

4.1.1 Core Equations

The rotational inertia of the model is estimated for the two planes introduced above, where I_C^ω and I_C^γ are defined as the scalar projections of $\tilde{I}_C$ onto vectors normal to the vertical ω and horizontal γ planes, respectively. Using these definitions, the angular momenta estimates about $\hat{A}$ (normal to the two planes) are

$$k^\omega(\omega) = (I_C^\omega + ml^2)\omega = I^\omega\omega \tag{4.1}$$

$$k^\gamma(\theta, \gamma) = (I_C^\gamma + ml^2\sin^2(\theta))\gamma = I^\gamma(\theta)\gamma \tag{4.2}$$

The estimated kinetic energy $T(\theta, \omega, \gamma) = T^\omega(\omega) + T^\gamma(\theta, \gamma)$ can therefore be written as

$$T = \frac{1}{2}I^\omega\omega^2 + \frac{1}{2}I^\gamma\gamma^2 = \frac{(k^\omega)^2}{2I^\omega} + \frac{(k^\gamma)^2}{2I^\gamma} \tag{4.3}$$

Writing the potential energy as $U(\theta) = mgl\cos\theta$, we can then write the system's total estimated energy as

$$E(\theta, \omega, \gamma) = T^{\omega}(\omega) + T^{\gamma}(\theta, \gamma) + U(\theta) \tag{4.4}$$

If $\hat{C}$ is above the support polygon, then there is the possibility that a given disturbance doesn't require the biped to step, but simply to shift its CMP. If the current estimated energy of the model, E_0, is no more than the maximum potential energy ($E_0 \leq mgl_0$), then no step is required (assuming full control over the CMP) [109].

In this case, the balance point $\hat{S}$ is defined as the closest point the COM ground projection $\hat{G}$ will reach, relative to the edge of the SP, if the CMP is held at $\hat{A}_0$. This is found by setting the final leg length l_* to the maximum height $\hat{C}$ will reach above $\hat{A}_0$, $l_* = E_0/mg$, and applying trigonometry to determine the distance d_{min} from $\hat{G}$ to $\hat{A}_0$ at that height (using the constant leg length l_0 between $\hat{C}$ and $\hat{A}_0$). S is the point between the current $\hat{G}$ and $\hat{A}_0$ which is d_{min} away from $\hat{A}_0$ (see the bottom row of Figure 4.3).

If a step is required (i.e., $E_0 > mgl_0$), then the three sets of impact equations are required to determine where to step to achieve the desired final state. To simplify these equations, it is assumed that the model's pre- and post-impact leg lengths are equal ($l_0 = l_* = l$) and therefore that all of the above equations can be used at all instants in question.

As was done in the original FPE method (and its extensions), at each time step the current 3D inertia of the multibody system is used to estimate the planar inertia(s) of the simplified model. For the SFPE, this means the planar inertia estimates about the axes normal to the two planes of motion (I_C^{ω} and I_C^{γ}) are assumed to be constant at the three critical instants for prediction purposes. The predicted location of the SFPE balance point is recalculated at each time step using the current inertial and dynamic properties of the multibody system, instead of making one prediction just after the disturbance.

4.1.2 Pre-Impact Equations

Pre-impact equations enable prediction of where the biped will need to step at some point in the future to recover its balance, by defining x_- in terms of the current state, x_0.

In [18] and [20], it was assumed that a foot could be instantaneously placed anywhere on the ground, and therefore that $x_0 = x_-$. In [14] and [16], conservation of energy was used to determine the pre-impact equations, although they assumed vertical planar rotation about the COP (or its projection) and ignored the biped's rotational inertia.

The assumption of spherical motion about $\hat{A}_0$ leads to a formulation using conservation of energy and conservation of angular momentum to determine the pre-impact state, x_-.

Since gravity cannot create a torque about a vertical axis and no active torque is applied about the vertical axis, k^γ around that axis is a conserved quantity. Based on this, the angular momentum around the vertical axis at $\hat{A}_0$ prior to impact is assumed to be constant: $k_0^\gamma = k_-^\gamma$.

Defining $T_C^\gamma = (k_0^\gamma)^2/2I_C^\gamma$, we can rewrite $T^\gamma(\theta_-)$ as

$$T_-^\gamma = \frac{(k_-^\gamma)^2}{2I_-^\gamma} = \frac{(k_0^\gamma)^2}{2I_C^\gamma \eta_-} = \frac{T_C^\gamma}{\eta_-} \tag{4.5}$$

where $\eta(\theta) = 1 + (ml^2/I_C^\gamma)\sin^2\theta$ and $\eta_- = \eta(\theta_-)$.

Conservation of energy is used as the basis of the pre-impact equations by setting the pre-impact energy equal to the current energy (i.e., $E_0 = E_-$). We assume that the values of I_C^ω and I_C^γ, in their respective planes, will have the same value at impact as their current values, and that all current values (those with subscript 0) are known. The only remaining unknowns in the pre-impact equation are the impact angle θ_- and its derivative ω_-.

This leads to an equation for T_-^ω in terms of θ_-, known values E_0 and T_C^γ, and constants:

$$T_-^\omega = E_0 - \frac{T_C^\gamma}{\eta(\theta_-)} - U(\theta_-) \tag{4.6}$$

4.1.3 Impact Equations

Due to the loss of energy during impact, conservation of angular momentum is used to generate the impact equations to relate the pre-impact kinetic energy, T_-, to the post-impact kinetic energy, T_+. The constants used to generate the pre-impact equations are also assumed to remain constant across impact, since there is effectively no change in the model's dynamics other than an instantaneous change in velocity.

The vertical plane in which impact will occur is assumed to currently include both $\hat{A}_0$ and the DCM $\hat{\xi}$ (and, of course, the SFPE balance point $\hat{S}$), but not necessarily $\hat{C}$, as shown in Figure 4.4. This ensures that $\hat{S}$ is calculated relative to where $\hat{C}$ is heading, as opposed to its current location, as the COM is attracted towards the DCM according to $\hat{\xi} = \hat{C} + b\vec{v}_C$, where b is the time constant of the DCM dynamics [11].

Note that the impact and post-impact equations are being calculated before the future anchor foot lands (i.e., as part of a prediction), so the predicted landing foot location maintains its label ($\hat{A}_*$). In the vertical impact plane, the angular momentum around $\hat{A}_*$

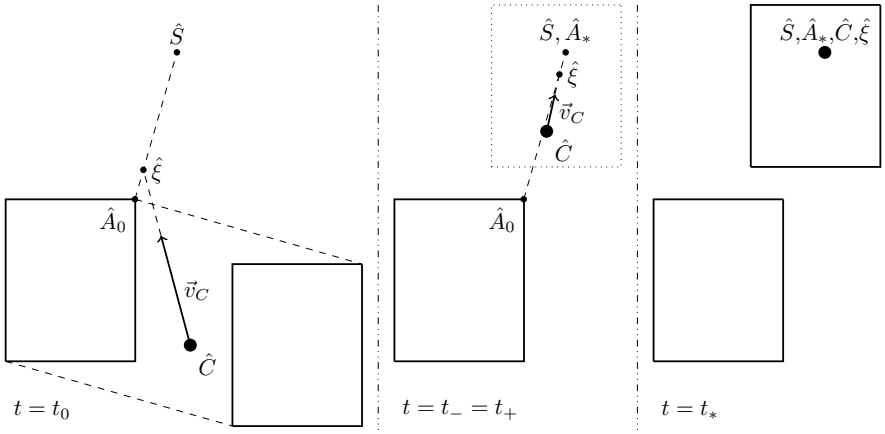


Figure 4.4: Illustrations showing a bird's eye view of the vertical plane of impact and the points $\hat{C}$, $\hat{A}_0$, $\hat{S}$, and the DCM $\hat{\xi}$. It is clear that although $\hat{C}$ is not currently in the impact plane defined by $\hat{A}_0$ and $\hat{\xi}$, that it will be located in the plane (or very close to it) once impact occurs.

just after impact will be k_+^ω. However, since the model is assumed to be purely rotating about $\hat{A}_0$ just prior to impact and about $\hat{A}_*$ just after impact, $k_+^\omega \neq k_-^\omega$.

To relate k_+^ω to k_-^ω, the standard FPE equations for conservation of angular momentum at impact are used [18]. As shown in Figure 4.5, due to the geometry of the model the equality $v_+ = \cos(2\alpha)v_- = \cos(2\alpha)lw_-$ holds, so k_+^ω can be written in terms of ω_- as

$$k_+^\omega = I_C^\omega \omega_- + mlv_+ = \left(I_C^\omega + ml^2\cos(2\alpha)\right)\omega_- \tag{4.7}$$

This leads to the dimensionless impact ratio $\zeta(\alpha)$ relating the pre- and post-impact angular momenta in the ω plane:

$$\zeta = \frac{k_+^\omega}{k_-^\omega} = \frac{(I_C^\omega + ml^2\cos(2\alpha))\omega_-}{I^\omega \omega_-} \tag{4.8}$$

Therefore, T_+^ω can be written in terms of T_-^ω as:

$$T_+^\omega = \frac{(k_+^\omega)^2}{2I^\omega} = \frac{(\zeta k_-^\omega)^2}{2I^\omega} = \zeta^2 T_-^\omega \tag{4.9}$$

Using a similar method, the dimensionless inertia ratio $\psi(\theta_-, \theta_+)$ can be written as (using $d(\theta) = l\sin\theta$):

$$\psi = \frac{k_+^\gamma}{k_-^\gamma} = \frac{(I_C^\gamma - md_+d_-)\gamma_-}{I_-^\gamma \gamma_-} \tag{4.10}$$

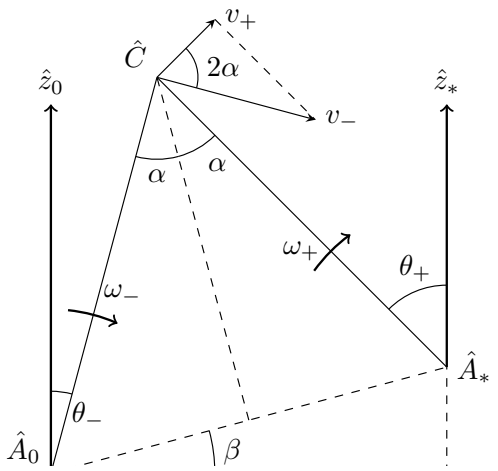


Figure 4.5: Illustration of relationship between various angles and points in the impact plane at the moment of impact (assuming $\hat{C}$ is in plane during impact). Since the leg length is assumed constant, the distance from $\hat{A}_0$ to $\hat{C}$ is equal to the distance from $\hat{A}_*$ to $\hat{C}$. Note that v_- is perpendicular to CA_0, while v_+ is perpendicular to CA_*.

Combining this with the assumption of conservation of angular momentum about the axis normal to the horizontal plane leading up to impact (i.e., $k_0^\gamma = k_-^\gamma$) enables us to write T_+^γ as:

$$T_+^\gamma = \frac{(k_+^\gamma)^2}{2I_+^\gamma} = \frac{(\psi k_0^\gamma)^2}{2I_C^\gamma \eta_+} = \psi^2 \frac{T_C^\gamma}{\eta_+} \tag{4.11}$$

4.1.4 Post-Impact Equations

After impact, conservation of energy can again be applied by assuming that the model remains in a fixed configuration and that the desired final angle θ_* and final angular velocity ω_* are known. This builds on the assumptions that the model is in pure rotation about $\hat{A}_*$ after impact, and that the model's fixed parameters (l, I_C^ω, etc) remain constant.

Based on these assumptions, the model's post-impact equations can be determined in the same way as the pre-impact equations. Conservation of angular momentum about the axis normal to the horizontal plane can be used again to define the equality $k_+^\gamma = k_*^\gamma$. Using the pre- and post-impact conservation of k^γ and the inertia ratio ψ, we can then write T_*^γ as

$$T_*^\gamma = \frac{(k_*^\gamma)^2}{2I_*^\gamma} = \frac{(\psi k_0^\gamma)^2}{2I_C^\gamma \eta(\theta_*)} = \psi^2 \frac{T_C^\gamma}{\eta_*} \tag{4.12}$$

which, along with the desired values of θ_* and ω_*, allows us to calculate the final estimated energy $E_* = T_*^\omega + T_*^\gamma + U_*$.

This leads to the post-impact equation, very similar to the pre-impact equation (4.6), for T_+^ω in terms of only θ_-, θ_+, known values E_* and T_C^γ, and constants:

$$T_+^\omega = E_* - \psi^2(\theta_-, \theta_+)\frac{T_C^\gamma}{\eta(\theta_+)} - U(\theta_+) \tag{4.13}$$

By combining the pre- and post-impact equations (4.6) and (4.13) with the ratios ζ and ψ, the following energy-based SFPE equation in terms of α, θ_-, and θ_+ is produced:

$$E_* - \psi^2\frac{T_C^\gamma}{\eta_+} - U_+ = \zeta^2\left(E_0 - \frac{T_C^\gamma}{\eta_-} - U_-\right) \tag{4.14}$$

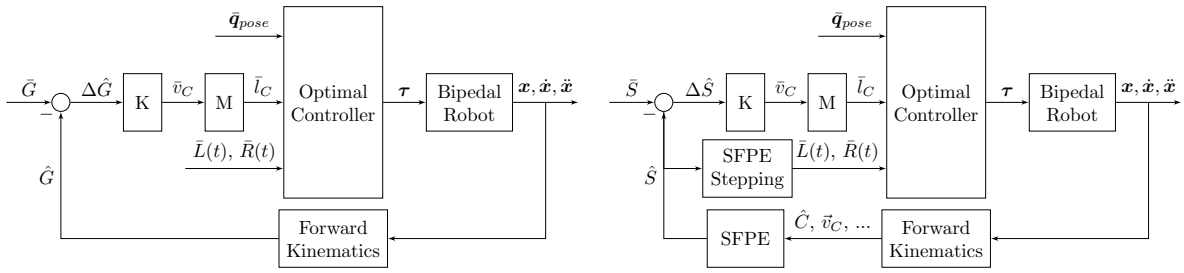
To determine the location of $\hat{S}$, the θ_- and θ_+ values must be redefined to allow this equation to be in terms of one common angle. As shown in Figure 4.5, these angles can be defined as $\theta_- = \alpha - \beta$ and $\theta_+ = \alpha + \beta$, where β is the angle between $\hat{S}$ and a horizontal plane, measured at $\hat{A}_0$. For a horizontal planar ground surface, it is easy to show that $\beta = 0$ at all instants, and therefore that $\theta_- = \theta_+ = \alpha$.

For a planar, but not necessarily horizontal, ground surface, β can be easily determined based on the impact plane and the value of α. To determine the angle β for intersecting planar ground surfaces, the methods discussed in [14] can be used to define β as a function of α and the surface slopes. In general, it is assumed β can be defined as a function of α, leading to an equation whose only variable is α.

Solving the SFPE equation for α and applying trigonometry in the impact plane gives the location of $\hat{S}$, the SFPE balance point, with respect to the current anchor point $\hat{A}_0$.

In general, we assume that the final desired state consists of the COM ($\hat{C}$) held directly above the final anchor point ($\hat{A}_*$), by setting $\theta_* = 0$ and $\omega_* = 0$. Based on this assumed final state, the final estimated energy E_* is the sum of the final potential energy, $U_* = mgl$, and the final kinetic energy, $T_* = T_*^\gamma = \psi^2 T_C^\gamma$.

Although the SFPE was developed with this specific final state in mind, an alternative final state only requires modifications to the values of θ_* and ω_*. For example, a final state which includes a desired velocity for the COM (e.g., for gait generation purposes) requires setting $\omega_* \neq 0$, and would likely require $\theta_* \neq 0$ as well. See [18] and Chapter 3 for examples of the application of the FPE and its extensions for the generation of bipedal gait.



(a) COM-Based Control System Diagram

(b) SFPE-Based Control System Diagram

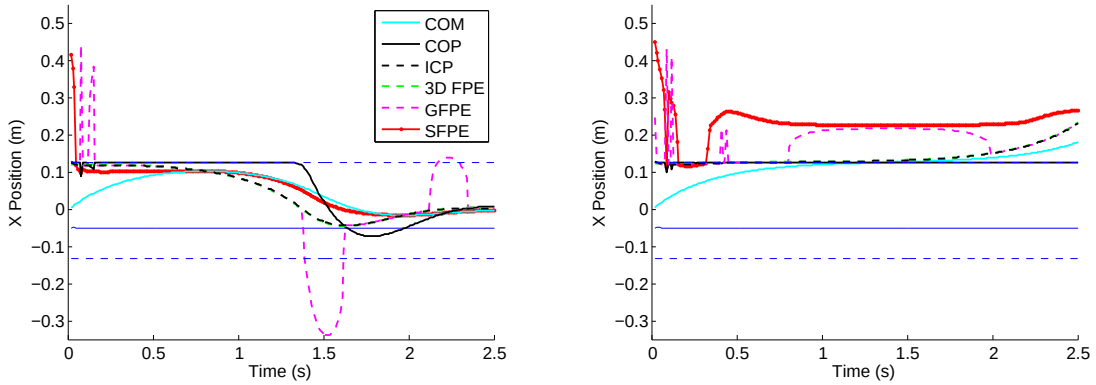
Figure 4.6: System diagrams for the COM-based and SFPE-based control systems. The labelled signals are: $\boldsymbol{\tau}$ for torque, $\vec{v}_C$ for COM velocity, $\vec{l}_C$ for linear momentum, $\hat{G}$ for the ground projection of $\hat{C}$, S for the SFPE balance point, $\boldsymbol{q}_{pose}$ for postural joint angles, $\hat{L}(t)$ and $\hat{R}(t)$ for the foot trajectories, and $\boldsymbol{x}$ for the state of the full bipedal robot (joint angles and 6-DOF floating base). Dots over a variable signify a derivative, a bar over a variable signifies a desired value, and a Δ in front of a variable signifies an error. In both of these systems, the optimal controller from [112] is augmented with a linear momentum input, which is calculated using a proportional gain (K = 5) on an error term. As shown in (a), for the comparisons in Figure 4.7 the COM ground projection error was used. For the comparisons in Figure 4.8, the controller in (b) was used, where the SFPE error was used to generate the linear momentum reference and an SFPE-based stepping controller was included.

4.2 Comparison

To clearly show the differences between the SFPE and existing balance point estimators, a simulation of Boston Dynamics' Atlas robot [110] in the MATLAB toolbox Drake [111] was subjected to initial instantaneous velocity disturbances. These initial velocities are similar to the disturbances used in [16] to compare two balance points and the instantaneous velocity changes applied as disturbances in [11]. The optimal controller from [112] (the default Atlas controller in Drake) was slightly modified to take a desired linear momentum as an additional control reference, to be tracked using PD gains on the linear momentum error (see Figure 4.6a).

The SFPE is formulated to handle piecewise flat ground, similar to the GFPE. However, the results in this chapter have been generated for flat ground, by setting $\beta = 0$ everywhere. Also, the upper body joints (back, arms, neck) were held static, to allow fair comparisons between criteria that assume a point mass at the COM and those with rotational inertia.

The desired linear momentum of the COM was set in the same way as in [8–10], with damping on the COM velocity and the desired COM set to the center of the single supporting foot. The SFPE point and other balance points are calculated and graphed



(a) X Disturbance of 0.410 m/s

(b) X Disturbance of 0.420 m/s

Figure 4.7: Comparison between the SFPE and other balance points for an Atlas robot [110] when the pelvis is subjected to initial velocity disturbances in the X direction. The values of the velocity disturbances were chosen to illustrate the differences between the balance points and hold no special significance. Smaller velocities cause similar patterns to Figure (a), with the COM remaining closer to the foot center, while larger velocities cause the COM to exit the foot faster. The robot is standing on one foot throughout each simulation, with the ground projection of the ankle shown by a thin blue line, while dashed blue lines show the toe (top) and heel (bottom) of the foot. Since the angular momentum and COM velocity are damped, the 3D FPE point and ICP are approximately equal. Since it has no concept of the support polygon, the GFPE point sometimes predicts a step is required when the system can recover by leaning. Anytime the GFPE does not predict a step is required, the GFPE point is not clearly defined. Here we define it using similar methods as used for the SFPE (see the bottom row of Figure 4.3). In these simulations, the robot is controlled using the COM-based controller shown in Figure 4.6a. In (a), the COM-based controller is able to compensate for the disturbance. Since no step is required, the SFPE predicts the location at which the COM could stop, assuming the COP is used to maximum effect (i.e., placed at $\hat{A}_0$). Note the SFPE and GFPE points start outside of the foot in both cases, due mainly to their initial kinetic energy assumptions being violated right after the initial disturbance. In (b), the COM-based controller is not able to compensate for the disturbance, and a step is required. Although the SFPE point returns to the foot temporarily, suggesting that a better controller may be able to avoid stepping, it then leaves the SP and quickly predicts a stepping location to recover from the given disturbance. Note that in (b) the 3D FPE point and ICP do not leave the SP until 0.6 s and 0.8 s, respectively, and the GFPE point does not reliably predict a step is needed until 0.8 s and then stops just before 2 s. Also, due to the SFPE including rotational inertia, it predicts a larger step is needed than the GFPE, which only considers a point mass.

at each time step.Figure 4.7 illustrates the results and highlights the differences between the balance points: The ICP is purely a function of the COM kinematics and gravity, and therefore is not directly influenced by inertia, angular momentum, or any of the internal system kinematics. The 3D FPE also makes no assumptions about the system's kinematics, but includes an impact model, rotational inertia, and angular momentum in addition to the COM kinematics and gravity.

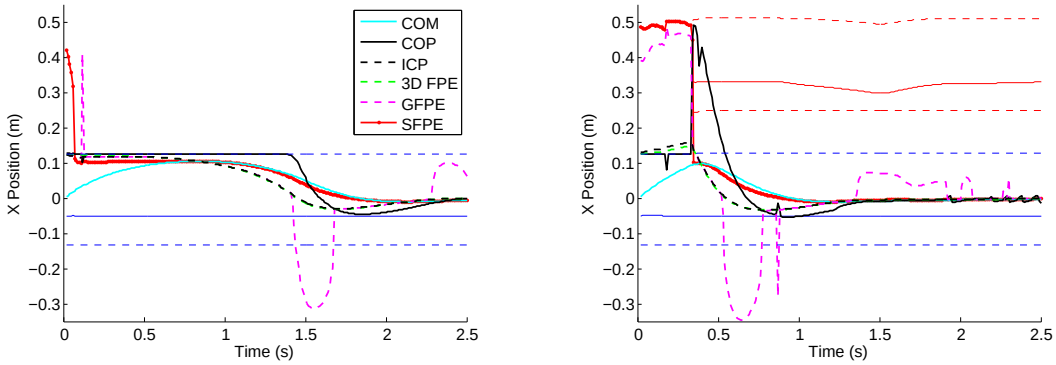
This effectively means the 3D FPE is an extended form of the ICP, which includes the system's rotational properties and the energy losses due to impact. In these initial comparisons, due to regulation of angular momentum, the 3D FPE tracks along with the ICP very consistently, confirming this. In general, it is apparent that the exit of the ICP or 3D FPE from the support polygon can be used as a good indicator that the biped will need to take a step in the near future.

However, if a step is required, only the SFPE and GFPE are predictive and can therefore provide a desired landing location ahead of time for the motion planner to generate a suitable trajectory for the swing foot. Although a predictive CP was discussed in [16], it was shown to be a worse step indicator than their predictive FPE-based solution, so was left out of this comparison. Capture regions [15] are also a predictive extension of the ICP, but require flat ground and supply a region in which to step based on a minimum step time instead of a specific location based on COM motion.

As predictive models, the GFPE and SFPE use an anchor point to define the pre-impact equations of motion, which directly influences their associated balance points. The GFPE anchor point is at the orthogonal projection of the COP into the vertical plane used for the GFPE model (described in Section 2.2). This means the GFPE anchor point moves due to changes in the relative locations of the ICP, COM, or COP, which causes the GFPE point in Figures 4.7b and 4.8a to falsely predict a step is needed when the biped is balanced. This is also the cause of the fluctuations in the GFPE in the figures.

The GFPE equations are not clearly defined when the initial energy is lower than the model's peak potential energy. In this chapter, we assume this no-step-required GFPE point is defined similarly to the SFPE when no step is required.

Instead of using the projected COP, the SFPE anchor point moves along the edges of the stance foot, which is kinematically fixed (barring foot rotation or slipping) and moves smoothly. Also, when no step is required, an alternative set of equations are defined for its use as a predictive COM reference signal (see Subsection 4.1.1). During this low energy portion of motion, the SFPE predicts where the COM will stop, assuming a COP at $\hat{A}_0$. Unlike the 3D FPE or GFPE, which only consider energy in a vertical plane, the SFPE also takes into consideration the kinetic energy due to motion about a vertical axis, which causes the model's kinetic energy to be both smoother and closer to the real kinetic energy.



(a) X Disturbance of 0.420 m/s

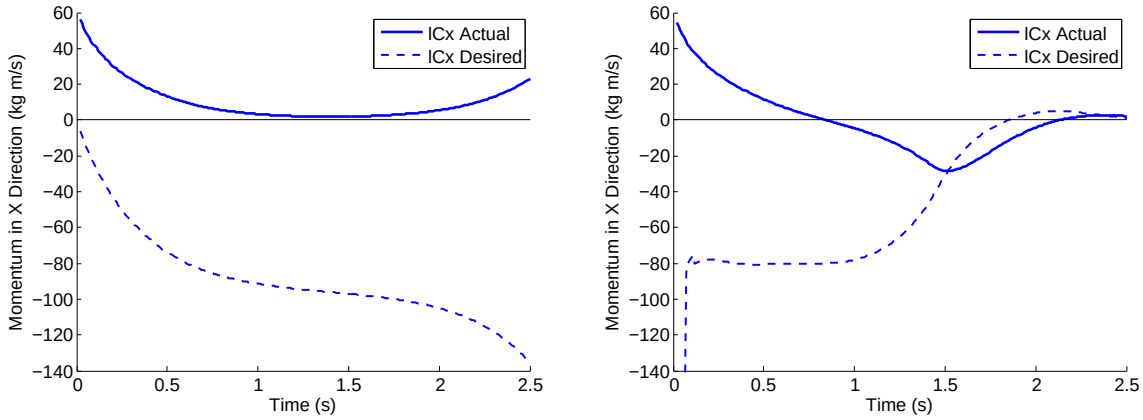
(b) X Disturbance of 0.430 m/s

Figure 4.8: Comparison between the SFPE and other balance points when the pelvis of an Atlas [110] is subjected to large initial velocity disturbances in the X direction, and an SFPE-based controller is used instead of the COM-based controller. Immediately after the disturbances, the SFPE is far from its desired location (the foot center), which causes the controller to generate a desired COM velocity which will bring the SFPE back towards its desired location (as shown in 4.6b). Note that other than changing the method for generating the reference momentum, the conditions of (a) are identical to those in Figure 4.7b, including standing in single support at rest before the disturbance. Again, the SFPE and GFPE points start outside of the foot due to their initial kinetic energy assumptions being violated right after the instantaneous velocity disturbance. However, in (b) the SFPE point maintains its position outside the foot over several time steps, so a step is required. The SFPE point is used as a control reference to place the raised foot on the ground (as shown at approximately 0.3 s into the simulation, using dashed red lines for the extents of the right foot and a thin red line for the ground projection of the right ankle). Note that in this case, the COP immediately moves to the toes of the landed foot, moving the SFPE point (and the DCM and COM) back towards the desired location at the center of the original support polygon (the left foot). For larger velocities, a longer step is required but the behavior is qualitatively similar.

4.3 SFPE-Based Controller

As a proof of concept, we demonstrate that the SFPE can be used for both leaning and stepping using an example of an SFPE-based controller. This controller uses a proportional gain on the error between the current and desired SFPE points (at each time step) to generate a desired COM velocity, which is then multiplied by the mass of the robot to produce a linear centroidal momentum reference (see Figure 4.6b).

If the desired SFPE is set as the ground projection of the desired COM location, the key difference between this controller and the COM-based controller is the error term used to generate this linear momentum reference. The new controller uses the error between the predicted future COM ground projection (i.e., the current SFPE) and the desired COM ground projection, instead of using the error between the current and desired projections.



(a) Using COM-Based Controller

(b) Using SFPE-Based Controller

Figure 4.9: Comparison between the linear momentum behavior for the COM- and SFPE-based controllers. The results in (a) correspond to Figure 4.7b, while those shown in (b) correspond to Figure 4.8a. As shown here, since the SFPE predicts where $\hat{G}$ will come to rest if the COP is placed at $\hat{A}_0$, it can be used to preemptively move the COM backwards.

As shown in Figure 4.8, this momentum reference signal enables the robot to respond to stronger disturbances without losing its balance. The only difference between Figures 4.7b and 4.8a is the formulation of the momentum reference which is provided to the CMM controller. This is clearly shown in the linear momentum graphs of Figure 4.9, where the immediate reaction of the SFPE-based controller (Figure 4.6b) allows the biped to remain balanced, while the delayed reaction of the COM-based controller (Figure 4.6a) does not.

The SFPE controller's response to a larger disturbance is shown in Figure 4.8b, where even though the controller reacts immediately, the limitations of balancing solely using the COP are exceeded and, in the absence of other strategies, a step is needed. When taking a step, the controller uses the SFPE point as a control reference to help determine where to place the swinging foot. As shown in Figure 4.8b, the foot is placed to ensure that the COP can be used to move the SFPE back into the original support polygon.

Note that the robot itself is not constrained to the assumed motion of the simplified pendulum, and in fact departs significantly from the 3D pendulum motion predicted using the SFPE's internal model. Evidence of this is clearly shown at the beginning the graphs in Figures 4.7 and 4.8, where the SFPE and GFPE points are initially located outside the foot just after the disturbance due to violations of their kinetic energy assumptions.

Much like the FPE point, the SFPE point serves only as an estimate (albeit a reasonably good one) of where the full multibody system should step to come to rest. A dual version

of the SFPE with an anchor point at each of the robot's ankles and a corresponding stepping point for each foot was also developed, but was found not to perform as well as the support-polygon-based SFPE that was used in this chapter.

4.4 Discussion

The SFPE evaluates the current system dynamics in 3D by finding the current centroidal momentum and inertia and the overall energy of the system. It was designed to address the drawbacks of existing balance point estimators, by incorporating and extending desirable features of the GFPE, 3D FPE, and other methods discussed in Section 2.2.

Like the GFPE, the SFPE is predictive and can be applied to piecewise planar ground surfaces. Unlike the GFPE or the predictive estimators in [16], the SFPE also defines a balance point when it does not predict that a step is required, which can be used for leaning control. The SFPE also includes knowledge about the support polygon, instead of using the projected COP anchor point of the GFPE, avoiding false indications that stepping is required when it is not.

Unlike the more common ICP, FPE, and 3D FPE, the SFPE is predictive, which enables it to estimate both when a step is required and where to step ahead of time, allowing time to plan a swing foot trajectory. Although we have previously used the 3D FPE for the control of a biped in Chapter 3, the predictive nature of the SFPE and its inclusion of non-flat ground make it the better choice in most cases. This prediction also incorporates rotation about a vertical axis, which is a novel estimator feature not found in the literature.

The main limitation of the SFPE approach is due to the estimation of the inertia in the two assumed planes of motion, ignoring any intrinsic rotation of the system about the COM, coupled inertia terms, or changes in centroidal inertia, and the assumption of a constant leg length in the simplified model. As discussed in [20], a numerical sensitivy analysis can be conducted to judge the effects of these assumptions.

The original FPE, on which the SFPE and 3DFPE are based, was found to be insensitive to changes in leg length, moment of inertia, or overall energy [20]. A direct effect of these assumptions is the location of the SFPE point being outside of the foot when the system is subject to a large initial velocity disturbance (as seen in Figures 4.7 and 4.8).

Some insight into the behavior of a biped at impact can be gained by analyzing the $\zeta(\alpha) = k_+^\omega / k_-^\omega$ inertia ratio of equation (4.8). The equation can be further simplified, using the trigonometry identity $\cos(2\alpha) = 1 - 2\sin^2\alpha$, to

$$\zeta(\alpha) = 1 - 2(ml^2/I^\omega)\sin^2\alpha \tag{4.15}$$

By assuming a maximum leg separation of $\pi/2$, the range of α is restricted to $0 \leq \alpha \leq \pi/4$, which leads to the inequality: $0 \leq 2\sin^2\alpha \leq 1$. In other words, the angular velocity of the biped is reduced during impact by the product of the positive inertia ratio, $ml/I^{\omega} \leq 1$, and a positive scaling factor, $2\sin^2\alpha \leq 1$. Therefore, $0 \leq \zeta(\alpha) \leq 1$.

Effectively, this means that for a given biped which can be approximately modeled as above, the loss of energy at impact is purely a function of the separation angle of the legs, 2α. It also means that if an impulsive force (or an approximation of one) is generated by the biped's stance leg at the moment of impact, which produces an increase in ω_+ equal to the loss due to impact above, the effects of impact on the biped could be ignored. Since the effects of impact are routinely assumed to be negligible in many humanoid control strategies, the equations above could be used to develop a simple controller which might enforce the validity of the lossless-impact assumption.

Finally, a number of existing controllers can benefit from the SFPE, particularly those without the capability to generate footsteps online or which do not control momentum. For example, the optimal controller from [112] (Drake's default Atlas controller) was augmented with linear momentum feedback using the SFPE (see Figure 4.6b), enabling an easy combination of their pre-planned movements with online compensation while adding minor additional complexity.

4.5 Summary

In this chapter, a novel balance point estimator called the Spherical Foot Placement Estimator has been introduced. It has been formulated to overcome a number of drawbacks of existing balance point estimators, by combining and extending desirable features of several different balance points.

The SFPE was compared to other balance point estimators, and was shown to outperform them by providing recovery step location prediction and momentum objectives with smooth dynamics. An SFPE-based feedback loop was used in a momentum-based controller as an example of how to add leaning to an existing whole-body controller and a dynamic SFPE-based stepping strategy was used to deal with large disturbances, demonstrating its utility in bipedal control.

So far in this thesis, several performance metrics related to balancing and stepping controllers were considered. Specifically, balance point estimators and the points they generate were applied and compared. In the remaining chapters, performance metrics related to a system's physical structure and dynamics will be investigated.

Chapter 5

Generalized Gains and Optimization[1]

The ability of a biped to balance and walk is impacted by both the physical properties of the biped and the controller used to achieve the desired behavior(s). Much of the existing work discussed in Chapter 2, along with the novel research described in Chapters 3 and 4, has focused on developing methods for achieving dynamic gait by using Balance Point estimators as part of a biped's controller. However, as discussed in Section 2.4, only limited research has focused on how to quantify a mechanism's inherent balancing capabilities, or how to modify the mechanism to improve these capabilities.

For legged systems, achieving balance and locomotion depends on the system's capability to effectively and efficiently move its Center of Mass relative to its contact point(s). The two main existing methods for achieving this kind of quantified measurement of a mechanism's abilities are Azad's dynamic COM manipulability [23] and Featherstone's velocity and momentum gains [27], as discussed in Section 2.4.

This chapter starts by introducing notation to simplify impulsive dynamics equations with passive contact (including the existing velocity and momentum gains) and then uses this simplified notation to build on Featherstone's work by:

- Extending the momentum gains from their existing 2D, 2-link inverted pendulum formulation to general 2D and 3D models;
- Defining two methods for calculating momentum gains for these general models; and
- Relating the velocity and momentum gains of a system to its centroidal momentum (a commonly used measure of aggregate system behavior).

[1]Earlier versions of portions of this chapter were previously published in [113–115].

These general momentum gains inherit the corresponding velocity gains' independence from the applied control scheme and invariance properties, in addition to including a consideration of the inertial properties of the system. Effectively, they measure how efficiently an articulating system balancing on a passive (point or rolling) contact can move its COM via actuated joint motions. These properties make the momentum gains a good choice for analyzing the balance capabilities of a given mechanism, or as a criterion for optimizing the design of balancing mechanisms.

In Section 5.1, simplified notation is introduced and used to re-formulate Featherstone's velocity gain matrix definitions before the momentum gain definitions are extended to the general 3D case in Section 5.2. This includes a discussion of the two different methods for calculating the momentum gains, namely the Augmented Inertia Method in Section 5.2.1 and the Spatial Method in Section 5.2.2. The spatial gain defined as part of the Spatial Method is then used to draw comparisons between a system's velocity and momentum gains and its centroidal momentum in Section 5.2.3 before a brief discussion of controls applications and limitations of the general momentum gains in Section 5.4.2.

Building on this foundation, a generalized gain is defined in Section 5.3 which can be used to broadly characterize a system's ability to move its COM. All of the existing and novel velocity and momentum gains are shown to be special cases and/or subsets of this generalized gain. This generalized gain formulation is also used to directly show that the differences between velocity and momentum gains are entirely a function of scaling.

Building on these gain definitions, a generalized optimization framework for performing optimal mechanism design is then defined in Section 5.4. This framework uses gains as the core of its objective functions (as described in Section 5.4.1) to design a mechanism which maximizes its ability to move its COM throughout a given subset of the mechanism's configuration space, without having to define an associated controller or trajectory. The generalized objective function uses weighted matrix norms to enable the application of domain knowledge in the parameterized optimization of 2D and 3D mechanisms. The results of the application of this optimization framework to a set of simple planar mechanisms, a 5-link planar biped, and a 5-link 3D biped are discussed in Chapter 6.

[1]In this context, balance is assumed to be primarily a function of COM motion. Angular momentum about the COM is assumed to be regulated.

5.1 H-Bar Notation

To simplify the formulation of impulsive dynamics equations which include an initial passive joint, an augmented joint space inertia matrix notation is introduced. This notation is used to denote the relationship between an impulse and the associated impulse response in a given configuration (i.e., $\vec{\iota}_0 = \bar{\boldsymbol{H}}_{0a}(\boldsymbol{q})\Delta\dot{\boldsymbol{q}}_a$ or $\boldsymbol{\iota}_a = \bar{\boldsymbol{H}}_{aa}(\boldsymbol{q})\Delta\dot{\boldsymbol{q}}_a$).

In general, this notation is written as follows (with $\bar{\boldsymbol{H}}_{\beta\alpha} = \bar{\boldsymbol{H}}_{\beta\alpha}(\boldsymbol{q})$ and $\boldsymbol{H}_{\beta\alpha} = \boldsymbol{H}_{\beta\alpha}(\boldsymbol{q})$):

$$\bar{\boldsymbol{H}}_{\beta\alpha} = \boldsymbol{H}_{\beta\alpha} - \boldsymbol{H}_{\beta p}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{p\alpha} \tag{5.1}$$

where α and β can each be either a single joint index or a range of indices provided α is a subset of the actuated joint indices, a. In 2D, for a planar system with only one passive rotational contact joint (i.e., where $p = \{1\}$), $\boldsymbol{H}_{pp}$ is a scalar, giving $\boldsymbol{H}_{pp}^{-1} = H_{11}^{-1} = 1/H_{11}$.

This simplifies the notation for the impulsive change in COM velocity (due to $\Delta\dot{\boldsymbol{q}}_a$) to

$$\Delta\vec{v}_C = \frac{1}{m}\left(\boldsymbol{H}_{0a} - \boldsymbol{H}_{0p}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\right)\Delta\dot{\boldsymbol{q}}_a = \frac{1}{m}\bar{\boldsymbol{H}}_{0a}\Delta\dot{\boldsymbol{q}}_a \tag{5.2}$$

which then also simplifies the velocity gain matrices from this:

$$\boldsymbol{G}_{va}(\boldsymbol{q}) = \frac{1}{m}\begin{bmatrix}\boldsymbol{H}_{xa} - \boldsymbol{H}_{xp}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\\ \boldsymbol{H}_{ya} - \boldsymbol{H}_{yp}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\end{bmatrix} \qquad \boldsymbol{G}_{\omega a}(\boldsymbol{q}) = \frac{\tilde{c}}{m|\vec{c}|^2}\left(\boldsymbol{H}_{0a} - \boldsymbol{H}_{0p}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}\right) \tag{5.3}$$

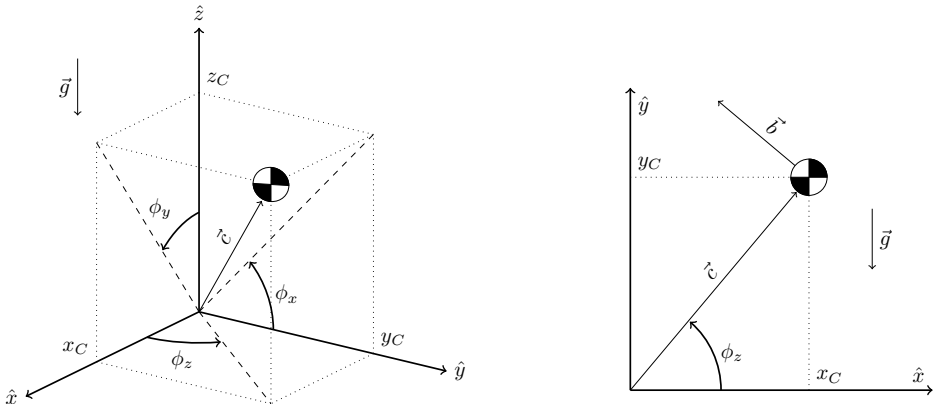
to this:

$$\boldsymbol{G}_{va}(\boldsymbol{q}) = \frac{1}{m}\begin{bmatrix}\bar{\boldsymbol{H}}_{xa}\\ \bar{\boldsymbol{H}}_{ya}\end{bmatrix} \qquad \boldsymbol{G}_{\omega a}(\boldsymbol{q}) = \frac{\tilde{c}}{m|\vec{c}|^2}\bar{\boldsymbol{H}}_{0a} \tag{5.4}$$

where $\tilde{c}$ again represents the 3x3 skew symmetric matrix which performs the cross product operation $\vec{c}\times\vec{d} = \tilde{c}\vec{d}$ for any 3D vector $\vec{d}$ (for details, see A.17).

Using the matrix determinant lemma, it can be shown that the matrix $\bar{\boldsymbol{H}}_{aa} = \boldsymbol{H}_{aa} - \boldsymbol{H}_{ap}\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa}$ is always invertible since $\boldsymbol{H}_{aa}^{-1}$ always exists ($\boldsymbol{H}_{aa}$ is positive definite as it is a principal submatrix of the positive definite matrix $\boldsymbol{H}$). This can be used to define the change in actuated joint velocities $\Delta\dot{\boldsymbol{q}}_a$ given an actuator impulse: $\Delta\dot{\boldsymbol{q}}_a = \bar{\boldsymbol{H}}_{aa}^{-1}\boldsymbol{\iota}_a$.

This simplified notation will be used throughout this chapter and the next to facilitate the definition of the generalized momentum gains introduced in the next section and the use of both velocity and momentum gains as objective criteria in the optimization framework outlined in Section 5.4 and applied in Chapter 6.



(a) 3D Coordinates (b) 2D Coordinates

Figure 5.1: Here, $\vec{c} = [x_C \; y_C \; z_C]^T$ is a vector from the contact point to the COM and the angles $\vec{\phi} = [\phi_x \; \phi_y \; \phi_z]^T$ are measured from the reference frame to $\vec{c}$. For 2D, gravity acts in the $-\hat{y}$ direction.

5.2 Momentum Gain

In this section, the momentum gains from Section 2.4.2 are expanded to general models in 2D and 3D, and the augmented inertia method is modified for use in calculating these general momentum gains. A novel method is also introduced for calculating both velocity and momentum gains using *spatial notation* (see [28, 29]) which depends on the direct relationship between a system's gains and its *centroidal momentum* [10, 33].

Recall the definitions of linear and angular momentum gain for planar, 2-link inverted pendulums as measures of the change in horizontal COM momentum and moment of momentum of the COM about the contact, respectively, due to an instantaneous change in joint torque $\Delta\boldsymbol{\tau}_a$, called an impulse and labeled $\boldsymbol{\iota}_a$ (see Section 2.4.2).

Building on this, the linear and angular momentum gains ($\vec{G}_m$ and $\vec{G}_o$, respectively) for a general system are (using the change in linear COM momentum $\Delta\vec{l} = m\Delta\vec{v}_C = \vec{\iota}_0$):

$$\vec{G}_m(\boldsymbol{\iota}_a) = \begin{bmatrix} \Delta l_x \\ \Delta l_y \end{bmatrix} \qquad \vec{G}_o(\boldsymbol{\iota}_a) = m|\vec{c}|^2\Delta\dot{\vec{\phi}} = \vec{c} \times \Delta\vec{l} \tag{5.5}$$

where $\vec{c} = [x_C \; y_C \; z_C]^T$ is a vector from the contact point to the COM with length $|\vec{c}| = \|\vec{c}\|_2$, the change in linear COM momentum is $\Delta\vec{l} = [\Delta l_x \; \Delta l_y \; \Delta l_z]^T = m\Delta\vec{v}_C$, the change in angular COM velocity about the contact point is $\Delta\dot{\vec{\phi}} = [\Delta\dot{\phi}_x \; \Delta\dot{\phi}_y \; \Delta\dot{\phi}_z]^T = (\vec{c} \times \Delta\vec{v}_C)/|\vec{c}|^2$, the total mass is m, and gravity acts in the $-\hat{z}$ direction (see Figure 5.1a).

In the planar case, the gains (G_m and G_o) and the change in angular COM velocity ($\Delta\dot{\phi}$) reduce to scalars, and the change in linear COM velocity ($\Delta\vec{v}_C$), linear COM momentum ($\Delta\vec{l}$), and COM vector ($\vec{c}$) reduce to 2D vectors. Defining $\hat{b} = [-y_C \; x_C]^T/|\vec{c}|$, a 2D unit vector perpendicular to $\vec{c}$ (see Figure 5.1b), the gains for general planar models are:

$$G_m(\boldsymbol{\iota}_a) = \Delta l_x \qquad G_o(\boldsymbol{\iota}_a) = m|\vec{c}|^2\Delta\dot{\phi}_z = |\vec{c}|(\Delta\vec{l}\cdot\hat{b}) \tag{5.6}$$

where $\vec{c} = [x_C \; y_C]^T$ is the vector from the contact point to the COM (with length $|\vec{c}| = \|\vec{c}\|_2$), the instantaneous change in linear COM momentum is $\Delta\vec{l} = [\Delta l_x \; \Delta l_y]^T$, the instantaneous change in angular COM velocity about the contact point is $\Delta\dot{\phi} = (\Delta\vec{l}\cdot\hat{b})/(m|\vec{c}|)$, and the impulse at the joints is $\boldsymbol{\iota}_a$ (which must satisfy $||\boldsymbol{\iota}_a|| = 1$).

Note that $\vec{G}_o$ and G_o are defined using the change in the angular momentum about the contact due only to COM motion. This is not the change in total angular momentum about the contact, which is always 0 for a passive rotary joint [27]: the instantaneous change in total angular momentum about a passive rotary joint cannot be changed by any impulsive change in a robot's joint angles.

The actuator impulse vector (i.e., the instantaneous change in torque $\boldsymbol{\iota}_a = \Delta\boldsymbol{\tau}_a$) is assumed to be a unit step impulse ($\|\boldsymbol{\iota}_a\| = 1$). After dividing the gains by the step impulse magnitude, this means that $\vec{G}_m$ has units of reciprocal length and $\vec{G}_o$ is dimensionless.

Comparing the different types of gains, there are two key differences between the momentum gain equations in (5.5) and the velocity gain equations in (2.11):

- First, the angular momentum gain equation does not include the division by $|\vec{c}\,|^2$ which is present in the angular velocity gain equation. This means that angular momentum gain is always finite, while the angular velocity gain approaches infinity as $|\vec{c}|$ approaches 0 and is undefined at $|\vec{c}| = 0$. Although this may not be an issue for some systems, for those that it affects it can be detrimental.

- Second, these equations now assume a unit step impulse instead of a unit velocity step, which inherently incorporates inertial information into the momentum gain. As shown in the next section, this inertial information enables the momentum gain to act as a measurement of how quickly the model can move its COM for unit motor impulses. In effect, a higher momentum gain implies less motor effort (i.e., less power) is required to achieve the same COM movement.

5.2.1 Augmented Inertia Method

Recall that the notation from Section 5.1 for augmented inertia matrices can be used to show that $\boldsymbol{\iota}_a = \bar{\boldsymbol{H}}_{aa}\Delta\dot{\boldsymbol{q}}_a$ and, via the matrix determinant lemma, that $\Delta\dot{\boldsymbol{q}}_a = \bar{\boldsymbol{H}}_{aa}^{-1}\boldsymbol{\iota}_a$. Therefore, the change in linear momentum $\Delta\vec{l}$ can be defined in terms of $\boldsymbol{\iota}_a$:

$$\Delta\vec{l} = \vec{\iota}_0 = \bar{\boldsymbol{H}}_{0a}\Delta\dot{\boldsymbol{q}}_a = \bar{\boldsymbol{H}}_{0a}\bar{\boldsymbol{H}}_{aa}^{-1}\boldsymbol{\iota}_a \tag{5.7}$$

This enables the definition of the momentum gain matrices $\boldsymbol{G}_{ma}(\boldsymbol{q})$ and $\boldsymbol{G}_{oa}(\boldsymbol{q})$ in terms of the velocity gain matrices, where $\vec{G}_m(\boldsymbol{\iota}_a) = \boldsymbol{G}_{ma}\boldsymbol{\iota}_a$ and $\vec{G}_o(\boldsymbol{\iota}_a) = \boldsymbol{G}_{oa}\boldsymbol{\iota}_a$, as:

$$\boldsymbol{G}_{ma} = \begin{bmatrix}\bar{\boldsymbol{H}}_{xa}\\ \bar{\boldsymbol{H}}_{ya}\end{bmatrix}\bar{\boldsymbol{H}}_{aa}^{-1} = m\boldsymbol{G}_{va}\bar{\boldsymbol{H}}_{aa}^{-1} \tag{5.8}$$

$$\boldsymbol{G}_{oa} = \tilde{c}\bar{\boldsymbol{H}}_{0a}\bar{\boldsymbol{H}}_{aa}^{-1} = m|\vec{c}|^2\boldsymbol{G}_{\omega a}\bar{\boldsymbol{H}}_{aa}^{-1} \tag{5.9}$$

The equation $\boldsymbol{\iota}_a = \bar{\boldsymbol{H}}_{aa}\Delta\dot{\boldsymbol{q}}_a$ can also be used to determine the impulse required at the actuators to achieve a given change in actuated joint velocity. This can be used to select which of a set of possible $\Delta\dot{\boldsymbol{q}}_a$ unit vectors would require the least energy to achieve, given the inertias of the links.

5.2.2 Spatial Method

Using spatial notation, the centroidal momentum of a system is defined as the aggregated angular and linear momenta of the system's links computed at the system's overall COM [33]. The linear component ($\vec{l}$) of the centroidal momentum ($\vec{\boldsymbol{h}}_C$) is the linear momentum of the system ($\vec{l} = m\vec{v}_C$), while the angular component ($\vec{k}_C$) is the total angular momentum the system has about its overall COM.

As shown in Equation (5.5), the momentum gains of a system are calculated using only its change in linear momentum, $\Delta\vec{l}$. This is due to the use of a passive rotational contact point, which causes $\Delta\vec{k}_0 = \vec{0} = [0\ 0\ 0]^T$, as discussed in Section 5.2.

Therefore, by using the contact point as the reference point, the change in angular momentum about the COM is: $\Delta\vec{k}_C = \Delta\vec{k}_0 - \vec{c}\times\Delta\vec{l} = -\vec{c}\times\Delta\vec{l}$. This relation is part of the spatial transformation matrix (${}^C\tilde{\boldsymbol{X}}_0^*$) from the contact to the COM, which maps the total system momentum at the contact ($\vec{\boldsymbol{h}}_0$) to the centroidal momentum: $\vec{\boldsymbol{h}}_C = {}^C\tilde{\boldsymbol{X}}_0^*\vec{\boldsymbol{h}}_0$.

Using ${}^C\tilde{\boldsymbol{X}}_0^*$ and $\Delta\vec{k}_0 = \vec{0}$ (due to passive contact), spatial gain is defined as:

$$\vec{\boldsymbol{G}}_h := \Delta\vec{\boldsymbol{h}}_C = {}^C\tilde{\boldsymbol{X}}_0^* \Delta\vec{\boldsymbol{h}}_0 = \begin{bmatrix} \tilde{1} & \tilde{c}^T \\ \tilde{0} & \tilde{1} \end{bmatrix} \begin{bmatrix} \vec{0} \\ \Delta\vec{l} \end{bmatrix} =: \begin{bmatrix} \vec{G}_k \\ \vec{G}_l \end{bmatrix} \tag{5.10}$$

where $\vec{G}_k = -\vec{c} \times \Delta\vec{l} = -m|\vec{c}|^2 \Delta\dot{\vec{\phi}}$ and $\vec{G}_l = \Delta\vec{l} = m\Delta\vec{v}_C$.

Assuming $\|\Delta\dot{\boldsymbol{q}}_a\| = 1$, these gains can be defined in terms of velocity gains:

$$\vec{G}_k(\Delta\dot{\boldsymbol{q}}_a) = -m|\vec{c}|^2 \vec{G}_\omega, \qquad \vec{G}_l(\Delta\dot{\boldsymbol{q}}_a) = m \begin{bmatrix} \vec{G}_v \\ G_g(\Delta\dot{\boldsymbol{q}}_a) \end{bmatrix}, \tag{5.11}$$

while assuming $||\boldsymbol{\iota}_a|| = 1$ gives their definitions in terms of momentum gains:

$$\vec{G}_k(\boldsymbol{\iota}_a) = -\vec{G}_o, \qquad \vec{G}_l(\boldsymbol{\iota}_a) = \begin{bmatrix} \vec{G}_m \\ G_g(\boldsymbol{\iota}_a) \end{bmatrix}. \tag{5.12}$$

Notice that $\vec{G}_l$ includes a vertical component of linear gain, labeled G_g, which could be used as a measure of hopping ability (as discussed in Section 5.4.2). This is similar to the component of $\vec{G}_\omega$ about a vertical axis, which could be used as a measure of spinning ability (as noted in [27]).

These equations can also be used to define gain matrices for the new gains $\vec{G}_l$ and $\vec{G}_k$, similar to those defined for the velocity and momentum gains:

$$\begin{aligned} \boldsymbol{G}_{la}(\Delta\dot{\boldsymbol{q}}_a) &:= \bar{\boldsymbol{H}}_{0a}, & \boldsymbol{G}_{la}(\boldsymbol{\iota}_a) &:= \bar{\boldsymbol{H}}_{0a}\bar{\boldsymbol{H}}_{aa}^{-1} = \boldsymbol{G}_{la}(\Delta\dot{\boldsymbol{q}}_a)\bar{\boldsymbol{H}}_{aa}^{-1}, \\ \boldsymbol{G}_{ka}(\Delta\dot{\boldsymbol{q}}_a) &:= \tilde{c}^T \bar{\boldsymbol{H}}_{0a}, & \boldsymbol{G}_{ka}(\boldsymbol{\iota}_a) &:= \tilde{c}^T \bar{\boldsymbol{H}}_{0a}\bar{\boldsymbol{H}}_{aa}^{-1} = \boldsymbol{G}_{ka}(\Delta\dot{\boldsymbol{q}}_a)\bar{\boldsymbol{H}}_{aa}^{-1}. \end{aligned} \tag{5.13}$$

Combining these equations, the spatial gain matrices $\boldsymbol{G}_{ha}(\Delta\dot{\boldsymbol{q}}_a)$ and $\boldsymbol{G}_{ha}(\boldsymbol{\iota}_a)$ can be defined in terms of $\bar{\boldsymbol{H}}_{0a}$ and $\bar{\boldsymbol{H}}_{aa}^{-1}$:

$$\begin{aligned} \vec{\boldsymbol{G}}_h(\Delta\dot{\boldsymbol{q}}_a) &= \begin{bmatrix} \tilde{c}^T \\ \tilde{1} \end{bmatrix} \bar{\boldsymbol{H}}_{0a}\Delta\dot{\boldsymbol{q}}_a = \boldsymbol{G}_{ha}(\Delta\dot{\boldsymbol{q}}_a)\Delta\dot{\boldsymbol{q}}_a, \\ \vec{\boldsymbol{G}}_h(\boldsymbol{\iota}_a) &= \begin{bmatrix} \tilde{c}^T \\ \tilde{1} \end{bmatrix} \bar{\boldsymbol{H}}_{0a}\bar{\boldsymbol{H}}_{aa}^{-1}\boldsymbol{\iota}_a = \boldsymbol{G}_{ha}(\boldsymbol{\iota}_a)\boldsymbol{\iota}_a = \boldsymbol{G}_{ha}(\Delta\dot{\boldsymbol{q}}_a)\bar{\boldsymbol{H}}_{aa}^{-1}\boldsymbol{\iota}_a. \end{aligned} \tag{5.14}$$

5.2.3 Centroidal Momentum Comparison

The centroidal momentum matrix ${}^{C}\boldsymbol{A}$ relates the centroidal momentum of a system to its joint velocities: $\vec{\boldsymbol{h}}_C = {}^{C}\boldsymbol{A}\dot{\boldsymbol{q}}$ [10, 33]. As discussed in Section 2.1 and earlier in this chapter, the centroidal momentum of a system is a common tool for measuring and controlling the overall COM dynamics of a balancing system.

By dividing ${}^{C}\boldsymbol{A}$ into passive and actuated components (${}^{C}\boldsymbol{A} = [{}^{C}\boldsymbol{A}_p \; {}^{C}\boldsymbol{A}_a]$), an actuated centroidal momentum matrix ${}^{C}\bar{\boldsymbol{A}}_a$ can be defined:

$$\begin{aligned} \Delta\vec{\boldsymbol{h}}_C &= {}^{C}\boldsymbol{A}_p\Delta\dot{\boldsymbol{q}}_p + {}^{C}\boldsymbol{A}_a\Delta\dot{\boldsymbol{q}}_a \\ &= ({}^{C}\boldsymbol{A}_a - {}^{C}\boldsymbol{A}_p\boldsymbol{H}_{pp}^{-1}\boldsymbol{H}_{pa})\Delta\dot{\boldsymbol{q}}_a \\ &= {}^{C}\bar{\boldsymbol{A}}_a\Delta\dot{\boldsymbol{q}}_a \end{aligned} \tag{5.15}$$

Therefore, since the spatial gain has been defined as $\vec{\boldsymbol{G}}_h = \Delta\vec{\boldsymbol{h}}_C$, it can be shown that ${}^{C}\bar{\boldsymbol{A}}_a$ is equivalent to the velocity-based spatial gain matrix (${}^{C}\bar{\boldsymbol{A}}_a = \boldsymbol{G}_{ha}(\Delta\dot{\boldsymbol{q}}_a)$):

$$\boldsymbol{G}_{ha}(\Delta\dot{\boldsymbol{q}}_a)\Delta\dot{\boldsymbol{q}}_a = \vec{\boldsymbol{G}}_h = \Delta\vec{\boldsymbol{h}}_C = {}^{C}\boldsymbol{A}\Delta\dot{\boldsymbol{q}} = {}^{C}\bar{\boldsymbol{A}}_a\Delta\dot{\boldsymbol{q}}_a \tag{5.16}$$

These definitions provide another alternative method of calculating both the velocity and momentum gains of a system using the well known centroidal momentum equations. They also provide an intuitive interpretation for control algorithms based on centroidal momentum [10, 73, 74]. Using velocity and momentum gains as a metric against which to evaluate these controllers could also be used as a controller-agnostic tool to analyze and compare the balancing performance of different mechanisms and controllers.

Velocity and momentum gains quantify the fundamental physical limits of a system's COM motion, and by extension its ability to balance. As demonstrated in [101–103], using the velocity gain directly in the system model enables excellent balance performance with even a simple PID controller. In fact, it could be argued that any existing balancing controller which is capable of reliably returning the COM to a balanced state is actually using some form of these gains already, typically by regulating angular momentum.

Based on the inherent relationship between the centroidal momentum and these gains, which led to the definition of the spatial gain, more complex controllers which use centroidal momentum to balance should also demonstrate excellent balance performance. While using the gains (either directly or indirectly) as part of a control strategy can lead to balance performance near the peak of what a given system is capable of, they can also be used to optimize a system's physical properties to maximize its peak balance performance.

5.3 Generalized Gain

Velocity and momentum gains quantify how fast an articulating system balancing on a passive (possibly rolling) contact can move its COM, and by extension how well it will be able to balance or walk. In addition to their independence from the controller and/or trajectory used, these gains are invariant to a scaling of the total mass of the system, and the angular gains are also invariant to a scaling of total length, allowing the balancing capabilities of an entire class of mechanisms to be quantified with a single metric [27].

Therefore, a generalized gain equation is desired which can be used to calculate any of these gains. Building on the spatial gain formulation, the generalized gain $\boldsymbol{G}(\boldsymbol{u}_a) = \boldsymbol{G}_a\boldsymbol{u}_a$ is defined for a given system (with $\boldsymbol{G}_a$ and $\boldsymbol{u}_a$ the generalized gain matrix and input vector, respectively) as

$$\boldsymbol{G} = \boldsymbol{S}_G\vec{\boldsymbol{G}}_h(\Delta\dot{\boldsymbol{q}}_a) \qquad \text{s.t. } \|\boldsymbol{u}_a\| = \|\boldsymbol{S}_a\Delta\dot{\boldsymbol{q}}_a\| = 1 \tag{5.17}$$

where $\boldsymbol{S}_G$ is an $n_G \times 6$ gain selection and scaling matrix and $\boldsymbol{S}_a$ is an $n_a \times n_a$ actuator scaling matrix (with n_G the desired gain vector length and n_a the number of actuated joints). This generalized gain can be used to formulate all of the velocity and momentum gains described in 2.4 and 5.2.

Using (5.14), the generalized gain matrix is defined as

$$\boldsymbol{G}_a = \boldsymbol{S}_G\boldsymbol{G}_{ha}(\Delta\dot{\boldsymbol{q}}_a)\boldsymbol{S}_a^{-1} \tag{5.18}$$

In general, the matrix $\boldsymbol{S}_G$ can be used to both scale each of the computed gains and select which gains are of interest. When computing velocity gains, the generalized input vector $\boldsymbol{u}_a$ is equal to $\Delta\dot{\boldsymbol{q}}_a$, so the actuator scaling matrix is set to $\boldsymbol{S}_a = \boldsymbol{1}_{n_a\times n_a}$. For momentum gains, the actuator scaling matrix is set to $\boldsymbol{S}_a = \bar{\boldsymbol{H}}_{aa}$ based on the definition $\boldsymbol{\iota}_a = \bar{\boldsymbol{H}}_{aa}\Delta\dot{\boldsymbol{q}}_a$ from Subsection 5.1. As shown in (5.18), this leads to the familiar post-multiplication of velocity gain matrices by $\bar{\boldsymbol{H}}_{aa}^{-1}$ to become momentum gain matrices.

The scaling factors and selection matrices required to formulate each of the gains from Sections 2.4 and 5.2 are shown in Table 5.1. It also shows the key differences between velocity and momentum gains:

- The velocity gains include a division by m, the total mass, and (for the angular portion) by the square of $|\vec{c}|$, the distance between the contact and the COM.
- The momentum gains include $\bar{\boldsymbol{H}}_{aa}$, which defines the relationship between the change in joint velocities $\Delta\dot{\boldsymbol{q}}_a$ and the impulse $\boldsymbol{\iota}_a$ that causes that change.

Table 5.1: Generalized Gain Formulations for 2D, 3D, and Spatial Gains

For these gains, the matrix $\boldsymbol{S}_G$ can be defined as the product of a scaling factor k_G and a selection matrix $\boldsymbol{I}_G$: $\boldsymbol{S}_G = k_G \boldsymbol{I}_G$. To concisely define $\boldsymbol{I}_G$, $\hat{\boldsymbol{e}}_i$ is defined as the ith column of the 6×6 identity matrix $\tilde{\mathbf{1}}$. With a minor abuse of notation, the $n_a \times n_a$ identity matrix is written as $\mathbf{1}_{a\times a}$. For these gains, the generalized gain equation can be rewritten as:

$$\boldsymbol{G} = k_G \boldsymbol{I}_G \begin{bmatrix} \tilde{c}^T \\ \tilde{1} \end{bmatrix} \bar{\boldsymbol{H}}_{0a} \boldsymbol{S}_a^{-1} \boldsymbol{u}_a \quad \text{s.t. } \|\boldsymbol{u}_a\| = \|\boldsymbol{S}_a \Delta\dot{\boldsymbol{q}}_a\| = 1$$

2D, 3D, and Spatial Gains	k_G	$\boldsymbol{I}_G$	$\boldsymbol{S}_a$
Spatial Velocity Gain: $\vec{\boldsymbol{G}}_h(\Delta\dot{\boldsymbol{q}}_a)$	1	$\tilde{\mathbf{1}}$	$\mathbf{1}_{a\times a}$
3D Angular Velocity Gain: $\vec{G}_\omega(\Delta\dot{\boldsymbol{q}}_a)$	$\frac{-1}{m\lvert\vec{c}\rvert^2}$	$\begin{bmatrix}\hat{\boldsymbol{e}}_1 & \hat{\boldsymbol{e}}_2 & \hat{\boldsymbol{e}}_3\end{bmatrix}^T$	
2D Angular Velocity Gain: $G_\omega(\Delta\dot{\boldsymbol{q}}_a)$		$\hat{\boldsymbol{e}}_3$	
3D Linear Velocity Gain: $\vec{G}_v(\Delta\dot{\boldsymbol{q}}_a)$	$\frac{1}{m}$	$\begin{bmatrix}\hat{\boldsymbol{e}}_4 & \hat{\boldsymbol{e}}_5\end{bmatrix}^T$	
2D Linear Velocity Gain: $G_v(\Delta\dot{\boldsymbol{q}}_a)$		$\hat{\boldsymbol{e}}_4$	
Spatial Momentum Gain: $\vec{G}_h(\boldsymbol{\iota}_a)$	1	$\tilde{\mathbf{1}}$	$\bar{\boldsymbol{H}}_{aa}$
3D Angular Momentum Gain: $\vec{G}_o(\boldsymbol{\iota}_a)$	-1	$\begin{bmatrix}\hat{\boldsymbol{e}}_1 & \hat{\boldsymbol{e}}_2 & \hat{\boldsymbol{e}}_3\end{bmatrix}^T$	
2D Angular Momentum Gain: $G_o(\boldsymbol{\iota}_a)$		$\hat{\boldsymbol{e}}_3$	
3D Linear Momentum Gain: $\vec{G}_m(\boldsymbol{\iota}_a)$	1	$\begin{bmatrix}\hat{\boldsymbol{e}}_4 & \hat{\boldsymbol{e}}_5\end{bmatrix}^T$	
2D Linear Momentum Gain: $G_m(\boldsymbol{\iota}_a)$		$\hat{\boldsymbol{e}}_4$	

Based on these differences, Featherstone's conclusion that there was no objective reason to use momentum gains after defining them for the 2-link planar system in [27] are justified: The mechanism in [27] has only one actuated joint and fixed link lengths and inertias, which would cause the scaling matrices to all simplify to constant positive scalar values.

However, it should be evident based on Table 5.1 that in any other case (such as a planar system with more joints or a 3D system) the momentum gains provide additional useful and different information about the system compared to the velocity gains. This is primarily due to the consideration of inertial loading on the actuators via k_G and $\boldsymbol{S}_a$.

5.4 Optimization Framework

A general optimization-based framework has been developed for designing parameterized mechanisms without the need for designing/defining a controller or trajectory. By relying only on the gains, the optimization is formulated solely using the physical properties of the system within a desired motion space. This enables the framework to determine where a mechanism's fundamental limits exist for the desired behavior, independent of the controller used to achieve the behavior (typically gait or balance[2]) or the desired trajectory.

Specifically, the parameters of the model are found which maximize the *potential* of the given mechanism to balance and locomote, independent of the controller used to achieve those goals. This gives an upper limit on how well the mechanism can balance and walk using any controller in a range of configurations near the desired motion subspace, avoiding overfitting to a specific controller and/or trajectory.

This framework requires five main elements:

- A *model* of the parameterized mechanism (the number of links, joint details, etc.);
- The modifiable parameters $\boldsymbol{x}$ of the model (e.g., link mass, length, COM), with their allowable upper and lower bounds ($\boldsymbol{x}_{min}$ and $\boldsymbol{x}_{max}$, can be $\pm\infty$ if desired);
- A parameter map $\boldsymbol{x}$, which dictates how to assign the parameters $\boldsymbol{x}$ to the model;
- A set of key poses $\boldsymbol{Q} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \dots\}$ in the model's configuration space (including the passive joint); and
- An objective function $J(\boldsymbol{x})$ which quantifies a model's ability to achieve a desired behavior for a given $\boldsymbol{x}$.

[2]In this framework, it's assumed that the angular momentum about the COM is negligible or regulated, so balance is only concerned with the horizontal COM motion.

Once these five elements have been selected, a global optimization is used to find the optimal parameterization within the limits set on the parameters:

$$\max_{\boldsymbol{x}} J(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max} \tag{5.19}$$

The model must define the overall morphology of the desired mechanism. This will typically include the relative positions and orientations of the joints, motion freedoms of the joints, and the links' inertial properties. Any aspect of the mechanism which cannot be changed or modified as part of the optimization is included in the model definition.

Aspects of the mechanism to be optimized are then defined as the modifiable parameters $\boldsymbol{x}$ of the system. This will typically consist of link properties (e.g., mass, length, inertia), but can also include joint directions, orientations, or any other desired model property. A set of upper and lower bounds on the parameters, labeled $\boldsymbol{x}_{min}$ and $\boldsymbol{x}_{max}$ must also be defined. For ease of use, these limits can be set equal to each other to fix a parameter at a particular value or can be set to $\pm\infty$, as appropriate, to allow unbounded exploration.

Once the model and parameters are defined, a mapping $\boldsymbol{x}$ is required which assigns a given set of parameters onto the model. This enables the use of an existing modeling platform to be used in the objective function calculations. As an example, in 2D the scalar inertia of each link about its COM can be parameterized using a ratio between the link's radius of gyration and its length. As part of the mapping, these ratios would be converted to an equivalent inertia, using the (possibly also parameterized) length and mass of the link, and applied to the dynamic model.

The key poses $\boldsymbol{q}_i$ in $\boldsymbol{Q}$ can be chosen to approximate the entire configuration space of a given model, or can be used more selectively to focus the optimization on a particular subset of the space. Whether some or all of the joints are passive or active, the full configuration of the model must be specified for each key pose in the set $\boldsymbol{Q}$.

Finally, the objective function $J(\boldsymbol{x})$ generates a scalar value which quantifies the ability of a particular parameterization of the model to achieve a desired behavior. In general, $J(\boldsymbol{x})$ uses the parameter map $\boldsymbol{x}$ to assign the parameters $\boldsymbol{x}$ to the given model, and then computes an objective value based on the model's dynamics for the set of key poses $\boldsymbol{Q}$.

5.4.1 Objective Functions

The global objective function used in this work is the mean of the pose-specific objective function $J(\boldsymbol{q}, \boldsymbol{x})$ across all n_q key poses:

$$J(\boldsymbol{x}) = \frac{1}{n_q} \sum_{i=1}^{n_q} J(\boldsymbol{q}_i, \boldsymbol{x}), \qquad \forall \boldsymbol{q}_i \in \boldsymbol{Q}. \tag{5.20}$$

Pose-specific objective functions are defined for each of the velocity- and momentum-based gains, quantifying the model's balancing ability for each $(\boldsymbol{q}, \boldsymbol{x})$ pair as the maximum magnitude each gain could achieve for the given pair (where $J_* = J_*(\boldsymbol{q}, \boldsymbol{x})$):

$$\begin{aligned} J_* &= \max_{\Delta\dot{\boldsymbol{q}}_a} |\boldsymbol{G}_*(\Delta\dot{\boldsymbol{q}}_a)| = \|\boldsymbol{G}_{*a}\| \quad \text{s.t. } \|\Delta\dot{\boldsymbol{q}}_a\| = 1, \\ J_* &= \max_{\boldsymbol{\iota}_a} |\boldsymbol{G}_*(\boldsymbol{\iota}_a)| = \|\boldsymbol{G}_{*a}\| \qquad \text{s.t. } \|\boldsymbol{\iota}_a\| = 1. \end{aligned} \tag{5.21}$$

Note that although $\boldsymbol{q}$ and $\boldsymbol{x}$ are fixed, the $\Delta\dot{\boldsymbol{q}}_a$ or $\boldsymbol{\iota}_a$ unit vector which maximizes the velocity or momentum gain, respectively, must be determined for each pair of $\boldsymbol{q}$ and $\boldsymbol{x}$.

By maximizing the mean of these objectives across all key poses as the overall objective and using the magnitude of the gains, any parameter set with zero crossings (or gains near zero) in the configuration set $\boldsymbol{Q}$ will be avoided.

The selected key poses in $\boldsymbol{Q}$ may cover a model's entire (active and passive) configuration space, if it is expected to regularly operate throughout the entire space, or can be used selectively to focus the optimization on a particular subset of the space (e.g., a biped standing on one or both feet).

These objective formulations enable the design of parameterized mechanisms without a controller or trajectory, as the gains are a function only of the physical properties of the system in a given configuration. In fact, the gains are independent of the controller used, invariant to gravitational and velocity product dynamics, and (when balancing on a point or line contact) independent of the contact angle. Therefore, the framework is able to determine and improve a system's fundamental physical limits for a desired behavior (typically gait or balance[3]) independent of the controller.

[3]Here, it's assumed that the angular momentum about the COM is negligible or regulated, so balance is only concerned with the COM motion.

Planar Objective Functions

A typical goal for this framework would be to find a unit velocity or impulse vector which maximizes a given gain for one configuration $\boldsymbol{q}$ of the model. The gain vectors are a useful tool for finding the $\Delta\dot{\boldsymbol{q}}_a$ or $\boldsymbol{\iota}_a$ vector which achieves this maximization.

For the purposes of balancing in the plane, whether the system is 2D or can be mapped from 3D into a given plane, the pose-specific objective functions can be simplified as follows:

- For velocity-based gains, the objective function is the norm of the gain vector, either linear or angular. Since velocity gains require $||\Delta\dot{\boldsymbol{q}}_a|| = 1$, these vector norms maximize the selected gain for the given $(\boldsymbol{q}, \boldsymbol{x})$ pair.

$$J_*(\boldsymbol{q}, \boldsymbol{x}) = \max_{\Delta\dot{\boldsymbol{q}}_a} |G_*(\Delta\dot{\boldsymbol{q}}_a)| = ||\boldsymbol{G}_{*a}(\boldsymbol{q}, \boldsymbol{x})||. \tag{5.22}$$

- For momentum-based gains, the gain vectors are replaced with momentum gain vectors and the assumption is $||\boldsymbol{\iota}_a|| = 1$. As above, the objective function is the norm of the gain vector, which maximizes the selected gain for the given $(\boldsymbol{q}, \boldsymbol{x})$ pair.

$$J_*(\boldsymbol{q}, \boldsymbol{x}) = \max_{\boldsymbol{\iota}_a} |G_*(\boldsymbol{\iota}_a)| = ||\boldsymbol{G}_{*a}(\boldsymbol{q}, \boldsymbol{x})||. \tag{5.23}$$

Maximizing Gains via Norms

In the planar case, the pose-specific objective functions make use of vector norms to simplify their calculations. The vector norm to use will depend on the norm being used to determine the magnitude of the unit vector in question. In general, assuming a unit vector $\boldsymbol{u}_a$, gain $G(\boldsymbol{u}_a)$, and gain vector $\boldsymbol{G}_a$, the following definition holds:

$$\begin{aligned} J(\boldsymbol{q}, \boldsymbol{x}) &= \max_{\boldsymbol{u}_a} |G(\boldsymbol{u}_a)| && \text{s.t. } ||\boldsymbol{u}_a|| = 1 \\ &= \max_{\boldsymbol{u}_a} |\boldsymbol{G}_a \boldsymbol{u}_a| && \text{s.t. } ||\boldsymbol{u}_a|| = 1 \\ &= \max_{\boldsymbol{u}_a} |\textstyle\sum_{i \in a} G_i u_i| && \text{s.t. } ||\boldsymbol{u}_a|| = 1 \end{aligned} \tag{5.24}$$

Since $\boldsymbol{u}_a$ can be freely chosen, provided it satisfies $||\boldsymbol{u}_a|| = 1$, the magnitude brackets can be dropped by matching the signs of G_i and u_i (i.e., $\forall i \in a, G_i u_i \geq 0$):

$$J(\boldsymbol{q}, \boldsymbol{x}) = \max_{\boldsymbol{u}_a} \textstyle\sum_{i \in a} G_i u_i \quad \text{s.t.} ||\boldsymbol{u}_a|| = 1 \tag{5.25}$$

If the ∞-norm is used to determine the magnitude of the unit vector ($||\boldsymbol{u}_a||_\infty = \max_{i \in a} |u_i| = 1$), the maximum magnitude of the gain will occur when $\forall i \in a, |u_i| = 1$. In other words, the maximum gain will occur when every entry in the unit vector has a magnitude of 1 (where the sign doesn't matter due to the norm formulation).

Similarly, if the 1-norm is used to determine the magnitude of the unit vector ($||\boldsymbol{u}_a||_1 = \sum_{i \in a} |u_i| = 1$), the maximum magnitude of the gain will occur when $|u_j| = 1$ and $\forall k \in a \neq j, u_k = 0$ (where j is the index of the largest magnitude element in $\boldsymbol{G}_a$). In other words, the maximum gain will occur when the only non-zero entry in the unit vector is at the same index as the maximum magnitude entry in the gain vector.

This leads to the following simplifications:

$$\begin{aligned} J(\boldsymbol{q}, \boldsymbol{x}) &= ||\boldsymbol{G}_a||_\infty \qquad \text{if } ||\boldsymbol{u}_a||_1 = 1 \\ J(\boldsymbol{q}, \boldsymbol{x}) &= ||\boldsymbol{G}_a||_1 \qquad \text{if } ||\boldsymbol{u}_a||_\infty = 1 \end{aligned} \tag{5.26}$$

Finally, if the 2-norm is used to determine the magnitude of the unit vector ($||\boldsymbol{u}_a||_2 = \sqrt{\sum_{i \in a} u_i^2} = 1$), the following simplification can be used based on the definition of the inner product as $\vec{a} \cdot \vec{b} = ||\vec{a}||_2 \; ||\vec{b}||_2 \cos\theta$ (where θ is the angle between the vectors):

$$\begin{aligned} J(\boldsymbol{q}, \boldsymbol{x}) &= \max_{\boldsymbol{u}_a} |\boldsymbol{G}_a \boldsymbol{u}_a| \qquad && \text{s.t. } ||\boldsymbol{u}_a||_2 = 1 \\ &= \max_{\boldsymbol{u}_a} |\boldsymbol{G}_a^T \cdot \boldsymbol{u}_a| \qquad && \text{s.t. } ||\boldsymbol{u}_a||_2 = 1 \\ &= \max_{\boldsymbol{u}_a} ||\boldsymbol{G}_a^T||_2 \; |\cos\theta| \qquad && \text{s.t. } ||\boldsymbol{u}_a||_2 = 1 \end{aligned} \tag{5.27}$$

Since $\boldsymbol{G}_a$ is a function only of $\boldsymbol{q}$ and $\boldsymbol{x}$, and $\boldsymbol{u}_a$ can be freely chosen provided $||\boldsymbol{u}_a||_2 = 1$, $|\cos\theta| = 1$ (its maximum value) can be guaranteed by setting $\boldsymbol{u}_a$ parallel to $\boldsymbol{G}_a$. As their signs are equal, setting $\boldsymbol{u}_a = \boldsymbol{G}_a^T / ||\boldsymbol{G}_a^T||_2$ will maximize the gain when $||\boldsymbol{u}_a||_2 = 1$. Given that norms are independent of the orientation (row vs column) of the normed vector (i.e., $||\boldsymbol{G}_a|| = ||\boldsymbol{G}_a^T||$), this leads to the final simplification that is used in this work:

$$J(\boldsymbol{q}, \boldsymbol{x}) = ||\boldsymbol{G}_a||_2 \qquad \text{if } ||\boldsymbol{u}_a||_2 = 1 \tag{5.28}$$

Therefore, the type of vector norm that must be used on the gain vector in order to maximize the gain in these objective functions depends directly on the type of norm that is used to define the unit vectors $||\Delta\dot{\boldsymbol{q}}|| = 1$ and $||\boldsymbol{\iota}_a|| = 1$:

- If the unit vector uses a 1-norm, the objective function must use the ∞-norm.
- If the unit vector uses an ∞-norm, the objective function must use the 1-norm.
- If the unit vector uses a 2-norm, the objective function must also use the 2-norm.

Generalized Objective Function

An objective function is now defined to optimize mechanism design using the generalized gain equations from above, building on the planar approach proposed above. This generalized objective function is then augmented with weighting matrices to enable the application of domain knowledge to guide the optimization.

As before, the purpose of this novel objective function is to guide the automated optimal design of a mechanism by quantifying its ability to move its COM throughout a given subset of the mechanism's configuration space, without defining or generating an associated controller or trajectory.

To automatically generate an optimal mechanism, one or more of the gains defined in Section 5.3 are maximized. When working in 3D, the gains are no longer scalars but vectors. Similarly, if optimization over multiple gains is desired (in 2D or 3D), the overall gain is also a vector. In light of this, a method for choosing what value to optimize as the scalar representation of the overall gain is required.

Therefore, the general pose-specific objective function used in this work (under the assumption that $\|\boldsymbol{u}_a\| = 1$) is defined as

$$J(\boldsymbol{q}, \boldsymbol{x}) = \max_{\boldsymbol{u}_a} \|\boldsymbol{G}(\boldsymbol{u}_a)\| = \max_{\boldsymbol{u}_a} \|\boldsymbol{G}_a \boldsymbol{u}_a\| = \|\boldsymbol{G}_a\| \tag{5.29}$$

where $\|\boldsymbol{G}_a\|$ represents the induced matrix norm of $\boldsymbol{G}_a$.

Choosing which gain(s) and norm(s) to use in the objective function to achieve the desired outcome is a critical step in successfully applying the optimization framework. Here, three different types of high-level goals are considered:

- Maximizing one particular direction of the selected gain for each configuration, with the specific direction either chosen *a priori* (e.g., sagittal motion) or automatically determined using an ∞-norm on the gain matrix.
- Maximizing the selected gain based on the use of only a single joint in each configuration, with the specific joint either chosen *a priori* (e.g., ankle pitch) or automatically determined using a 1-norm on the gain matrix.
- Maximizing the largest possible gain in any included direction for each configuration, with the direction determined using a 2-norm on the gain matrix.

Using the first type of goal with a direction chosen *a priori* effectively results in projecting the 3D system into 2D, where the gain is a scalar and the planar formulation of

the objective function from Equations (5.22) and (5.23) can be used. As an example, the scaling and selection matrices can be set as defined in Table 5.1 to duplicate the objective functions defined in (5.21) and apply 2-norms to reproduce the planar optimization functions.

For the second goal, which focuses on a specific joint, the formulation of the generalized gain will result in a scalar quantification of how well the system can balance using only that joint. For a joint chosen *a priori*, this scalar gain is a metric for how well that specific joint can balance the system with the given parameters. For automatically chosen joints, it can also be used to define which joint(s) to use for balance in each region of the robot's configuration space.

For the final goal, which attempts to maximize the overall gain for each configuration, the result is comparable to the ellipses and ellipsoids generated by the (dynamic) COM manipulability metrics defined in [21–23].

Although the results of these metrics (both generalized gain and dynamic COM manipulability) can be applied directly to the optimization of a mechanism, their real flexibility and power are revealed when combined with weighting matrices to incorporate domain knowledge into the optimization. For example, combinations of weights can be used to account for scaling issues, differences in units, joint limits (on position, torque, etc.), and even the relative importance of the gain directions and/or the inputs.

To augment the objective function with this kind of expert domain knowledge, weights can be applied to the gain matrix and input vector by using weighted matrix norms in place of the standard induced norms. To this end, the weighted general pose-specific objective function (with $J_W = J_W(\vec{q}, \vec{x})$ and symmetric positive definite weights) are defined as

$$\begin{aligned} J_W &= \max_{\boldsymbol{u}_a} \|\boldsymbol{G}_a \boldsymbol{u}_a\|_{\boldsymbol{W}_G} && \text{s.t. } \|\boldsymbol{u}_a\|_{\boldsymbol{W}_a} = 1 \\ &= \max_{\boldsymbol{u}_a} \|\boldsymbol{W}_G \boldsymbol{G}_a \boldsymbol{u}_a\| && \text{s.t. } \|\boldsymbol{W}_a \boldsymbol{u}_a\| = 1 \\ &= \max_{\boldsymbol{v}_a} \|\boldsymbol{W}_G \boldsymbol{G}_a \boldsymbol{W}_a^{-1} \boldsymbol{v}_a\| && \text{s.t. } \|\boldsymbol{v}_a\| = 1 \\ &= \|\boldsymbol{W}_G \boldsymbol{G}_a \boldsymbol{W}_a^{-1}\| \\ &= \|\boldsymbol{G}_a\|_{\boldsymbol{W}_G, \boldsymbol{W}_a} \end{aligned} \tag{5.30}$$

where $\boldsymbol{W}_G$ and $\boldsymbol{W}_a$ are the gain and actuator weights, respectively, and the weighted vector norm $\|\cdot\|_{\boldsymbol{A}} = \|\boldsymbol{A} \cdot\|$ induces the weighted matrix norm $\|\cdot\|_{\boldsymbol{A},\boldsymbol{B}} = \|\boldsymbol{A} \cdot \boldsymbol{B}^{-1}\|$.

5.4.2 Discussion

The main existing application of velocity and momentum gains are the planar momentum based controllers developed and used in [101–103]. The plant model gains which were defined in [103] can also be defined using $\bar{\boldsymbol{H}}_{\beta\alpha}$ notation as:

$$Y_1 = -\frac{1}{\bar{H}_{x2}}\frac{H_{x1}}{H_{11}} \qquad Y_2 = -\frac{1}{\bar{H}_{x2}}g \qquad \boldsymbol{Y}_3 = \frac{1}{\bar{H}_{x2}}\bar{\boldsymbol{H}}_{x3} \tag{5.31}$$

Comparing the common element $\bar{H}_{x2} = mG_v$ to Equation (5.13), the model defined in [103] could equivalently use the horizontal component of $\vec{G}_l(\Delta\dot{\boldsymbol{q}}_a)$.

Similarly, a majority of recent work on whole body balance uses centroidal momentum (e.g., [10, 73, 74]), which also use controllers which could be defined using spatial gain. An examination of the relationship between balancing a system using centroidal momentum and the evolution of its spatial gain is warranted.

As mentioned in [27], the angular gains in 3D include a vertical component, which does not contribute to balance but to spinning around the contact point. The linear analog are the vertical gains defined in this work as G_g, which also do not contribute to balance. However, they could be used as a method of quantifying the capability of a system to move vertically, with applications in hopping or bouncing.

The momentum gains defined in this chapter can be easily extended to the case of rolling contact, knife-edge contact, or a compliant base of support in 2D or 3D, similar to how the velocity gains have been extended in [27]. However, assuming the system has only a single contact with the environment (or mimics one via compliance) is the main limitation of both velocity and momentum gains.

This single contact limitation can be partially overcome by assuming a set of one or more co-located joints are passive (e.g., an ankle or hip). This assumed-passive set of joints can then be used in place of the contact point, allowing the momentum and velocity gains to be applied. To determine the physical capabilities of a system with multiple contacts where a set of co-located joints cannot be assumed passive, a more complex metric is needed such as dynamic COM manipulability [23].

Alternatively, the passive contact point can be assumed to be located in the common rotation point of a set of co-located actuated joints of the system (e.g., the ankle or hip). If this is done, the actuated joints can be used to simulate a rolling contact by artificially modifying and controlling the apparent radius of gyration of the system's contact surface, thereby changing the balance and gain properties directly.[4]

[4]This observation is from a discussion with Roy Featherstone at IROS 2018 about actuated ankles.

5.5 Summary

In this chapter, Featherstone's initial definition of momentum gain for planar 2-link inverted pendulums have been extended to general 2D and 3D mechanisms, along with defining notation to simplify both the velocity and momentum gain equations. Two different methods were described for calculating these general momentum gains, namely the Augmented Inertia method and the Spatial method.

The Spatial method development also provided insight on the relationship between a system's centroidal momentum and its velocity and momentum gains. This enabled the definition of the spatial gain of a system, incorporating both angular and linear components in all 6 directions of 3D motion, and to define an actuated centroidal momentum matrix which was shown to be equivalent to the spatial gain matrix.

Building on this work, a generalized gain formulation was developed for walking and balancing mechanisms that quantifies their COM motion properties and encapsulates and clarifies the relationship between velocity and momentum gains. This novel gain formulation provides an easily defined metric for analyzing the COM motion capabilities of any mechanism using passive contact to interact with its environment, which enables simple quantitative comparisons between different mechanisms across the design space.

A general optimization framework was also introduced for the design of parameterized mechanisms using these generalized gains in combination with weighted matrix norms as optimization criteria. Since the gains are all invariant to a scaling of total mass and the angular gains are invariant to a scaling of total length, entire families of mechanisms can be optimized in one application of the framework. The gains' invariance to mass and length scaling also means that a smaller, cheaper prototype can be built and tested to evaluate the performance of a full size system.

This framework provides an additional tool for mechanism designers to automatically explore the design space of their given mechanism, without having to generate simulations, trajectories, or controllers. An existing design with fixed kinematic and inertial parameters can also benefit from the application of a gain-based optimization. For example, using matrix 1-norms to evaluate a system's gains over a given configuration space (or subset of the space) can provide insights into which joint can be used most effectively to move the COM in each region of its configuration space.

The next chapter will apply and validate both the generalized gains and the optimization framework introduced in this chapter using several parameterized mechanisms: a set of simple planar mechanisms, a 2D 5-link biped, and a 3D 5-link biped.

Chapter 6

Optimization Results[1]

In this chapter, the generalized gains and optimization framework described in the previous chapter are used to optimize progressively more complex balancing systems. Since the gains are independent of the controller used, invariant to gravitational or velocity product dynamics, and (when balancing on a point or line contact) independent of the contact angle, entire families of mechanisms can be optimized at once.

First, planar velocity and momentum gains are compared in Section 6.1 as criteria for the design of simple balancing systems using the parameterized optimization framework introduced in the previous chapter. For this initial exploratory work, planar 2- and 3-link inverted pendulums and a simple planar 3-link biped are used to determine how optimizing for each of the four planar gains affects the resulting optimized design.

Based on the results of this comparison, angular momentum gain was used as the objective in optimizing the design of a 5-link planar biped and compared the results to optimizing the same mechanism using a cost of transport based objective function in Section 6.2. These two objectives were shown to produce very similar results, even though the angular momentum gain calculation requires only the joint space inertia matrix at each configuration of interest while the cost of transport calculation requires slow hybrid dynamics equations and pre-generated or co-optimized trajectories.

Finally, in Section 6.3 the generalized objective framework and the weighted generalized objective function are validated through the optimized design of a 3D 5-link biped. This validation includes a comparison between using 1-, 2-, and ∞-norms on the gain matrices along with 2 different types of weighting matrices and an examination of how the horizontal gains change throughout a single step of the biped for a given set of parameters.

[1] Earlier versions of portions of this chapter were previously published in [113–115].

6.1 Optimization of Simple 2D Examples

Using the planar objective functions and the optimization framework described in Chapter 5, three examples of simple planar models were selected to help demonstrate and compare the benefits of using velocity and momentum gains as a design criterion (with n_i the number of independent links and n_a the number of actuated joints):

- A 2-link inverted pendulum, with 2 independent links and 1 actuated joint;
- A 3-link biped, with 2 independent links (identical legs) and 2 actuated joints; and
- A 3-link inverted pendulum, with 3 independent links and 2 actuated joints.

Note that in the biped model the legs are assumed to be identical, so only one of the legs is independently parameterized. Three parameters are used for each independent link:

- The length of the link, $0 < l_i \leq 1$;
- The fraction of total mass in each link, $0 < m_i/m < 1$; and
- The position of the link's COM as a fraction of the link's length, $0 < c_i/l_i \leq 1$.

Unless otherwise noted, the elements of $\boldsymbol{x}_{min}$ are all set to a small positive value $\epsilon > 0$, to ensure that the link masses, COMs, and lengths remain within a reasonable range for an actual mechanism. Since the gains are invariant to a scaling of the total mass, assume $m = \sum_i m_i = 1$ (reducing the number of independent parameters by 1 for every model).

For each of the example models, an optimization was run for each objective function. To generate specific key poses to populate $\boldsymbol{Q}$, ranges were set for each joint (passive and active) as shown in Table 6.1 and key poses were automatically generated to uniformly span the desired joint space.

Table 6.1: Joint Ranges used to Generate Key Poses for 2- and 3-Link Models

Joint Type:	**Passive**	**Actuated**	
Model	$\boldsymbol{q_1}$	$\boldsymbol{q_2}$	$\boldsymbol{q_3}$
2-Link Inverted Pendulum	$\pm\pi/4$	$\pm\pi/2$	
3-Link Biped	$\pm\pi/4$	$\pm\pi/2$	$\pm\pi/2$
3-Link Inverted Pendulum	$\pm\pi/4$	$\pm\pi/2$	$\pm\pi/2$

The joint ranges shown in Table 6.1 have been selected to maximize the number of key poses which are likely to be encountered by a real-world mechanism with the given form. With this in mind, the body angle specified for the biped model is relative to a vertical axis to mimic the typically upright bodies of bipeds. Similarly, the biped's swing leg hip joint has been flipped so that an angle of 0 points down, as is typical in biped robots.

6.1.1 2-Link Inverted Pendulum

For this model, both links were parameterized as discussed above, leading to a total of 6 parameters. Since the length of the final link (furthest from the contact point) has no effect on the gains, assume $c_2/l_2 = 1$. This change, in combination with setting $m = 1$, leaves 4 independent parameters.

Although it has been suggested that analysis of a planar 2-link inverted pendulum with only 1 actuated joint does not benefit from using momentum gain over velocity gain [27], in the case of mechanism design there is a benefit to using momentum rather than velocity gains. Since the lengths, masses, and link COMs are free to change in this formulation, the momentum gain includes additional useful information about the mechanism's ability to balance which is not available if the velocity gain is used.

This is demonstrated in Table 6.2 and the diagrams in Figure 6.1a, where the results of running the four different optimizations are summarized. Each optimization has successfully found a parameter set which maximizes the desired gain, but the best solution found differs greatly for each gain. Note that $\epsilon = 0.1$ for these initial examples.

Table 6.2: 2D 2-Link IP Optimization Results

2-Link IP	$\frac{m_1}{m}$	$\frac{c_1}{l_1}$	l_1	$\frac{m_2}{m}$	$\frac{c_2}{l_2}$	l_2
Optim. for G_v	.1	1	1	.9	1	1
Optim. for G_ω	.9	1	.134	.1	1	.831
Optim. for G_m	.9	.1	.1	.1	1	.1
Optim. for G_o	.1	.1	.1	.9	1	1

Table 6.3: 2D 3-Link Biped Optimization Results

3-Link Biped	$\frac{m_1}{m}$	$\frac{c_1}{l_1}$	l_1	$\frac{m_2}{m}$	$\frac{c_2}{l_2}$	l_2
Optim. for G_v	.45	.1	1	.1	1	1
Optim. for G_ω	.45	.1	.232	.1	1	.719
Optim. for G_m	.1	.9	.1	.8	1	.1
Optim. for G_o	.1	.9	.1	.8	1	1

6.1.2 3-Link Biped

For this model, 2 of the 3 links were parameterized as discussed above, leading to a total of 6 parameters. The third link is defined as an identical copy of the first link.

Due to the symmetry created by using identical legs, the upper bounds in $\boldsymbol{x}_{max}$ were set to $1 - \epsilon$ for the leg link COMs and remained set to 1 for all other parameters. Since the length of the body link has no effect on the gain calculations (much like the final link in inverted pendulums), assume $c_2/l_2 = 1$, which leaves 4 independent parameters.

Table 6.3 summarizes the results of the 4 different optimizations using the framework for the 3-link biped model, with associated diagrams shown in Figure 6.1b.

6.1.3 3-Link Inverted Pendulum

For this model, each of the 3 links was parameterized as discussed above, leading to a total of 9 parameters. As before, set $m = 1$ and the COM of the final link at the end of the link ($c_3/l_3 = 1$), which leaves 7 independent parameters.

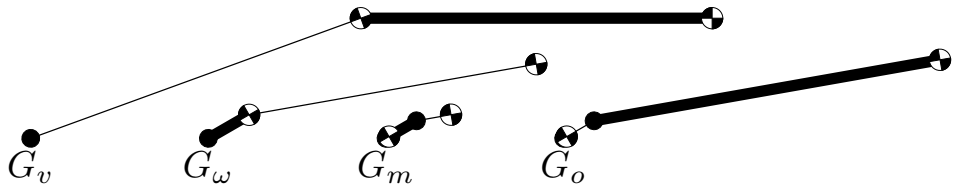
This type of model was used as a design example by Featherstone in [27] where the center of mass of each of the links was fixed to the end of the link (i.e., $c_i/l_i = 1$), leaving 6 independent parameters which were initialized to $m_i = .5$ and $l_i = .3$. They then manually explored the parameter space and compared angular velocity gain plots to determine how to improve on their initial design.

Table 6.4 and the diagrams in Figure 6.1c include the two parameterizations from [27], as well as the results of the 4 different optimizations using the framework. It should be noted that although $m = 1.5$ for the original Featherstone models, the mass fractions are used in the table and gain calculations as it is only the relative mass which matters since the gains are invariant to a scaling of the mass.

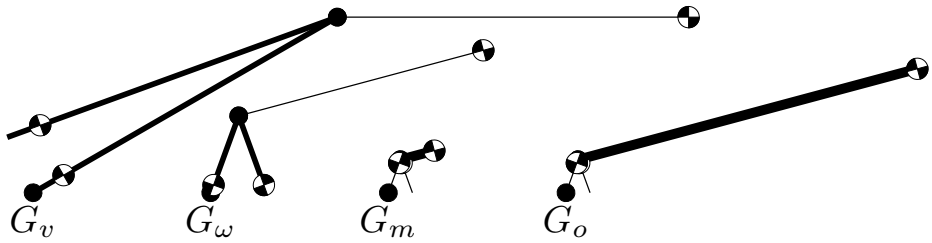
Table 6.4: 2D 3-Link Inverted Pendulum Optimization Results[2]

3-Link IP	$\frac{\boldsymbol{m}_1}{\boldsymbol{m}}$	$\frac{\boldsymbol{c}_1}{\boldsymbol{l}_1}$	$\boldsymbol{l}_1$	$\frac{\boldsymbol{m}_2}{\boldsymbol{m}}$	$\frac{\boldsymbol{c}_2}{\boldsymbol{l}_2}$	$\boldsymbol{l}_2$	$\frac{\boldsymbol{m}_3}{\boldsymbol{m}}$	$\frac{\boldsymbol{c}_3}{\boldsymbol{l}_3}$	$\boldsymbol{l}_3$
Optim. for G_v	.1	1	1	.1	.1	1	.8	1	1
Feath. Initial [27]	.333	1	.3	.333	1	.3	.333	1	.3
Feath. Improved [27]	.467	1	.2	.333	1	.25	.2	1	.35
Optim. for G_ω	.8	1	.115	.1	.1	.1	.1	1	1
Optim. for G_m	.8	.1	.1	.1	.1	.1	.1	1	.1
Optim. for G_o	.1	.1	1	.1	.1	.1	.8	1	1

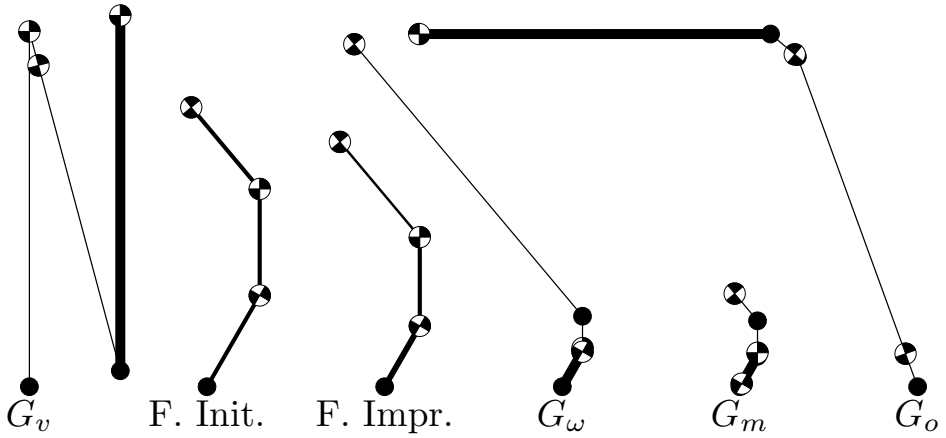
[2]Although $m = 1.5$ for both of Featherstone's models, the mass fractions are used in Table 6.4 and the gain calculations. Only relative mass matters, since all the gains are invariant to a scaling of the mass.



(a) 2-Link Inverted Pendulum Models from Table 6.2



(b) 3-Link Biped Models from Table 6.3



(c) 3-Link Inverted Pendulum Models from Table 6.4

Figure 6.1: Diagrams of the optimized 2- and 3-link planar models which correspond to the optimization results in Tables 6.2, 6.3, and 6.4. In these diagrams, the relative thicknesses of each link denote their relative masses. Note that the configurations of the models' joint angles in these figures are arbitrary.

6.1.4 Observations for Inverted Pendulums

For G_v, placing maximum mass as far from the contact point as possible maximizes every joint's potential ability to move the COM. The links are expected to be as long as possible, since G_v has units of length.

For G_ω, the COM is expected to be as close to the first joint as possible while maintaining the ability to rotate the COM around the contact point by placing a small mass far away. This is due to the invariance of G_ω to changes in q_1 for a point contact (which shifts the desired COM from the contact point to the first joint) and that $G_\omega \propto 1/|\vec{c}|$.

Both momentum gains include their corresponding velocity gains, but the inertial loading on the joints is now included via multiplication by $\bar{\boldsymbol{H}}_{aa}^{-1}$, which in the 2-link IP case reduces to $\bar{H}_{22}^{-1} = H_{11}/det(\boldsymbol{H}) = \Delta\dot{q}_2/\iota_2$.

This effectively makes it expensive to move mass which is far from the actuated joints by penalizing high joint torques. This is qualitatively different from penalizing high joint accelerations, as is the case with velocity gains.

The effects of this are clearly seen in the linear momentum gain (G_m) inverted pendulum examples, with the majority of the mass very close to the contact point. Very short links are also present, as G_m has units of reciprocal length.

Considering the angular momentum gain (G_o) inverted pendulum results, the influence of the additional $|\vec{c}|^2$ term can be seen which causes $G_o \propto |\vec{c}|$: the majority of the mass is placed as far away from the contact as is possible. The remainder of the mass is then placed within the mechanism so as to minimize the required torque demands for balancing (i.e., close to the contact for inverted pendulums).

6.1.5 Observations for 3-Link Biped

For $\boldsymbol{G}_v$, the biped's mass has moved to its feet instead of keeping it far from the contact. This is mainly a function of the selected key poses maintaining the body link above and the feet below the hips, such that a smaller mass in the leg contributes more to the horizontal COM motion for a given joint velocity than a larger body mass.

Effectively, due to the passive contact enforcing conservation of angular momentum about the contact, moving a moderate-sized mass close to the contact requires a smaller passive rotation to compensate, and therefore a larger horizontal COM motion.

These heavy legs are also due to there being no consideration of the inertial loading or location of the COM in the linear velocity gain (G_v) equations, only a ratio between horizontal COM velocity and joint velocity. The biped which is optimized for angular velocity gain (G_ω) also has heavy feet, due to the same effect.

The analog of the momentum gain IP effects are also seen in the biped, where locating most of the mass in the body and placing the COMs of the legs close to the hips means it

is very cheap to move the swing leg. This is the direct opposite of what was shown for the velocity gains, where there is no cost to swing a heavy leg. For G_o, this serves to minimize the required torque demands for balancing (as done in the IPs).

6.1.6 Comparison between Gains

Based on the results above, as well as comparing the formulations of the velocity and momentum gains, several key differences become apparent. These observable differences exist not only between velocity and momentum gains, but also between the linear and angular gains of the same type.

The inclusion of inertial effects in the momentum gains compared to the velocity gains enables them to act as a proxy for energy efficiency. Since momentum gains incorporate a consideration of the amount of joint impulse required to move the COM relative to the contact point, maximizing the momentum gain for a given configuration has the same effect as minimizing the joint torques. Maximizing velocity gains will only provide a proxy for minimizing joint accelerations for a given configuration, which is not equivalent.

Comparing the linear and angular gains of each type, the angular gains are more suitable for mechanism design as they are dimensionless and, when the contact can be approximated as a spherical (rotary in 2D) joint, invariant to the passive contact angle(s).

Both these properties reduce the complexity of the design space, while maintaining the means to quantify the physical ability of a mechanism to move its COM relative to the contact. Using dimensionless metrics removes the dependence on scale from the calculation, allowing angular velocity and momentum gains to be used across a family of different mechanisms at various masses and lengths.

Looking specifically at the biped results in Table 6.3 and Figure 6.1b, it is apparent that designing a biped using velocity gains will produce an inefficient walking mechanism due to the heavy feet. Compare this to the designs based on momentum gain, which place most of their mass in the body and shift the leg masses as close to the hip as possible to minimize hip torques.

Due to the above issues with the linear gains and velocity gains, the preferred gain for mechanism design (at least in the case of a planar biped) is the angular momentum gain, which is used in this chapter for this purpose.

6.1.7 Discussion

Since the linear velocity and momentum gains have units of length and reciprocal length, respectively, they are dependent on not just the relative lengths of the links but the total length (height for biped) of the system. This is directly evident in the examples in the previous section, where all of the links had the maximum length when optimizing for G_v and minimum length when optimizing for G_m.

In light of this, in a more practical application of these gains as part of the framework there would need to be either more restrictive limits set on the lengths for the links (or fixed lengths, if appropriate) or a fixed overall length (or height) which can then be used to parameterize using a ratio of link length to total length.

Other forms of overall objective function were also considered which would change the results or achieve higher gains across more key poses, such as using the median or minimum of the gains over all poses. For example, when using the minimum of the gains, the value of the objective function represents the guaranteed worst case gain for the system across all poses. A critical consideration when using the mean is that it can be more sensitive to the set of key poses used to define the joint ranges than other options.

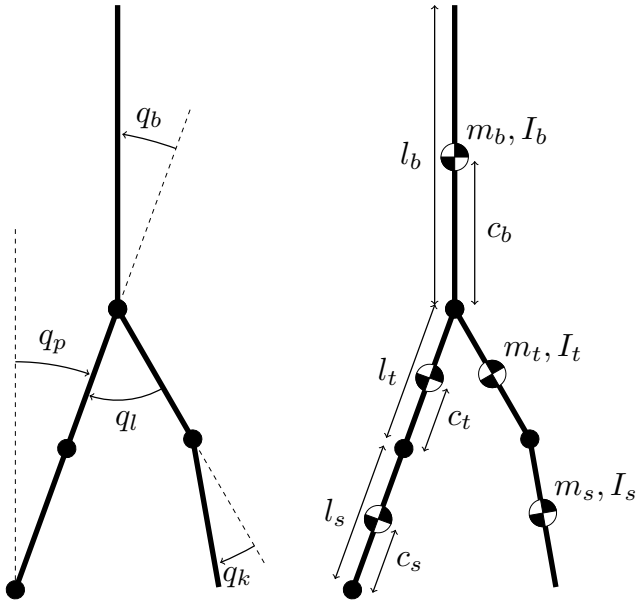
6.2 Optimization of a 2D 5-Link Biped

In this section, the framework is illustrated on a 2D 5-link biped (see Figure 6.2). This mechanism can be used as a simplified representation of a broad range of natural and artificial bipeds, including humans, ostriches, and others [95]. The parameters, parameter mapping and key configuration poses for this mechanism are outlined in the following subsections, along with an alternative objective function based on the Cost of Transport for comparison to the proposed objective function from Section 5.4.1.

6.2.1 Parameters

A modified version of the 5-link biped parameters defined by Haberland and Kim are used [95]. Since the model has a symmetric form, the legs are assumed to be identical so only one set of leg links are independently parameterized.

In [95], the mass and length of the body link are defined in units of kg and m, respectively, and all other lengths and masses are defined relative to these values. However, since



(a) Joint Labels (b) Parameters

Table 6.5: 2D 5-Link Biped Parameters

Parameter	**Ratio**	**Min**	**Max**
Body Link Mass	m_b/m	1/2	3/4
Body Link Length	l_b/l	1/4	1/2
Thigh Link Mass	m_t/m_l	1/4	3/4
Thigh Link Length	l_t/l_l	1/3	2/3
All Link COMs	c_i/l_i	1/4	3/4
All Link Inertias	r_i/l_i	0	2/3

Figure 6.2: Diagram of the 2D 5-link biped model's joints and parameters. In (a), passive rotation about the contact point q_p is measured from a vertical axis, knee rotation q_k is measured relative to a straight leg, body link rotation q_b is measured relative to the stance thigh, and leg rotation q_l is measured from the swing thigh to the stance thigh. In (b), the link parameters are the mass m_i, length l_i, COM c_i, and inertia $I_i = m_i r_i^2$, where the index i is replaced with b, t, or s for the body, thigh, and shank links, respectively. In this figure, quartered circles show the locations of each link's COM and solid circles represent the joints.

the objective functions will be using measures which are invariant to a scaling of the total mass m or length l of the system, all parameters will be defined as ratios.

Four parameters are used for each independent link i: mass m_i, inertia $I_i = m_i r_i^2$, COM c_i, and length l_i. For each of the links, the COM and inertia (via the radius of gyration r_i) are defined relative to that link's length, while the link length and mass are defined relative to other links.

A minimal representation of these ratios is shown in Table 6.5. It's assumed that the system has identical, symmetric legs and the indices b, s and t are used to indicate the body, shank and thigh links, respectively. The mass and length of the leg are then defined as $m_l = m_s + m_t$ and $l_l = l_s + l_t$, respectively, and the total mass and length of the biped are $m = m_b + 2m_l$ and $l = l_b + l_l$, respectively.

Note that in Table 6.5, the mass and length of the body link are scaled by the total

mass and length of the system, respectively, while the mass and length of the thigh links are scaled by the mass and length of the leg, respectively. This eliminates the need to parameterize the (dependent) mass and length of the shank link.

This parameterization consists of 10 independent parameters, which can be satisfied strictly using lower and upper bounds: two mass ratios, two length ratios, and the COM and inertia for each of the three independent links.

The parameter map which applies these parameters to the model, $\boldsymbol{X}$, assumes $m = 50$ kg and $l = 2$ m to facilitate the calculation of the kinematic and dynamic properties. This means the system's masses, lengths, COMs, and inertias can be defined using real values and applied to the model used in the objective function. However, the measures used in the objective function are invariant to m and l, so these values could be chosen arbitrarily with the same results.

6.2.2 Key Poses

For this example, the key poses are used to outline a set of typical motion paths within the configuration space of the biped model. This set of poses is chosen to be representative of the swing phase of a standard walking gait, where the swing leg starts on the ground behind the stance leg and finishes in the same pose but with the leg positions switched.

The leg poses are assumed to be symmetric in the starting/ending configuration. In this type of gait, the stance knee is fully extended for the duration of the stance phase following the results of [116], who showed that optimal periodic gaits for simple bipeds always involve pendular motion in the stance phase, due to the elimination of work when the system is acting as an inverted pendulum.

The considered motion subspace in the biped's configuration space is defined by the following joint angle ranges:

$$\begin{aligned} -\pi/6 \leq \quad & q_p \quad \leq \pi/6 \\ 0 \leq \quad & q_k \quad \leq \pi/3 \\ -\pi/3 \leq \quad & q_l \quad \leq \pi/3 \\ -\pi/4 \leq \quad & q_b \quad \leq \pi/4 \end{aligned} \tag{6.1}$$

To ensure a fair comparison to the cost of transport based objective function defined below, only configurations which are in the CoT trajectories will be sampled to generate key poses. Compared to the larger motion subspace above, this primarily limits the motion of the body link to within $\pi/12$ of vertical and limits the swing leg joint angles to remain within typical walking ranges.

6.2.3 Objective Function

Motivated by the discussion in Section 6.1.7, for this example the angular momentum gain is used to define the objective function for four reasons:

- Momentum gains incorporate inertial information, quantifying the effort needed at the actuated joints to generate COM motion (either for balancing or gait).
- Angular gains are dimensionless, removing any dependence on the total length of the system and allowing link lengths to be parameterized as ratios.
- Angular gains for mechanisms with a point contact (line contact in 3D) are invariant to the passive contact angle.
- Angular momentum gain is defined everywhere, whereas the angular velocity gain approaches infinity as the COM nears the contact point and becomes undefined when the COM is at the contact.

The angular momentum gain is therefore an ideal candidate for use in the objective function: It is a dimensionless measure of how efficiently a system's actuated joints can move the COM around a passive contact, independent of the total mass, total length, gravity, and the passive contact angle between the stance leg and the ground.

Since the angular momentum gain is linear with respect to $\boldsymbol{\iota}_a$, define a gain vector $\boldsymbol{G}_{oa} = [G_{o2}\ G_{o3}\ \dots]$ such that $G_o(\boldsymbol{\iota}_a) = \sum_{i \in a} G_{oi}\iota_i = \boldsymbol{G}_{oa}\boldsymbol{\iota}_a$. If the 2-norm is used to define the step impulse as $||\boldsymbol{\iota}_a||_2 = 1$, then the maximum angular momentum gain for any configuration and parameter pair $(\boldsymbol{q}, \boldsymbol{x})$ is given by $||\boldsymbol{G}_{oa}(\boldsymbol{q}, \boldsymbol{x})||_2$ (as discussed above).

The momentum gain based objective function J_G is then defined as the mean over $\boldsymbol{Q}$ of the maximum angular momentum gains (assuming $||\boldsymbol{\iota}_a||_2 = 1$):

$$J_G(\boldsymbol{x}) = \frac{1}{n_q} \sum_{\boldsymbol{q}} ||\boldsymbol{G}_{oa}(\boldsymbol{q}, \boldsymbol{x})||_2 \qquad \forall \boldsymbol{q} \in \boldsymbol{Q} \tag{6.2}$$

where n_q is the number of configurations in the set $\boldsymbol{Q}$.

6.2.4 Comparison Objective Function

In addition to the momentum gain-based objective function defined above, an alternative objective function is evaluated based on the Cost of Transport (CoT) (the most

commonly used cost function in the literature, e.g., [96–98]). The CoT for a system is defined as a ratio between the energy consumed (W) and the product of the system's weight (mg) and the distance (d) it travels while consuming said energy: W/mgd. For this objective function, the energy required to move along a given trajectory parameterized as shown in Equation (6.3) below is evaluated using Featherstone's spatial2 MATLAB library [117].

Around a nominal swing phase pattern (with a fully extended stance knee and symmetric start/end configurations), the initial/final and midstance poses are varied to increase the effective motion subspace being optimized over.

These variations are parameterized using:

- The initial angle between the legs, $\pi/9 \leq 2\theta_p \leq \pi/3$;
- The swing knee angle at midstance, $0 \leq \theta_k \leq \pi/3$; and
- The body link angle at midstance, $-\pi/12 \leq \theta_b \leq \pi/12$.

If the joint positions are defined as $\boldsymbol{q} = [q_p\ 0\ q_b\ q_l\ q_k]$, using the labels shown in Figure 6.2a, the configurations at the start, middle, and end of the step (with smooth transitions between these poses based on quintic splines) are:

$$\begin{aligned} \boldsymbol{q}_0 &= [-\theta_p \quad 0 \quad -\theta_p \quad -2\theta_p \quad 0\ ]^T \\ \boldsymbol{q}_m &= [\ 0 \quad 0 \quad \theta_b \quad \theta_k \quad \theta_k]^T \\ \boldsymbol{q}_f &= [\ \theta_p \quad 0 \quad \theta_p \quad 2\theta_p \quad 0\ ]^T \end{aligned} \tag{6.3}$$

These trajectories are then used to generate the required joint torques to achieve required motions via hybrid dynamics, to ensure that the trajectories are dynamically feasible and satisfy the dynamic constraints of the model. These joint torques are not necessary for the momentum gain based optimization, but are used in the comparison objective function based on the cost of transport below.

For this purpose, since the ground is flat and impact is ignored, assume the energy consumed is $W = \int \boldsymbol{\tau}_a \cdot \dot{\boldsymbol{q}}_a dt$, where $\boldsymbol{\tau}_a$ is a vector of actuated joint torques. This is similar to the CoT from [98] without the absolute power assumption, which itself is a modified version of the CoT from [97] when dealing with flat ground and no impact.

The objective function then minimizes the average of this CoT over the set of n_ψ trajectories $\boldsymbol{q}(t) \in \boldsymbol{Q}_\psi$ (where the step length d is defined as the distance between the feet at both the start and end of the step):

$$J_{CoT}(\boldsymbol{x}) = -\frac{1}{n_\psi} \int \frac{\boldsymbol{\tau}_a^T \dot{\boldsymbol{q}}_a(t)}{mgd} dt \qquad \forall \boldsymbol{q}(t) \in \boldsymbol{Q}_\psi \tag{6.4}$$

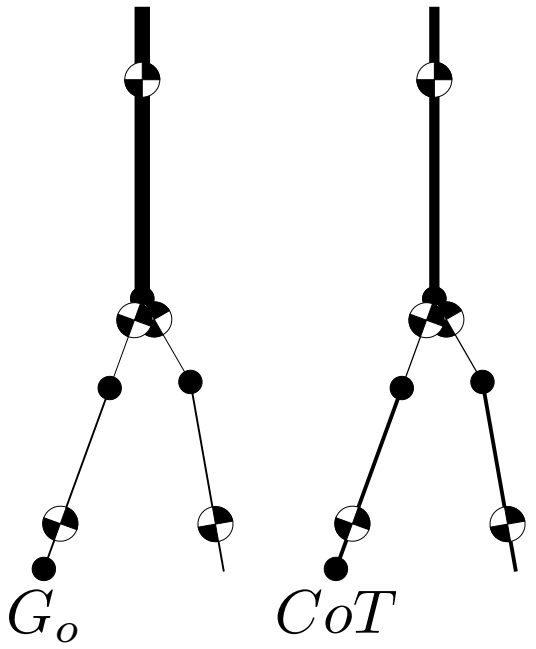


Table 6.6: 2D 5-Link Biped Optimization Results

	Objective Basis	
Parameters	Angular Momentum	Cost of Transport
m_b/m	3/4	1/2
l_b/l	1/2	1/2
m_t/m_l	1/4	1/4
l_t/l_l	1/3	1/3
c_b/l_b	3/4	3/4
c_t/l_t	3/4	3/4
c_s/l_s	1/4	1/4
r_b/l_b	0	0
r_t/l_t	0	0
r_s/l_s	0	0

Figure 6.3: Diagrams of the 2D 5-link biped models which correspond to the quantitative optimization results given in Table 6.6. In the diagrams, the relative thicknesses of each link denote their relative masses, quartered circles show the locations of each link's COM, and solid circles are the joints.

6.2.5 Observations

The results of optimizing the 2D 5-link biped are shown in Figure 6.3. These results show that optimizing for the cost of transport over a set of trajectories provides almost identical results to optimizing for the average angular momentum gain over a comparable configuration space. The only difference between the two results is the mass of the body link: for the CoT results $m_b = m/2$, while for the G_o results $m_b = 3m/4$.

The resulting mechanism obtained by optimizing the angular momentum gain has several characteristic properties: First, the body link has the maximum possible mass and length, and its COM is as far from the hips as possible. This confirms the observations earlier in this Chapter, where the large mass near the top of the body link enables small changes in stance hip angles to produce large COM angular displacements.

Much like the body link, the swing leg has also been optimized to place most of the leg mass near the foot and the remaining mass very close to the hip, allowing both swing leg joints to move the system's COM around with minimal effort. The long shank length,

relative to the thigh, ensures that the swing knee can produce maximal COM motion even if the swing hip is held fixed.

It is interesting to note that, although it was an available parameter, the inertia of all of the links has been eliminated. Although this is not realistic in a real world robot, it is feasible in a model like this as the addition and subtraction of virtual masses at the joints can add inertia back to the links (as described in [27]).

The only difference between the two optimizations is the relative mass of the body link to the total mass. The mass of the body link has been reduced to its minimum possible value in the model optimized for the cost of transport. This is likely due to the cost of transport having no concept of balancing outside of the given trajectories, as well as incorporating gravitational effects which make heavier feet slightly cheaper due to pendular swing leg motion.

6.2.6 Discussion

The similarity in the results is expected, as both the CoT and the G_o optimizations should produce a mechanism which can efficiently move its COM around in the plane. Note that the G_o based optimization was able to achieve similar results to the CoT optimization, without the need for trajectories, torques, an integration over time, or a controller. Despite this similarity in results, there are four main differences between the angular momentum gain and CoT approaches:

First, the cost of transport approach requires a trajectory and/or a controller to be defined. These elements could be either specified [118], or co-optimized in parallel with the physical optimization [97, 98]. The angular momentum gain approach requires only a desired configuration space, defined using a set of key poses which span the space.

Second, the cost of transport approach must include some form of hybrid or inverse dynamics calculation over time to determine the work required to take a step. By comparison, the angular momentum gain approach requires only the calculation of the joint space inertia matrix (or generalized inertia matrix, for systems with kinematic loops) for each configuration of interest, and the inverse of a positive definite symmetric submatrix.

Third, the cost of transport depends directly on the scale of the mechanism (i.e., its total mass and length), which means that it is only effective for a specific design. The angular momentum gain, however, is invariant to scaling of the total mass and/or total length, as well as to various modifications of the inertial properties as discussed in [27], which enables it to optimize an entire family of mechanisms at once. This is primarily due

to defining the gain using impulsive dynamics, which do not include gravitational terms or any velocity-product terms (e.g., Coriolis terms).

Finally, the cost of transport is concerned only with how much effort it takes the mechanism to follow the prescribed trajectory and/or use the prescribed controller. This may result in a system that has excellent performance near the nominal trajectory, but suffers from poor performance if disturbed away from the nominal trajectory. With angular momentum gain, the inherent physical ability of the mechanism to move the COM with minimal effort is maximized.

To improve the biped's balance not only along the specified trajectories generated for the CoT optimization but also throughout the configuration space near them, the motion subspace defined in Equation (6.1) should be uniformly sampled to generate a set of key poses for a subsequent application of the angular momentum gain optimization.

6.3 Optimization of a 3D 5-Link Biped

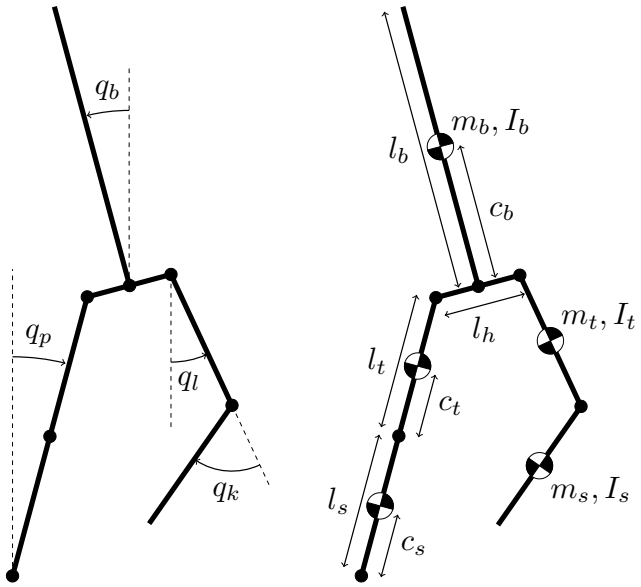
To validate the general pose-specific objective functions defined in (5.29) and (5.30) and show the effects that the different norms and weights have on the resulting parameterization of the model, they are applied to a 3D 5-link biped (see Figure 6.4). This biped has a passive spherical contact (3 DOF), universal hip joints (2 DOF) separated by a fixed offset[3], and a rotary knee joint (1 DOF) in the swing leg.

Similar to the parameterization of the 2D 5-link biped in the previous section, here a minimal representation of mass and distance ratios is used based on the parameters defined in Figure 6.4b (as shown in Table 6.7).

For this model, the parameter map assumes $m = 50$ kg and $l = 1$ m to facilitate the calculation of the kinematic and dynamic properties. However, since all of the gains are invariant to changes in m and the angular gains are invariant to changes in l, the values can be modified with minimal impact.

For this model, key poses have been defined which incorporate a set of typical motion paths within the configuration space of the biped. To generate these key poses, a set of joint ranges has been chosen to approximate the required motions in the swing phase of a standard 3D walking gait where the swing leg starts on the ground behind the stance leg and finishes in the same pose but with the leg positions switched.

[3]Although the relative width of the hips could be used in the optimization, for this work it has been fixed at 20% of the biped's total length.



(a) Joint Labels (b) Parameters

Table 6.7: 3D 5-Link Biped Parameters

Parameter	**Ratio**	**Min**	**Max**
Body Link Mass	m_b/m	1/2	3/4
Body Link Length	l_b/l	1/4	1/2
Thigh Link Mass	m_t/m_l	1/4	3/4
Thigh Link Length	l_t/l_l	1/3	2/3
All Link COMs	c_i/l_i	1/4	3/4
All Link Inertias	r_i/l_i	0	2/3

Figure 6.4: Diagram of the 3D 5-link biped model's joints and parameters. In (a), passive rotations about the contact point q_p, body rotations q_b, and swing leg rotations q_l are all measured relative to a vertical axis, while swing knee rotation q_k is measured relative to a straight leg. In (b), the link parameters are the mass m_i, length l_i, COM c_i, and inertia $I_i = m_i r_i^2$, where the index i is replaced with b, t, or s for the body, thigh, and shank links, respectively. The hip width (l_h) has been set at 20% of the total length l. In this figure, quartered circles show the locations of each link's COM and solid circles represent the joints.

The considered motion subspace in the biped's configuration space is defined by the following joint angle ranges (which, other than q_k, are all relative to a vertical axis):

$$\begin{aligned} -\pi/6 \leq\ & q_{px} && \leq \pi/6 \\ -\pi/18 \leq\ & q_{py} && \leq \pi/9 \\ -\pi/12 \leq\ & q_{bx} && \leq \pi/12 \\ -\pi/18 \leq\ & q_{by} && \leq \pi/18 \\ -\pi/3 \leq\ & q_{lx} && \leq \pi/3 \\ -\pi/9 \leq\ & q_{ly} && \leq \pi/18 \\ 0 \leq\ & q_k && \leq \pi/3 \end{aligned} \tag{6.5}$$

Next, the results of optimizing the parameterized model across this range of motions for the frontal (x), sagittal (y), and combined horizontal (xy) gains are compared using:

- Unweighted 2-norms on the xy gains;
- Unweighted 2-norms on the x and y gains;
- Unweighted ∞-norms on the xy gains;
- Gain-weighted 2-norms on the xy gains;
- Unweighted 1-norms on the x, y, and xy gains; and
- Input-weighted 2-norms on the xy gains.

6.3.1 Results

The resulting optimized parameter sets are summarized in Figure 6.5 and Table 6.8, where duplicates have been combined under a single label wherever possible. In cases where the only difference between two (or more) optimized parameter sets are the locations of their link COMs, a numbered label is used to distinguish between the lettered sets.

The high-level observations of these results are:

(1) In general, optimizing a mechanical design for a particular gain will result in a better performing system than any heuristically generated design or even an optimal parameter set from another family of mechanisms. For example, based on the results in the previous section one might assume that the mechanism with the best overall balance performance would be parameter set F (which has the same parameters as the optimal 2D 5-link biped). However, optimizing for the overall horizontal angular momentum gain (and by extension the 3D balance performance) leads to model G, which is very different.

(2) The design generated differs depending on the chosen objective function, with trade-offs between mobility in one direction vs another. This means that a mechanism which is optimized in one direction is typically poor in the other direction. However, based on the numerical results in Table 6.9, one can see that in some cases a slight reduction in an optimal gain can dramatically increase the gain in the perpendicular direction. See Subsection 6.3.1 for specific examples of these direction-dependent trade-offs.

(3) Using velocity gain(s) results in different mechanisms compared to using momentum gain(s), and each momentum gain objective also results in a different mechanism. The results in Table 6.9 illustrate that most of the velocity gain objectives result in the same

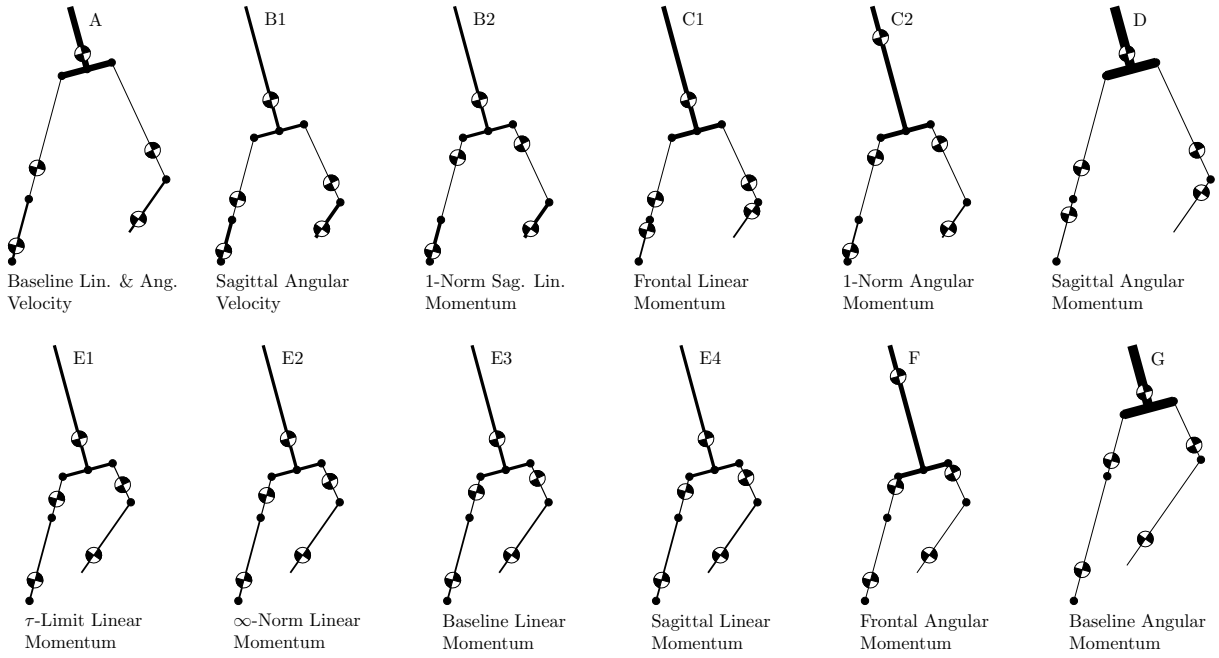


Figure 6.5: Diagrams of the 3D 5-link biped models which correspond to the results in Table 6.8. The relative thicknesses of each link denote their relative densities (i.e., short heavy links are thickest, long light links are thinnest). Note that the configuration of the models' joint angles are arbitrary but identical.

parameter set as they do not consider the additional inertial information used to generate momentum gains. In addition, every single optimized momentum gain objective results in a different parameter set, providing evidence that momentum gain objectives lead to quantifiably different designs even for the reduced set of parameters used in this section.

(4) Further, it is important to consider that gains are pose-dependent and will change throughout a given motion and/or configuration space. In Subsection 6.3.1, the continuously changing effect of a motion on the gains is shown. Gain ellipses are used to give a comparable measure to the ground-projected dynamic COM manipulability from [23].

(5) One final general observation about the results in Table 6.9 is that there are families of parameter sets that are good or bad at different types of motions (i.e., have different levels of gain for different types of gains). For example, models A, B1, B2, and E1-4 all have good general angular velocity gains, which leads directly to good balance performance for systems where the inertia is negligible. However, once inertia is taken into account (i.e., in the angular momentum gains) the models with a higher overall COM (C2, F, and G) replace B1, B2, and E1-4 as the peak performers.

Table 6.8: 3D 5-Link Biped Optimization Results[4]

Label & Main Gain	$\frac{m_b}{m}$	$\frac{l_b}{l}$	$\frac{m_t}{m_l}$	$\frac{l_t}{l_l}$	$\frac{c_b}{l_b}$	$\frac{c_t}{l_t}$	$\frac{c_s}{l_s}$
A: Baseline Velocity	1/2	1/4	1/4	2/3	1/4	1/4	1/4
B1: Sagittal Ang. Vel.	1/2	1/2	1/4	2/3	1/4	1/4	1/4
B2: 1-Norm S. Lin. Mom.						3/4	
C1: Frontal Lin. Mom.	3/4	1/2	1/4	2/3	1/4	1/4	3/4
C2: 1-Norm Ang. Mom.					3/4	3/4	1/4
D: Sagittal Ang. Mom.	3/4	1/4	1/4	2/3	1/4	1/4	3/4
E1: τ-Limit Lin. Mom.	1/2	1/2	1/4	1/3	1/4	.450	1/4
E2: ∞-Norm Lin. Mom.						.540	
E3: Baseline Lin. Mom.						.614	
E4: Sagittal Lin. Mom.						.631	
F: Frontal Ang. Mom.	3/4	1/2	1/4	1/3	3/4	3/4	1/4
G: Baseline Ang. Mom.	3/4	1/4	1/4	1/3	1/4	1/4	1/4

Baseline: Unweighted 2-Norms

First, a set of baseline results were generated by conducting optimizations using the horizontal components of the four different gains based on the matrix 2-norm with both weighting matrices set to appropriately sized identity matrices.

As shown in Table 6.9, the baseline results for the linear and angular horizontal velocity gain optimizations both correspond to model A in Figure 6.5 and Table 6.8. However, the linear and angular momentum gain optimizations resulted in very different parameter sets: E3 and G, respectively.

Compared to the velocity gain results of model A, the linear momentum gain results in E3 maintain the division of mass between the links but the COMs of the thighs shift towards the hips and the relative link lengths change, with the body and shank links growing to maximum length and the thighs shrinking to their minimum.

[4]Note that the r_i/l_i parameters are left out of this table, as the link inertias were optimized to 0 for every set of results generated in this work. In a physical robot this is not realistic, but for a model like this it is feasible since inertia can be added back to the links via the addition and subtraction of virtual masses at the joints (see [27] for details).

Table 6.9: 3D 5-Link Biped Objective Function Values for Selected Parameter Sets

Each of the numerical entries in this table represents the value of the global objective function $J(\boldsymbol{x})$, as defined in (5.20): the mean of the given pose-specific objective function $J(\boldsymbol{q}, \boldsymbol{x})$ evaluated across the set of n_q poses $\boldsymbol{q}_i \in \boldsymbol{Q}$ spread uniformly throughout the configuration space defined in (6.5) for the given parameter set $\boldsymbol{x}$. When defining these objective functions, the subscripts on the gain matrix or vector indicate which gain is being optimized while the subscript on the norm indicates which type of norm is being applied. Note that in this section, and by extension this table, models are only optimized for the horizontal components of the gains so the angular gains do not include a vertical component.

	Parameter Set ($\boldsymbol{x}$)											
$J(\boldsymbol{x})$	**A**	**B1**	**B2**	**C1**	**C2**	**D**	**E1**	**E2**	**E3**	**E4**	**F**	**G**
$\|\boldsymbol{G}_{va}\|_2$	0.1956	0.1346	0.1237	0.06235	0.06084	0.08505	0.1162	0.1154	0.1147	0.1146	0.05755	0.09023
$\|\boldsymbol{G}_{\omega a}\|_2$	0.2449	0.2227	0.2007	0.08022	0.07628	0.08483	0.1823	0.1808	0.1797	0.1795	0.07216	0.09339
$\|\boldsymbol{G}_{ma}\|_2$	43.87	36.94	37.49	28.43	36.36	28.90	42.64	43.64	43.91	43.90	41.75	41.45
$\|\boldsymbol{G}_{oa}\|_2$	21.92	13.87	14.52	14.38	23.14	18.69	16.10	16.56	16.83	16.87	25.84	26.01
$\|\vec{G}_{vxa}\|_2$	0.1175	0.09465	0.08723	0.04587	0.04488	0.05354	0.08186	0.08128	0.08081	0.08071	0.04289	0.05513
$\|\vec{G}_{vya}\|_2$	0.1376	0.08353	0.07711	0.03899	0.03781	0.06026	0.07387	0.07339	0.07302	0.07294	0.03650	0.06485
$\|\vec{G}_{\omega xa}\|_2$	0.1837	0.1513	0.1384	0.05753	0.05264	0.06636	0.1316	0.1306	0.1298	0.1297	0.05256	0.07382
$\|\vec{G}_{\omega ya}\|_2$	0.1352	0.1463	0.1309	0.05277	0.05095	0.04638	0.1165	0.1154	0.1145	0.1143	0.04740	0.04838
$\|\vec{G}_{mxa}\|_2$	19.50	19.05	7.359	25.59	7.034	24.99	12.65	10.97	9.913	9.720	9.212	17.93
$\|\vec{G}_{mya}\|_2$	38.11	32.22	37.08	14.78	35.95	14.12	41.34	42.78	43.27	43.29	40.91	36.61
$\|\vec{G}_{oxa}\|_2$	18.51	11.18	14.42	5.819	23.02	7.033	14.84	15.78	16.31	16.40	25.53	21.24
$\|\vec{G}_{oya}\|_2$	9.604	7.478	2.139	13.06	3.665	16.60	5.832	4.828	4.081	3.928	5.129	12.43

For the angular momentum gain results in G, there are only two changes from model A: the mass of the body increases to its maximum (lowering the relative mass in the leg links) while the long shanks and short thighs appear again, although this time with no change in the COMs of any link.

These results echo those in Section 6.1, where optimizing for linear momentum gain caused a shift of mass closer to the hips while optimizing for angular momentum gain typically led to most of the mass moving further away from the contact.

Intuitively, having heavy "feet" and long legs will result in larger velocity gains, where small hip velocities can produce large COM motions. Since momentum gains include inertial information, they are increased by shifting mass away from the feet towards the hips and body as this reduces the relative torques required to move the COM.

A Priori Direction Selection: Frontal and Saggital

To determine how many of the differences between the 2D and 3D 5-link biped results are due to rotating in multiple planes, optimizations were run to determine the parameters which would result in the maximum gain in each of the horizontal directions separately. Other than selecting which gain was included in the objective, these results used the same formulation as the baseline optimizations.

As shown in Table 6.9, the optimal frontal (x) and saggital (y) linear velocity gain parameter sets and the optimal frontal angular velocity gain parameter set are all model A, the same as the baseline velocity gain optimizations. However, the optimal sagittal angular velocity gain parameter set is B1, where the relative length of the body increases to half the model's length (from 1/4 the length in model A).

When optimizing for momentum gains, differences between the frontal and saggital results emerge when they are optimized independently. These differences can be quantitatively and qualitatively analyzed by comparing the four different parameter sets generated by these optimizations (corresponding to models C1, D, E4, and F).

The main observation here is that the baseline horizontal momentum gain optimization results (E3 and G) are each dominated by motions in a different direction:

- The baseline linear momentum results (E3) are dominated by the sagittal dynamics, with only a slight shift in c_t from the sagittal linear momentum results (E4).
- The baseline angular momentum results (G) are dominated by the frontal dynamics: the frontal component of the objective function for set G is almost double the frontal component (and almost $5\times$ for set F).

Comparing the frontal linear momentum results (C1) to the baseline results (E3), most of the mass has shifted to the body link, the thigh link gets as long as possible, and the COM of the leg links shift towards the knee. For the sagittal angular momentum set (D) compared to (G), the thighs get longer and the shank COMs shift towards the knees.

Although the baseline angular momentum results are dominated by the frontal dynamics, the doubling in sagittal angular momentum gain when switching from F to G increases the overall angular momentum gain enough to compensate for the relatively small drop in frontal gain. This also helps to demonstrate a key observation: the overall horizontal gain is always greater than either individual horizontal gain but smaller than their sum, due to the use of induced norms.

It is interesting to note that the non-dominant momentum gain results, corresponding to sets C1 and D, differ only in the relative length of the body link and have the top

two values for both the non-dominant momentum gain directions. They are also the two parameter sets with the lowest dominant momentum gains and lowest baseline linear momentum gain, but these differences are almost unnoticeable when looking at the baseline angular momentum gain.

Further, looking at models B2 and C1, where the objective values for the angular momentum gain are separated by less than 1%, the overall performance of the models might appear to be similar. However, looking at sizable differences in the directional angular momentum gains for these models shows that they will actually perform differently in the two given directions. This disparity between the baseline and directional gains justifies the need for weighting and selection when optimizing mechanisms for specific tasks.

Automatic Direction Selection: Unweighted ∞-Norms

When the gain needs to be optimized in a single direction but the direction is not known *a priori*, an ∞-norm can be used on the gain matrix to automatically select the single direction with the highest gain in each configuration.

To verify this, each 2-norm used to generate the baseline results was replaced with an ∞-norm. This formulation should find a parameter set which is similar to both the baseline and dominant *a priori* results of Sections 6.3.1 and 6.3.1, respectively.

Although the velocity gain results for this formulation are identical to the baseline results (see model A), the momentum gain results include some interesting differences. As expected, when using the ∞-norm, the linear momentum gain results (model E2) are similar to both the baseline and sagittal 2-norm results (E3 and E4, respectively).

However, the COM of the thighs has shifted further down the leg, almost to the halfway point between the hip and knee. This shift is in the opposite direction from the shift between the baseline results and the dominant sagittal *a priori* results, giving an indication that there are configurations where the frontal component of the gain is a larger contributor to the overall linear momentum gain than the sagittal dynamics.

Similarly, the angular momentum gain results (model G) are the same as the baseline results and shares features with the (dominant) frontal results (model F). This is a strong indication that the sagittal dynamics play a minor but critical role in the overall angular momentum gain formulation.

The information gained from an optimization using ∞-norms could also be used for a mechanism with fixed inertial and kinematic parameters: This optimization would provide insight into which directions the system can control the COM more easily (and by extension, balance and move).

Weighted Directions: Gain Weighting Matrix

Unlike the optimizations in Sections 6.3.1 and 6.3.1, which focus on maximizing the gain due to a single component (manually or automatically determined) of the generalized gain for each configuration, there are often applications where multiple components of the generalized gain are important to consider. This is where the 2-norm is generally used, which will result in maximizing the largest possible gain via a combination of the included gain components.

However, there are cases where one of the gain components may be more critical to the performance of the desired behavior. In this case, the gain weighting matrix can be used to bias the optimization towards a particular component without removing the others from the formulation entirely.

To demonstrate, two different gain weighting matrices were added to the baseline formulation. To bias the optimization towards the frontal plane, a weighting matrix was used which emphasizes the gain in the x direction:

$$\boldsymbol{W}_G = \boldsymbol{W}_{Xy} = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.6}$$

To bias towards the sagittal plane, a weighting matrix was used which emphasizes the gain in the y direction:

$$\boldsymbol{W}_G = \boldsymbol{W}_{xY} = \begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix} \tag{6.7}$$

Comparing the results of these weighted gain optimizations to the *a priori* direction selection results from Section 6.3.1, when a given plane is emphasized the results of the weighted optimization are identical to the results from the *a priori* direction selection when the same plane is selected. For example, when the x component is emphasized over the y component using a gain weighting matrix for this model, the results are the same as if the x direction was chosen *a priori* (and vice versa).

Scaling the relative values of the weights in the $\boldsymbol{W}_G$ matrix can therefore be used to shift an objective across a continuous subspace of gains, ranging from a complete focus on one direction (as seen in Section 6.3.1) to the equally weighted contributions from multiple directions used in Section 6.3.1.

Automatic Joint Selection: Unweighted 1-Norms

If a single known joint is to be used to realize balancing behavior, the formulation will reduce to the maximization of the chosen vector norm of a column vector (or a scalar, if only one component of the gain is being used).

However, if the goal of the optimization is to determine which single joint should be used for balancing in any particular configuration, possibly to free up the rest of the system's DOFs for other tasks, then the matrix 1-norm can be used in place of the 2-norm from the formulations of the baseline results. As with the previous optimizations, all velocity gain results were identical to the baseline results.

The linear momentum gain results using the 1-norm are different from any other linear momentum results, with the same parameter set as the frontal angular momentum gains (see model F). The angular momentum gain results (model C2) are closer to the rest of the angular momentum results, although they still differ from the (dominant) frontal angular momentum gain results (F) by switching the relative lengths of the thigh and shank links.

The 1-norm can also be applied to the *a priori* direction results, assuming the matrix norm is used for the single-row gain vector, to determine which joint contributes the most to the selected component of the gain in each pose. For these optimizations, all of the velocity gain results and two of the four momentum gain results (the non-dominant frontal linear momentum and sagittal angular momentum) were identical to their corresponding *a priori* results using the 2-norm.

The frontal angular momentum gain results were the same as the angular momentum 1-norm results (model C2), as expected since the angular momentum gain is dominated by the frontal dynamics. However, the sagittal linear momentum gain results (model B2) show a similar leg length switching behavior to what was seen for the angular momentum gain results above, when compared to the baseline and sagittal linear momentum gain results that used the 2-norm (corresponding to models E3 and E4, respectively).

These leg length adjustments when switching from the 2-norm to the 1-norm suggest that if a system would benefit from using only a single joint (or a reduced set of joints) for balancing, that longer links further from the contact will provide more relative gain. Intuitively, this makes sense as it implies that the subset of joints being used for COM motion would require longer lever arms to move distant masses.

An additional key observation here is that, even if a mechanism has already been designed and the inertial and kinematic parameters are fixed, an optimization which uses 1-norms to evaluate the gains over a given configuration space can provide insight into which joint can be used most effectively to balance. If the optimization is run on different

areas of the configuration space, this could also be used to map which joint should be used in which regions of the space to maximize the model's COM motion capabilities.

Weighted Inputs: Actuator Weighting Matrix

In cases where the joints are not all equivalent (different motors, energy sources, etc.) it is often useful to be able to weight these inputs to provide both an indication of relative importance and to avoid joint limits (e.g., maximum torque). This is also useful when it is desirable that only a subset of joints are to be used for balancing, as in the previous section, which can be given much higher weights than non-balancing joints to drive the solution towards a desirable outcome.

To this end, the generalized objective function also incorporates the ability to apply a weighting matrix to the input vector. As an example, the input vector weights could be similar to those used in [23] for COM manipulability, which correspond to defining either the relative importance of the joint accelerations or the maximum torques.

For this example, the following input weighting matrix was applied to the input vector, which prioritizes using the hip joints over the knee joints for balance:

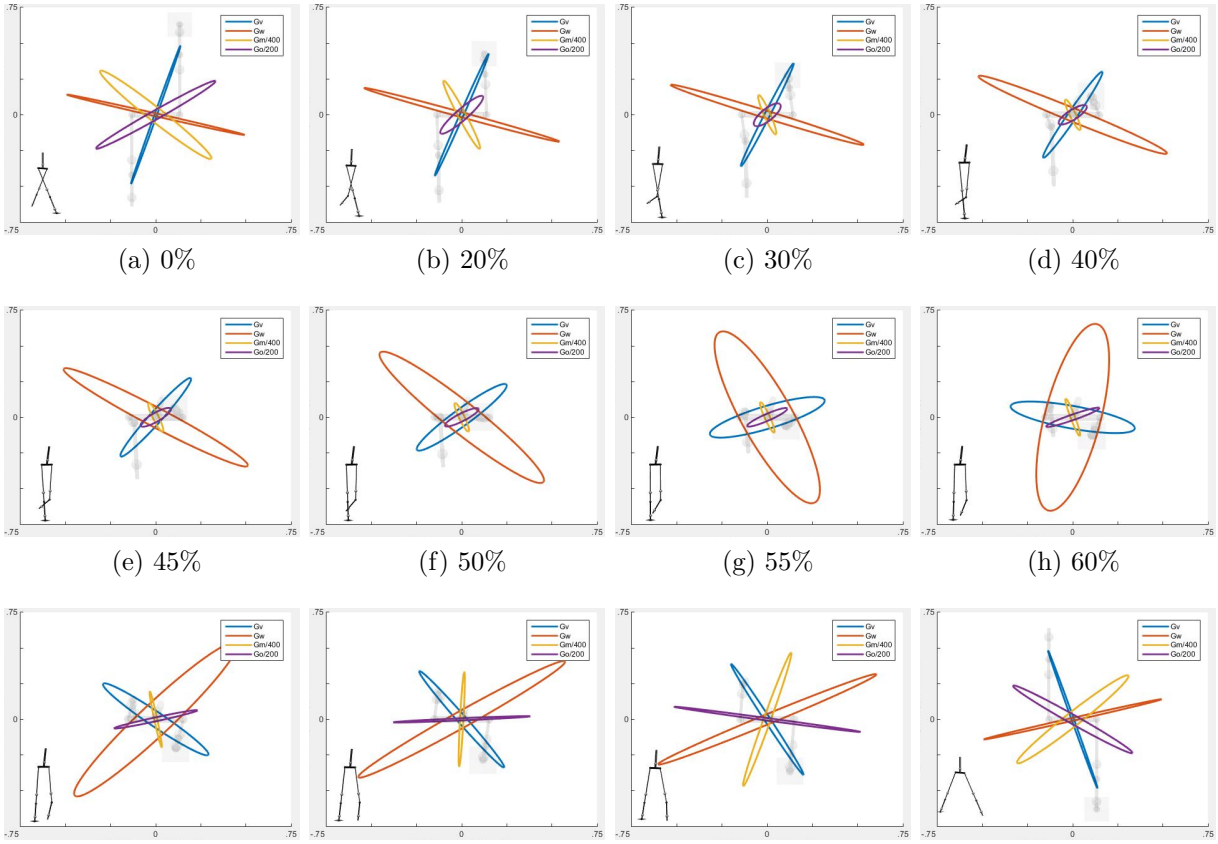
$$\boldsymbol{W}_a = 6 \cdot \mathrm{diag}(1,\ 1/9,\ 1/4,\ 1/4,\ 1/9,\ 1) \tag{6.8}$$

With this weighting matrix, the parameter sets either closely or exactly correspond to the automatic direction selection results (using ∞-norms) generated in Section 6.3.1, corresponding to models A, E1, E2, and G.

Variations In Gain due to Pose

In addition to the changes in overall gain due to differences between the parameter sets, it is also critically important to understand how the gains change based on a model's pose. As shown in Figure 6.6, the magnitudes and directions of each of the gains can vary substantially throughout a given motion, even for a single parameter set (in this case, A).

For this model, the main reason for these changes in gain direction and magnitude is the motion of the swing leg masses relative to the model's passive contact point, which can cause dramatic shifts in the model's COM and other inertial properties. Since the gains are effectively quantifying the potential of the system to move its COM, longer ellipse axes



(a) 0% (b) 20% (c) 30% (d) 40%

(e) 45% (f) 50% (g) 55% (h) 60%

(i) 65% (j) 70% (k) 75% (l) 100%

Figure 6.6: Variations in horizontal gains throughout a single step for the 3D 5-link biped model using a single parameter set (A). The four ellipses in each graph show the magnitudes and directions of the four different baseline horizontal gains for each selected pose, with linear velocity gain in blue, angular velocity gain in orange, scaled linear momentum gain in yellow, and scaled angular momentum gain in purple (the scaling of the momentum gains in this figure is purely for the purpose of visual clarity). The percentage labels on these graphs indicate progress through a complete swing phase (note the symmetry between (a) and (l)), while the black stick figure in the lower left of each graph and the underlying gray stick figures show the pose of the model at each point. In this figure, gain ellipses based on 2-norms are used to facilitate comparisons with the ground-projected dynamic COM manipulability defined in [23].

in Figure 6.6 correspond to more capacity in the inertial structure of the model to move its COM in that direction.

The gains are also symmetric, since the gains are entirely a function of the static configuration of the joints and the parameters of the model. The symmetry also means fewer gain calculations are required, as the gains can be mirrored from one half of the configuration space to the other.

6.3.2 Discussion

As introduced in Section 6.3, optimizing a system's gains is equivalent to maximizing the *potential* of that system to be able to move its COM and, by extension, to walk and balance. Applying the optimization framework and generalized objective function defined in this work to a parameterized model will therefore maximize the potential of the given mechanism to effectively move its COM in a region of its configuration space, independent of the controller used or any predefined trajectories within that space.

This optimization provides an upper limit on how well the mechanism can balance and walk using any controller in a range of configurations in the desired motion subspace, avoiding overfitting to a specific controller and/or trajectory. Additionally, the gains are effectively a quantification of the peak COM motion capabilities that could be achieved with a 'perfect' controller, so they could be used as a metric for how close to optimal a controller can move the system's COM.

Currently, humanoid robot performance is compared after the physical mechanism, controller, and trajectories have all been designed. Many of these design choices (especially in the mechanism design) are made through a combination of heuristics and physical fabrication constraints. The optimization approach proposed in this paper also helps to resolve computationally some of the legged robot design trade-offs that are currently approached heuristically: heavy vs. light feet, long vs. short legs, etc.

By providing designers with a quantitative way to compare between different designs (kinematic, inertial, or otherwise), the optimization can even be used to select a reduced region of parameter space within which to heuristically design a system.

To account for real world effects, such as joint limits and energy costs, here weighted norms were used in place of the standard input vector norms (as introduced in [27]). Recent work on dynamic COM manipulability provides some evidence for the benefits of this type of approach, where the input vector has been weighted to account for the torque and acceleration limits of the mechanism's actuators or the relative importance of minimizing

joint accelerations [23]. The weighting matrices from [23] were also applied in this work to instantaneous changes in velocity and momentum.

As discussed in [27] and Chapter 5, different norms on the actuator input vectors are useful in different scenarios:

- The ∞-norm is an appropriate choice if the most critical issue in the optimization is the effects of joint limits (provided each joint's limits are independent).
- The 1-norm is appropriate if energy consumption is the most critical issue and the energy cost of moving a joint is proportional to its entry in the input vector.
- The 2-norm should be used if the energy cost of moving a joint is proportional to the square of the input vector entry, or if other 2-norm based metrics are being used.

Building on this discussion, general guidelines are provided for choosing the right gain, norm, and weighting matrices given a desired task and mechanism: Typically, as discussed in Section 6.2, using an unweighted 2-norm to optimize the angular momentum gain will generate similar results to the standard cost of transport for legged locomotion. This is also a good choice for systems that will be using dynamic balance and require consideration of the inertial rotation about their passive support point (e.g., inverted pendulums).

If a very specific COM motion task must be achieved by the mechanism, to the exclusion and/or detriment of any other tasks, then optimizing for a single gain (or subset of gains) which correlates directly to the direction and goals of that task is recommended.

However, when a more general mechanism capable of achieving several different tasks is desired the balance of objective formulation shifts. In this case, all of the gains associated with the various tasks should be included in the objective formulation and a gain weighting matrix can be used to prioritize between them as needed or compensate for differences in scale and/or units. The constraints on the inputs will again dictate the type of norm and the input weighting matrix, although with a general system typically the 2-norm and the identity matrix will be appropriate.

Ideally, the goal would be to increase the gains in directions where stability is lacking to facilitate recovery motions when needed. Note that these gains represent the peak COM motion performance that any controller will be able to achieve for the given model and parameter set.

Therefore, a model with a high gain and a simple controller can potentially achieve the same COM motion performance as an optimal controller on a low-gain model. In other words, similar performance can be achieved with a simple controller running on an optimized mechanism compared to an optimal controller running on the original mechanism.

Due to the strong dependence of these (and other) gains on the configuration and parameters of the model, there is not yet a clear and effective method of manually designing a high-gain system. This is the key motivating factor behind the development of the generalized optimization framework in Section 5.4 and the validation of the components of the framework that has been described in this section.

6.4 Summary

In this chapter, the generalized gains defined in Chapter 5 have been applied as part of a parameterized design optimization. This serves both to validate the gain formulations and the optimization framework which uses them as part of the objective function.

The application of the novel gain-based optimization framework defined in Chapter 5 has been demonstrated using three sets of parameterized mechanism design optimizations:

- First, a set of optimizations were performed on three simple planar mechanisms over four different planar gains and the results were compared to the expected outcomes from a theoretical perspective, which led to angular momentum gain being found as the most useful for performing mechanism design;
- Second, angular momentum gain was used in the objective function for a 5-link planar biped mechanism, giving results very similar to those found using an objective function based on the cost of transport (the typical objective for walking mechanism optimization), while requiring less computational effort and not being limited to selecting or generating a specific controller or trajectory.
- Third, the generalized objective function was used to compare the results of different gain-based optimizations using three different types of norms and several different weighting schemes for a parameterized 3D 5-link biped mechanism.

Chapter 7

Conclusions

In this thesis, improvements to the performance metrics used both to control balancing bipeds and to quantify their physical balance capabilities have been developed. These improvements will enable bipeds and other legged robots to safely walk and work in human environments, even when faced with external disturbances.

An online gait generation and tracking controller capable of responding to unknown disturbances while walking or standing was introduced in Chapter 3. The controller uses the 3D FPE to locate the point on the ground where the robot must step to regain stability following a disturbance. The 3D FPE location is re-computed at each time-step, allowing the robot to adapt to disturbances online by adjusting its desired stepping location.

A simple state machine is used to generate a full gait cycle: the robot initiates lifting of the swing leg either when forward progress is desired or when a disturbance requires it, and tracks the target foot placement location to determine the swing leg placement. This controller was tested in simulation and shown to generate stabilizing foot placements in response to disturbances from arbitrary directions, both while standing still and in motion.

A novel balance point estimator called the Spherical Foot Placement Estimator (SFPE) was introduced in Chapter 4. It has been formulated to overcome the drawbacks of existing balance point estimators, by combining and extending desirable features of several estimators to produce a generalized, predictive 3D formulation (including impact and inertia).

The SFPE was compared to other balance point estimators, and shown to outperform them by providing recovery step location prediction and momentum objectives with smooth dynamics. An SFPE-based feedback loop was used in a momentum-based controller as an example of how to add leaning to an existing whole-body controller, and a dynamic SFPE-based stepping strategy was used to deal with large disturbances.

In Chapter 5, Featherstone's initial definition of momentum gain for planar 2-link inverted pendulums was extended to general 2D and 3D mechanisms by using simplified notation to facilitate the definition of both the velocity and momentum gain equations. Two different methods for calculating these general momentum gains were described, namely the Augmented Inertia method and the Spatial method.

The Spatial method provides insight on the relationship between a system's centroidal momentum and its velocity and momentum gains. The definition of the spatial gain of a system incorporates both angular and linear components in all 6 (translational and rotational) directions of 3D motion, and enables the definition of an actuated centroidal momentum matrix which was shown to be equivalent to the spatial gain matrix.

Building on the spatial definitions of the velocity and momentum gains, a generalized gain formulation was defined for walking and balancing mechanisms that concisely quantifies their COM motion properties and encapsulates and clarifies the relationship between velocity and momentum gains. This novel gain formulation provides an easily defined metric for analyzing the COM motion capabilities of any mechanism using passive contact to interact with its environment, which enables quantitative comparisons between different mechanisms across the design space.

At the end of Chapter 5, these generalized gains were then used to formulate an optimization framework for the design of parameterized mechanisms, without the need to specify or generate controllers or trajectories. To this end, the generalized gain formulation was combined with weighted matrix norms to produce a generalized objective function for optimizing the design of parameterized mechanisms. Since these gains are all invariant to a scaling of total mass and the angular gains are invariant to a scaling of total length, entire families of mechanisms can be optimized in one application of the framework.

This generalized framework and objective function was then demonstrated using a series of progressively more complex balancing mechanisms in Chapter 6. First, the simplest gains were compared as objective metrics for planar inverted pendulums and a 3-link biped.

Based on the results of these comparisons, a 5-link planar biped mechanism was optimized using angular momentum gain. It was shown that the results were very similar to those found using an objective function based on the cost of transport (the typical objective for mechanism optimization), while requiring less computational effort and not having the optimal motion potentially overfit to a specific controller and/or trajectory.

Finally, to demonstrate the wide variety of possible objective functions that are enabled via the generalized gain formulation, the framework was applied to a parameterized 3D 5-link biped using 3 different types of norms and several different weighting schemes.

7.1 Contributions

The contributions of this thesis to the state of the art are as follows:

C1. Dynamic Balance and Gait Strategy

A dynamic balance and gait control strategy based on an existing 3D balance point estimator and a continuously updated impact plane, which uses both leaning and stepping balance strategies to respond to external disturbances while standing and walking.

The goal of this overall strategy is to allow the robot to respond to external disturbances by stepping onto the 3D FPE, either while standing or walking. One of the key benefits of this strategy is that recovery from external disturbances is a subset of the elements required for dynamic gait. To walk, a robot need only push itself into an unstable state in the desired direction of motion and allow the recovery mechanism to avoid a fall.

To evaluate the proposed approach, the 3D FPE [20] is used to find a balance point for a 14-DOF lower-body humanoid robot. This point is then used as a control reference to inform both a high-level state machine and its associated task-level trajectory generator. The task-level trajectories are used as control inputs to a prioritized Jacobian-based feedback loop [60] and a simple low-level PD joint controller to drive a simulated robot.

Both [18] and [60] have investigated earlier versions of this form of state machine based control strategy. However, [18] deals only with planar bipeds with point feet, while [60] uses a constant plane of motion, requires separate lateral stabilization, and has very limited discussion of external disturbances. The approach developed in this thesis uses a continuously updating plane, which allows it to respond in 3D to unknown external disturbances. It also includes dynamic gait generation, unlike [14], and is used directly as part of a unified control method, as opposed to a measurement method such as in [20].

C2. Spherical Foot Placement Estimator

A novel balance point estimator which combines and extends the desirable properties of existing estimators while removing or reducing the majority of their drawbacks.

Building on the work in Chapter 3 and the definitions of the FPE, GFPE, and 3D FPE [14, 18, 20], a novel 3D balance point estimator called the Spherical Foot Placement Estimator (SFPE) was developed. The SFPE maps the biped's current kinematics and

dynamics to a simplified compass gait model and projects its motion into two perpendicular planes. These projections are used to continuously predict whether or not a step is needed, and if so where to step, in order to maintain or restore balance in 3D.

This novel estimator was developed to address the drawbacks of existing balance point estimators, as outlined in Section 2.2: it is predictive, considers impact, includes rotational inertia, allows ankle torques and 3D COM motion, does not require flat ground or heuristic parameters, and works for a biped at rest or in motion.

An example of an SFPE-based controller was also developed and compared to a COM-based controller in simulation. The addition of the SFPE to the control structure augments an existing optimal controller with both leaning and stepping strategies for disturbance rejection: SFPE-based feedback is used to generate a desired momentum for momentum-based leaning while the SFPE point is used as a control reference for stepping. The new estimator outperforms existing balance criteria by providing both recovery step location prediction and momentum objectives with smooth dynamics.

Unlike the existing 3D balance point estimators, such as the ICP, GFPE or 3D FPE [14,17,20], the SFPE does not assume instantaneous planar motion of the COM relative to a single point or plane. Through the predictive formulation of the SFPE, any desired "final" state of the system (after the next foot-fall) can be used to modify the resulting balance point. In this way, the SFPE can act both as a traditional balance point estimator and as a tool for the generation of dynamic gait and other whole body motions, which typically requires a significantly more complex strategy (such as those using Capturability [12,15]).

C3. Generalized 2D and 3D Gains

Generalized definitions of momentum gain for 2D and 3D systems, the definition of a novel spatial gain based on changes in centroidal momentum, and the formulation of a generalized gain equation which includes all existing gains as special cases.

In Chapter 5, simplified notation was developed which facilitates the definition of gains and other impulse based dynamic ratios. Then, building on Featherstone's initial definitions of momentum gains for 2-link planar inverted pendulums [27], the momentum gain formulation was extended to general 2D and 3D models and two different methods for calculating these general momentum gains were defined.

An additional gain was defined as part of the spatial method for calculating momentum gains, which was labeled the spatial gain of a system. Via this new gain definition, the velocity and momentum gains of a system were compared to its centroidal momentum

and the associated Centroidal Momentum Matrix (CMM) [33]. The spatial gain matrix was shown to be a direct link between centroidal momentum and the gains, along with providing an alternative method of calculating the CMM.

Bringing all of these novel elements together, a generalized gain formulation was defined which incorporates all existing gains as special cases and/or subsets. Specifically, this new formulation is able to reproduce all of the gains defined in [27] along with the novel gains defined in this thesis through the application of selection and scaling matrices. This new formulation was also used to show that the difference between velocity and momentum gains for is purely a function of scaling.

C4. Generalized Optimization Framework

A framework using generalized gains which enables the optimization of parameterized balancing mechanisms without the need to define or generate controllers or trajectories.

This optimization framework was defined in Chapter 5 along with its gain-based objective functions. Since the gains are purely functions of the physical properties and configuration of a system, a parameterized mechanism can be optimized for a given set of behaviors over a desired configuration space. This enables the optimization to be independent of the controller and/or trajectories used to achieve the given behaviors, invariant to gravitational or velocity product dynamics, and (when balancing on a point or line and using angular gains) independent of the contact angle.

By including the generalized gain formulation as part of the objective function, weighted matrix norms can be used to allow the optimization to incorporate domain knowledge such as torque limits and/or the relative importance of the various joints. As a demonstration of this framework and validation of the gains, several different sets of optimizations were carried out on simple planar mechanisms along with both 2D and 3D 5-link bipeds. The results of these optimizations, including a comparison to a cost-of-transport-based optimization, showed that the angular momentum gain generates comparable results without needing a controller and/or a trajectory to be defined.

Although there has been some work in the past on the optimization of mechanisms (as discussed in Section 2.4), there has not been an investigation of how to optimize a mechanism for better physical balance capabilities. There are only two contenders for measuring this type of performance: the dynamic COM manipulability of [23] and the gains defined in [27] then extended and generalized in Chapter 5. This framework enables the use of either of these metrics, and has been demonstrated using the novel gains that were introduced in this thesis to show the benefits of this approach.

7.2 Future Work

One main goal of the research in this thesis is to develop methods to facilitate the design and comparison of robust mechanisms which are inherently easy to balance and move, embedding as much capability into the mechanism itself as possible to enable the use of simpler controllers to achieve complex tasks.

One possible method to address this is the use of generalized gains in parallel with other suitable objectives for the design of a more complex system. This would benefit from an examination of the position gains of a system, defined briefly in [27] as the integral of velocity gain along a path in configuration space, and the analogous gains found by integrating the momentum gains along a similar configuration space path.

Another approach to maximizing the balancing capability of a mechanism would be to combine the benefits of the dynamic COM manipulability metric with those of the generalized gains defined in this thesis. The generalized gain formulation seems to be an instantaneous version of the dynamic COM manipulability of [23, 24], although more work is required to determine if this is in fact the case. Whether or not this is the case, an analysis of how the pairing of these instantaneous and continuous dynamic metrics could be used for balancing and gait would be an interesting and valuable contribution.

An important question which remains unanswered is how to effectively compare gains in a mechanism's configuration space. Since the configuration space of typical complex and/or 3D systems are not human-readable in most cases, this work could augment the existing tools available for designing these systems with an understanding of how effectively the mechanism could move its COM across a defined subset of the configuration space.

Finally, instead of using weighted matrix norms as is done in this work, other formulations of a scalar objective from a vector of gains could also be explored, such as minimizing the condition number of the gain matrix or maximizing its smallest singular value.

Once the physical form of the robot has been designed, the focus of future work will shift back to the controller and its associated metrics. In the case of this research, the chosen metrics are all dynamic balance point estimators such as the SFPE.

The benefits of extending the SFPE model to allow varying leg lengths, and different stance and swing leg lengths, should be analyzed. Both of these extensions would further relax the assumptions which currently restrict the SFPE model's COM motion to fixed-radius spherical motion about the anchor points. These changes are also included in a number of other balance point estimators which could be further examined, such as [16,119] which make use of analytical solutions to the energy equations.

Using a varying centroidal inertia in the SFPE model should be tested, inspired by the inertia shaping methods explored as part of the safe falling strategy in [83]. In combination with the average spatial velocity of the biped, the centroidal inertia could be used to modify the centroidal angular momentum and therefore could be used as part of a SFPE-based component of a balance strategy based on centroidal momentum.

Alternative pre-impact equations should also be explored to enable the SFPE to apply to a broader range of possible scenarios. For example, for a biped in flight the pre-impact equations could be generated by applying conservation of momentum and projectile motion of the COM to find the pre-impact state.

In [18], conservation of energy and momentum were used to determine the stable regions of the phase space of a planar compass gait biped in a predictive form, followed by a form of inversion to produce the real-time FPE derivation. Similarly, an inverted form of the SFPE should be developed which uses the current state and a desired footstep location to generate the required angular velocity for the SFPE model.

This desired angular velocity of the model could then be used to augment the control objectives of a biped. The integration of the SFPE and its inverse into a comprehensive disturbance compensation controller would enable a complementary combination of flexing, stepping, and leaning approaches within a single controller to respond to a wide variety of external disturbances.

References

[1] G. Pratt and M. Williamson, "Series elastic actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995, pp. 399–406.

[2] N. G. Tsagarakis, M. Laffranchi, B. Vanderborght, and D. G. Caldwell, "A compact soft actuator unit for small scale human friendly robots," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 4356–4362.

[3] M. Laffranchi, F. Cannella, and W. B. Sheffield, "Antagonistic and Series Elastic Actuators: a Comparative Analysis on the Energy Consumption," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5678–5684.

[4] B. J. Stephens and C. G. Atkeson, "Dynamic Balance Force Control for Compliant Humanoid robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1248–1255.

[5] B. Stephens, "Humanoid push recovery," *IEEE-RAS International Conference on Humanoid Robots*, pp. 589–595, 2007.

[6] S.-H. Hyon, "Compliant Terrain Adaptation for Biped Humanoids Without Measuring Ground Surface and Contact Forces," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 171–178, 2009.

[7] S.-H. Hyon, R. Osu, and Y. Otaka, "Integration of multi-level postural balancing on humanoid robots," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 1549–1556.

[8] S.-H. Lee and A. Goswami, "Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3157–3162.

[9] ——, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, vol. 33, no. 4, pp. 399–414, 4 2012.

[10] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, pp. 161–176, 2013.

[11] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[12] 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*, vol. 31, no. 10, pp. 1117–1133, 8 2012.

[13] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Compliant Locomotion Using Whole-Body Control and Divergent Component of Motion Tracking," in *IEEE International Conference on Robotics and Automation*, 2015.

[14] S.-k. Yun and A. Goswami, "Momentum-based reactive stepping controller on level and non-level ground for humanoid robot push recovery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3943–3950.

[15] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 7 2012.

[16] Z. Li, C. Zhou, H. Dallali, N. G. Tsagarakis, and D. G. Caldwell, "Comparison Study of Two Inverted Pendulum Models for Balance Recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 67–72.

[17] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[18] D. L. Wight, E. G. Kubica, and D. W. L. Wang, "Introduction of the Foot Placement Estimator: A Dynamic Measure of Balance for Bipedal Robotics," *Journal of Computational and Nonlinear Dynamics*, vol. 3, no. 1, 2008.

[19] P. v. Zutven and H. Nijmeijer, "Foot placement indicator for balance of planar bipeds with point feet," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[20] M. Millard, J. McPhee, and E. Kubica, "Foot Placement and Balance in 3D," *Journal of Computational and Nonlinear Dynamics*, vol. 7, no. 2, 2012.

[21] S. Cotton, P. Fraisse, and A. P. Murray, "On the Manipulability of the Center of Mass of Humanoid Robots: Application to Design," *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pp. 1259–1267, 2010.

[22] Y. Gu, C. S. Lee, and B. Yao, "Feasible center of mass dynamic manipulability of humanoid robots," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5082–5087.

[23] M. Azad, J. Babic, and M. Mistry, "Dynamic manipulability of the center of mass: A tool to study, analyse and measure physical ability of robots," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3484–3490.

[24] M. Azad, J. Babič, and M. Mistry, "Effects of the weighting matrix on dynamic manipulability of robots," *Autonomous Robots*, 2019.

[25] R. Featherstone, "Analysis and Design of Planar Self-Balancing Double-Pendulum Robots," in *RoManSy 19 Robot Design, Dynamics and Control*, V. Padios, P. Bidaud, and O. Khatib, Eds. Springer Vienna, 2013, pp. 259–266.

[26] ——, "Quantitative Measures of a Robots Ability to Balance," in *Robotics Science and Systems*, 2015.

[27] ——, "Quantitative measures of a robots physical ability to balance," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1681–1696, 2016.

[28] ——, *Rigid Body Dynamics Algorithms*. Springer US, 2008.

[29] R. Featherstone and D. E. Orin, "Dynamics," in *Springer Handbook of Robotics*, 2016, ch. 2, pp. 37–66.

[30] R. Featherstone, "A Beginner's Guide to 6-D Vectors (Part 1)," *IEEE Robotics & Automation Magazine*, pp. 83–94, 9 2010.

[31] ——, "A Beginner's Guide to 6-D Vectors (Part 2)," *IEEE Robotics & Automation Magazine*, pp. 88–99, 12 2010.

[32] F. L. Moro, G. Michael, A. Goswami, N. G. Tsagarakis, and D. G. Caldwell, "An attractor-based whole-body motion control (WBMC) system for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 42–49.

[33] D. E. Orin and A. Goswami, "Centroidal Momentum Matrix of a humanoid robot: Structure and properties," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 653–659.

[34] S.-H. H. Lee and A. Goswami, "Reaction Mass Pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4667–4672.

[35] M. Vukobratović and B. Borovac, "Zero-Moment Point Thirty Five Years of Its Life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[36] M. Popovic, A. Goswami, and H. Herr, "Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications," *The International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, 2005.

[37] M. Popovic, A. Hofmann, and H. Herr, "Angular momentum regulation during human walking: biomechanics and control," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 2405–2411.

[38] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 3785–3790.

[39] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and Control of Legged Robots," in *Springer Handbook of Robotics*, 2016, ch. 48, pp. 1203–1234.

[40] A. Goswami, "Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999.

[41] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1084–1091.

[42] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking." *Human movement science*, vol. 27, no. 1, pp. 112–25, 2 2008.

[43] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid Locomotion on Uneven Terrain Using the Time-Varying Divergent Component of Motion," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 266–272.

[44] J. Englsberger, G. Mesesan, and C. Ott, "Smooth trajectory generation and push-recovery based on Divergent Component of Motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4560–4567.

[45] J. Englsberger, G. Mesesan, C. Ott, and A. Albu-sch, "DCM-based gait generation for walking on moving support surfaces," in *IEEE-RAS International Conference on Humanoid Robots*, 2018, pp. 59–66.

[46] M. J. Coleman, "Dynamics and stability of a rimless spoked wheel: a simple 2D system with impacts," *Dynamical Systems*, vol. 25, no. 2, pp. 215–238, 2010.

[47] T. McGeer, "Passive Dynamic Walking," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 4 1990.

[48] M. J. Coleman, A. Chatterjee, and A. Ruina, "Motions of a rimless spoked wheel: a simple three-dimensional system with impacts," *Dynamics and Stability of Systems*, vol. 12, no. 3, pp. 139–159, 1 1997.

[49] A. Goswami, "Kinematic and dynamic analogies between planar biped robots and the reaction mass pendulum (RMP) model," in *IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 182–188.

[50] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1620–1626.

[51] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 1405–1411.

[52] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 239 – 246.

[53] Y. Zhao and L. Sentis, "A three dimensional foot placement planner for locomotion in very rough terrains," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 726–733.

[54] T. Komura, H. Leung, S. Kudoh, and J. Kuffner, "A feedback controller for biped humanoids that can counteract large perturbations during gait," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1989–1995.

[55] T. Komura, A. Nagano, H. Leung, and Y. Shinagawa, "Simulating Pathological Gait Using the Enhanced Linear Inverted Pendulum Model," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 9, pp. 1502–1513, 2005.

[56] A. K. Sanyal and A. Goswami, "Dynamics and Balance Control of the Reaction Mass Pendulum: A Three-Dimensional Multibody Pendulum With Variable Body Inertia," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, 2013.

[57] O. E. Ramos and K. Hauser, "Generalizations of the Capture Point to Nonlinear Center of Mass Paths and Uneven Terrain," in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 851–858.

[58] J. E. Pratt and R. Tedrake, "Velocity-based stability margins for fast bipedal walking," *Fast Motions in Biomechanics and Robotics*, vol. 340, pp. 299–324, 2006.

[59] M. Millard, D. Wight, J. McPhee, E. Kubica, and D. Wang, "Human foot placement and balance in the sagittal plane." *Journal of biomechanical engineering*, vol. 131, no. 12, p. 121001, 2009.

[60] S. Choudhury and D. Kulić, "Gait Generation via the Foot Placement Estimator for 3D Bipedal Robots," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 5 2013, pp. 5689–5695.

[61] M.-A. N. Mahani, S. Jafari, and R. Rahmatkhah, "Novel humanoid push recovery using knee joint," in *IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 446–451.

[62] N. G. Tsagarakis, S. Morfey, G. Medrano Cerda, Z. Li, and D. G. Caldwell, "COMpliant huMANoid COMAN: Optimal joint stiffness tuning for modal frequency control," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 673–678.

[63] F. Daerden and D. Lefeber, "Pneumatic artificial muscles: actuators for robotics and automation," *European Journal of Mechanical and Environmental Engineering*, vol. 47, no. 1, p. 1121, 2002.

[64] P. Kryczka, K. Hashimoto, A. Takanishi, H. O. Lim, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, "Walking despite the passive compliance: techniques for using conventional pattern generators to control intrinsically compliant humanoid robots," in *International Conference on Climbing and Walking Robots*, 2013, pp. 487–494.

[65] Z. Li, N. G. Tsagarakis, and D. G. Caldwell, "A passivity based admittance control for stabilizing the compliant humanoid COMAN," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 43–49.

[66] Z. Li, B. Vanderborght, N. G. Tsagarakis, L. Colasanto, and D. G. Caldwell, "Stabilization for the compliant humanoid robot COMAN exploiting intrinsic and controlled compliance," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 2000–2006.

[67] Z. Li, N. G. Tsagarakis, and D. G. Caldwell, "Walking trajectory generation for humanoid robots with compliant joints: Experimentation with COMAN humanoid," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 836–841.

[68] Y. Yoshida, K. Takeuchi, Y. Miyamoto, D. Sato, and D. Nenchev, "Postural balance strategies in response to disturbances in the frontal plane and their implementation with a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 692–704, 2014.

[69] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 1644–1650.

[70] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 1, 7 2009.

[71] A. Hofmann, M. Popovic, and H. Herr, "Exploiting angular momentum to enhance bipedal center-of-mass control," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 4423–4429.

[72] P. M. Wensing and D. E. Orin, "Generation of Dynamic Humanoid Behaviors Through Task-Space Control with Conic Optimization," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 3103–3109.

[73] ——, "Improved Computation of the Humanoid Centroidal Dynamics and Application for Whole-Body Control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550039, 2016.

[74] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 295–302.

[75] N. Perrin, N. Tsagarakis, and D. G. Caldwell, "Compliant Attitude Control and Stepping Strategy for Balance Recovery with the Humanoid COMAN," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4145–4151.

[76] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J. Laumond, "Combining suppression of the disturbance and reactive stepping for recovering balance," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3150–3156, 2010.

[77] B. Stephens, "Integral control of humanoid balance," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4020–4027, 2007.

[78] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, "Learning full body push recovery control for small humanoid robots," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2047–2052.

[79] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 190–195.

[80] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Footstep Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[81] Z. Aftab, T. Robert, and P. B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 159–164.

[82] Y. Kanamiya, S. Ota, and D. Sato, "Ankle and hip balance control strategies with transitions," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3446–3451, 2010.

[83] S.-k. Yun, A. Goswami, and Y. Sakagami, "Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 781–787.

[84] S.-k. Yun and A. Goswami, "Tripod Fall: Concept and Experiments of a Novel Approach to Humanoid Robot Fall Damage Reduction," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 2799–2805.

[85] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time Model Predictive Control of humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 292–299.

[86] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 3346–3351.

[87] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Robust Dynamic Walking Using Online Foot Step Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 5373–5378.

[88] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[89] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Soueres, "A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017.

[90] M. Shafiee-Ashtiani, A. Yousefi-Koma, and M. Shariat-Panahi, "Robust bipedal locomotion control based on model predictive control and divergent component of motion," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3505–3510.

[91] H. M. Joe and J. H. Oh, "Balance recovery through model predictive control based on capture point dynamics for biped walking robot," *Robotics and Autonomous Systems*, vol. 105, pp. 1–10, 2018.

[92] C. Paul and J. C. Bongard, "The Road Less Travelled : Morphology in the Optimization of Biped Robot Locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 226 – 232.

[93] K. Endo, T. Maeno, and H. Kitano, "Co-evolution of Morphology and Walking Pattern of Biped Humanoid Robot using Evolutionary Computation - Evolutionary Designing Method and its Evaluation -," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 340–345.

[94] M. Haberland and S. Kim, "On extracting design principles from biology: I. Method-General answers to high-level design questions for bioinspired robots," *Bioinspiration & Biomimetics*, vol. 10, 2015.

[95] ——, "On extracting design principles from biology: II. Case studythe effect of knee direction on bipedal robot running efficiency," *Bioinspiration & Biomimetics*, vol. 10, no. 1, 2015.

[96] U. J. Römer, C. Kuhs, M. J. Krause, and A. Fidlin, "Simultaneous optimization of gait and design parameters for bipedal robots," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1374–1381.

[97] G. Saurel, J. Carpentier, and J.-P. Laumond, "A Simulation Framework for Simultaneous Design and Control of Passive Walkers," in *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, 2016, pp. 104–110.

[98] G. Buondonno, J. Carpentier, G. Saurel, N. Mansard, A. De Luca, and J.-P. Laumond, "Actuator Design of Compliant Walkers via Optimal Control," *Rapport LAAS*, no. 17049, 2017.

[99] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[100] F. Romano, G. Nava, M. Azad, J. Camernik, S. Dafarra, O. Dermy, C. Latella, M. Lazzaroni, R. Lober, M. Lorenzini, D. Pucci, O. Sigaud, S. Traversaro, J. Babic, S. Ivaldi, M. Mistry, V. Padois, and F. Nori, "The CoDyCo Project achievements and

beyond: Towards Human Aware Whole-body Controllers for Physical Human Robot Interaction," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 516–523, 2018.

[101] M. Azad, "Balancing and Hopping Motion Control Algorithms for an Under-actuated Robot," Ph.D. dissertation, Australian National University, 2014.

[102] M. Azad and R. Featherstone, "Angular momentum based balance controller for an under-actuated planar robot," *Autonomous Robots*, vol. 40, no. 1, pp. 93–107, 2016.

[103] R. Featherstone, "A simple model of balancing in the plane and a simple preview balance controller," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1489–1507, 2017.

[104] B. J. DeHart and D. Kulić, "Push recovery and online gait generation for 3D bipeds with the foot placement estimator," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 1937–1942.

[105] S. Choudhury, "Design and Gait Synthesis for a 3D Lower Body Humanoid," Ph.D. dissertation, University of Waterloo, 2012.

[106] S. Choudhury, D. Wight, and D. Kulić, "Rapid Prototyping Toolchain for Humanoid Robotics Applications," in *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 817–822.

[107] B. J. DeHart, R. Gorbet, and D. Kulić, "Spherical foot placement estimator for humanoid balance control and recovery," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1747–1754.

[108] M. S. Orendurff, A. D. Segal, G. K. Klute, J. S. Berge, E. S. Rohr, and N. J. Kadel, "The effect of walking speed on center of mass displacement." *Journal of rehabilitation research and development*, vol. 41(6A), pp. 829–834, 2004.

[109] Z. Li, C. Zhou, J. Castano, X. Wang, F. Negrello, N. G. Tsagarakis, and D. G. Caldwell, "Fall Prediction of Legged Robots Based on Energy State and Its Implication of Balance Augmentation: A Study on the Humanoid," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5094–5100.

[110] Boston Dynamics, "Boston Dynamics' Atlas Robot," 2017. [Online]. Available: http://www.bostondynamics.com/atlas

[111] R. Tedrake and the Drake Development Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: http://drake.mit.edu

[112] S. Kuindersma, F. Permenter, and R. Tedrake, "An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 2589–2594.

[113] B. J. DeHart and D. Kulić, "Quantifying balance capabilities using momentum gain," in *IEEE-RAS International Conference on Humanoid Robots*, 2017, pp. 561–568.

[114] ——, "Legged mechanism design with momentum gains," in *IEEE-RAS International Conference on Humanoid Robots*, 2017, pp. 593–598.

[115] B. J. DeHart, R. Gorbet, and D. Kulić, "Quantifying balance capabilities for optimal mechanism design," *IEEE Transactions on Robotics (submitted)*, 2019.

[116] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," *Nature*, vol. 439, no. 7072, pp. 72–75, 2006.

[117] R. Featherstone, "Spatial_v2: Spatial Vector and Rigid-Body Dynamics Software," 2015. [Online]. Available: http://royfeatherstone.org/spatial/v2/index.html

[118] K. M. Digumarti, C. Gehring, S. Coros, J. Hwangbo, and R. Siegwart, "Concurrent Optimization of Mechanical Design and Locomotion Control of a Legged Robot," in *Mobile Service Robotics*. World Scientific, 2014, pp. 315–323.

[119] B.-K. Cho, J.-H. Kim, and J.-H. Oh, "Balancing Strategy using the Principle of Energy Conservation for a Hopping Humanoid Robot," *International Journal of Humanoid Robotics*, vol. 10, no. 3, 2013.

[120] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *The International Journal of Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.

[121] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "A Spatial Operator Algebra for Manipulator Modeling and Control," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 371–381, 8 1991.

[122] R. Featherstone, "Plücker basis vectors," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, pp. 1892–1897, 2006.

Appendices

Appendix A

Spatial Vector Algebra: Details

Spatial vector algebra was introduced by Featherstone in 1983 [120], while the closely related spatial operator algebra was introduced in 1991 [121]. Both approaches make use of 6D vectors and tensors, including a family of (usually sparse) matrices, to enable the efficient representation and calculation of the various physical properties of a multibody system. Spatial vectors are very similar to other 6D formulations of multibody dynamics, such as screw theory, the dual Lie algebras se(3) and se*(3), and real-number motors [28] .

The application of these 6D methods of representation and calculation, and in particular spatial vectors, have been shown to reduce the apparent complexity in modelling the dynamics of humanoids [10]. An introduction to the concepts of spatial vector algebra follows, based heavily on Part 1 of an introductory tutorial on the topic [30]. Part 2 of the tutorial serves as an introduction to how to write algorithms using spatial vectors [31].

Plücker Coordinates

Spatial vectors are defined in two vector spaces: motion vectors in $\mathbb{M}^6$ and force vectors in $\mathbb{F}^6$, where the superscript denotes the dimension of each space. The most common motion vectors are the spatial velocity $\vec{\boldsymbol{v}}$ and spatial acceleration $\vec{\boldsymbol{a}}$, while the most common force vectors are the spatial force $\vec{\boldsymbol{f}}$ and spatial momentum $\vec{\boldsymbol{h}}$. To facilitate the generalized definitions below, motion vectors are labelled $\vec{\boldsymbol{m}}$, force vectors are labelled $\vec{\boldsymbol{f}}$, and spatial vectors in either space are labelled $\vec{\boldsymbol{s}}$. Although $\vec{\boldsymbol{f}}$ is used for both a general force vector and specifically for spatial force, the equations in this section apply to all properties which are considered force vectors, such as momentum, unless otherwise noted. A scalar product

(dot product) is defined between the two vector spaces, which maps one vector from each space together to produce a single, real value: $\vec{\boldsymbol{m}} \cdot \vec{\boldsymbol{f}} = \vec{\boldsymbol{f}} \cdot \vec{\boldsymbol{m}} \in \mathbb{R}$.

In classic 3D vector notation, three independent unit vectors in Cartesian coordinates (i.e., $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$) are used as basis vectors to represent the 3D vectors, such as $\vec{a} = a_x\hat{\imath} + a_y\hat{\jmath} + a_z\hat{k}$, using 3D coordinate vectors: $\underline{\vec{a}} = [a_x,\ a_y,\ a_z]^T \in \mathbb{R}^3$. Similarly, a set of 12 independent unit vectors in Plücker coordinates [122] are used to represent spatial vectors. These are split into two sets of six basis vectors each: $\mathcal{D} = \{\hat{\boldsymbol{d}}_1,\ \hat{\boldsymbol{d}}_2,\ \ldots,\ \hat{\boldsymbol{d}}_6\} \subset \mathbb{M}^6$ for motion vectors and $\mathcal{E} = \{\hat{\boldsymbol{e}}_1,\ \hat{\boldsymbol{e}}_2,\ \ldots,\ \hat{\boldsymbol{e}}_6\} \subset \mathbb{F}^6$ for force vectors. This enables the use of spatial coordinate vectors $\underline{\vec{\boldsymbol{m}}} = [m_1,\ m_2,\ \ldots,\ m_6]^T \in \mathbb{R}^6$ and $\underline{\vec{\boldsymbol{f}}} = [f_1,\ f_2,\ \ldots,\ f_6]^T \in \mathbb{R}^6$ to represent motion vectors ($\vec{\boldsymbol{m}} = \sum_{i=1}^{6} m_i\hat{\boldsymbol{d}}_i$) and force vectors ($\vec{\boldsymbol{f}} = \sum_{i=1}^{6} f_i\hat{\boldsymbol{e}}_i$).

The most useful set of basis vectors comprises a dual basis, where $\hat{\boldsymbol{d}}_i \cdot \hat{\boldsymbol{e}}_i = 1$ and $\hat{\boldsymbol{d}}_i \cdot \hat{\boldsymbol{e}}_j = 0$ for $i \neq j$. Using a dual basis for the vector spaces allows the dot product between the motion and force spaces to be calculated in the same fashion as classic 3D vectors ($\vec{\boldsymbol{m}} \cdot \vec{\boldsymbol{f}} = \underline{\vec{\boldsymbol{m}}}^T \underline{\vec{\boldsymbol{f}}} = \sum_{i=1}^{6} m_i f_i$) and makes the product invariant with respect to the chosen coordinate frame. Since a given set of Plücker coordinates are defined entirely by the position and orientation of a single Cartesian reference frame, the specific dual basis used in this work (for an arbitrary Cartesian reference frame O) is:

$$\mathcal{D} = \{\hat{\boldsymbol{d}}_{Ox},\ \hat{\boldsymbol{d}}_{Oy},\ \hat{\boldsymbol{d}}_{Oz},\ \hat{\boldsymbol{d}}_x,\ \hat{\boldsymbol{d}}_y,\ \hat{\boldsymbol{d}}_z\} \subset \mathbb{M}^6 \tag{A.1}$$

$$\mathcal{E} = \{\hat{\boldsymbol{e}}_x,\ \hat{\boldsymbol{e}}_y,\ \hat{\boldsymbol{e}}_z,\ \hat{\boldsymbol{e}}_{Ox},\ \hat{\boldsymbol{e}}_{Oy},\ \hat{\boldsymbol{e}}_{Oz}\} \subset \mathbb{F}^6 \tag{A.2}$$

In these equations, the following conventions are used (where a replaces x, y, or z): $\hat{\boldsymbol{d}}_{Oa}$ is a unit angular motion around the Oa axis, $\hat{\boldsymbol{d}}_a$ is a unit linear motion in the a direction, $\hat{\boldsymbol{e}}_a$ is a unit couple in the a direction, and $\hat{\boldsymbol{e}}_{Oa}$ is a unit linear force along the Oa axis. Spatial coordinate vectors are given a presuperscript to specify the Cartesian reference frame used to define their Plücker coordinates, in this case O (e.g., ${}^{O}\underline{\vec{\boldsymbol{m}}}$, ${}^{O}\underline{\vec{\boldsymbol{f}}}$).

In cases where the spatial vector being represented has a subscripted label (e.g., $\vec{\boldsymbol{s}}_A$), if the label is the same as the coordinate vector's Plücker coordinate label, then typically only the subscript is used: ${}^{A}\underline{\vec{\boldsymbol{s}}}_A \equiv \underline{\vec{\boldsymbol{s}}}_A$. Note that if the coordinate vector is transformed to another coordinate frame (as discussed below), both labels are still needed.

Due to the dual coordinate system, spatial motion and force vectors use different rules for transforming between reference frames: if $\tilde{\boldsymbol{X}}$ is a transformation matrix for motion vectors, then the corresponding matrix for force vectors is labelled $\tilde{\boldsymbol{X}}^*$. These two matrices are directly related through the equation $\tilde{\boldsymbol{X}}^* = (\tilde{\boldsymbol{X}}^T)^{-1} = (\tilde{\boldsymbol{X}}^{-1})^T = \tilde{\boldsymbol{X}}^{-T}$, which guarantees that the spatial dot product is invariant to coordinate transformations:

$$(\tilde{\boldsymbol{X}}^* \underline{\vec{\boldsymbol{f}}})^T (\tilde{\boldsymbol{X}} \underline{\vec{\boldsymbol{m}}}) = \underline{\vec{\boldsymbol{f}}}^T \tilde{\boldsymbol{X}}^{-1} \tilde{\boldsymbol{X}} \underline{\vec{\boldsymbol{m}}} = \underline{\vec{\boldsymbol{f}}}^T \underline{\vec{\boldsymbol{m}}} \tag{A.3}$$

Spatial Vectors

Using 3D kinematics, the velocity of a rigid body can be defined using the linear velocity $\vec{v}_P$ of a point P on the body, and the body's angular velocity $\vec{\omega}$. The body's spatial velocity $\vec{\boldsymbol{v}}$ can be found using these 3D vectors, by converting between their respective coordinate vectors. Since angular velocity is a property of the rigid body, it can be moved from P to O and its coordinate vector $\underline{\vec{\omega}}$ can be used directly in the spatial coordinate vector ${}^{O}\underline{\vec{\boldsymbol{v}}}$. By moving $\vec{\omega}$, the linear velocity must be augmented to compensate, using $\vec{r}_{OP}$ (a 3D vector from O to P). The coordinate vectors $\underline{\vec{\omega}}$ and $\underline{\vec{v}}_O$ can then be used to construct ${}^{O}\underline{\vec{\boldsymbol{v}}} \in \mathbb{R}^6$ which, when combined with the basis $\mathcal{D}$, leads directly to the value of $\vec{\boldsymbol{v}} \in \mathbb{M}^6$

$$\vec{v}_O = \vec{v}_P + \vec{r}_{OP} \times \vec{\omega} \tag{A.4}$$

$${}^{O}\underline{\vec{\boldsymbol{v}}} = \begin{bmatrix} \underline{\vec{\omega}}^T & \underline{\vec{v}}_O^T \end{bmatrix}^T = \begin{bmatrix} \omega_x & \omega_y & \omega_z & v_{Ox} & v_{Oy} & v_{Oz} \end{bmatrix}^T \tag{A.5}$$

$$\vec{\boldsymbol{v}} = \omega_x \hat{\boldsymbol{d}}_{Ox} + \omega_y \hat{\boldsymbol{d}}_{Oy} + \omega_z \hat{\boldsymbol{d}}_{Oz} + v_{Ox} \hat{\boldsymbol{d}}_x + v_{Oy} \hat{\boldsymbol{d}}_x + v_{Oz} \hat{\boldsymbol{d}}_x \tag{A.6}$$

Similarly, in classical dynamics all forces acting on a rigid body can be defined using a linear force $\vec{f}$ acting through a point P on the body and a couple $\vec{n}_P$. Much like the spatial velocity calculations above, the spatial force $\vec{\boldsymbol{f}} \in \mathbb{F}^6$ can be found by recalling that $\vec{f}$ can be moved to any point by adding an appropriate couple:

$$\vec{n}_O = \vec{n}_P + \vec{r}_{OP} \times \vec{f} \tag{A.7}$$

$${}^{O}\underline{\vec{\boldsymbol{f}}} = \begin{bmatrix} \underline{\vec{n}}_O^T & \underline{\vec{f}}^T \end{bmatrix}^T = \begin{bmatrix} n_{Ox} & n_{Oy} & n_{Oz} & f_x & f_y & f_z \end{bmatrix}^T \tag{A.8}$$

$$\vec{\boldsymbol{f}} = n_{Ox} \hat{\boldsymbol{e}}_x + n_{Oy} \hat{\boldsymbol{e}}_y + n_{Oz} \hat{\boldsymbol{e}}_z + f_x \hat{\boldsymbol{e}}_{Ox} + f_y \hat{\boldsymbol{e}}_{Oy} + f_z \hat{\boldsymbol{e}}_{Oz} \tag{A.9}$$

Spatial Transforms

If we define two Cartesian frames with associated Plücker coordinate systems, labelled A and B, then we can also define the coordinate vectors ${}^{A}\underline{\vec{\boldsymbol{m}}}$, ${}^{B}\underline{\vec{\boldsymbol{m}}}$, ${}^{A}\underline{\vec{\boldsymbol{f}}}$, and ${}^{B}\underline{\vec{\boldsymbol{f}}}$, which represent the spatial vectors $\vec{\boldsymbol{m}}$ and $\vec{\boldsymbol{f}}$ in these coordinates. Transformations between the two frames can then be defined as ${}^{A}\underline{\vec{\boldsymbol{m}}} = {}^{A}\tilde{\boldsymbol{X}}_B \, {}^{B}\underline{\vec{\boldsymbol{m}}}$ and ${}^{A}\underline{\vec{\boldsymbol{f}}} = {}^{A}\tilde{\boldsymbol{X}}_B^* \, {}^{B}\underline{\vec{\boldsymbol{f}}}$, where ${}^{A}\tilde{\boldsymbol{X}}_B$ and ${}^{A}\tilde{\boldsymbol{X}}_B^*$ are the coordinate transformation matrices from B to A coordinates for motion and force vectors, respectively. Note the presuperscript denotes the new coordinate frame.

In classical methods, a transformation from B to A would make use of a 4x4 homogeneous transformation matrix which contains a 3×3 rotation matrix $\tilde{R}$ specifying the

rotation of A with respect to B, and a vector $\vec{r}$ specifying the position of A with respect to B, both in B coordinates. Using the spatial vector examples above, it is easy to show that the spatial transformation matrices from the frame at B to the one at A can be determined entirely using $\tilde{R}$ and the skew symmetric matrix $\tilde{r}$ (where $\tilde{r}\underline{\vec{a}} = \vec{r} \times \vec{a}$, as discussed later):

$$ {}^{A}\tilde{\boldsymbol{X}}_{B} = \begin{bmatrix} \tilde{R} & \tilde{0} \\ \tilde{0} & \tilde{R} \end{bmatrix} \begin{bmatrix} \tilde{1} & \tilde{0} \\ \tilde{r}^{T} & \tilde{1} \end{bmatrix} = \begin{bmatrix} \tilde{R} & \tilde{0} \\ \tilde{R}\tilde{r}^{T} & \tilde{R} \end{bmatrix} \tag{A.10} $$

$$ {}^{A}\tilde{\boldsymbol{X}}^{*}_{B} = \begin{bmatrix} \tilde{R} & \tilde{0} \\ \tilde{0} & \tilde{R} \end{bmatrix} \begin{bmatrix} \tilde{1} & \tilde{r}^{T} \\ \tilde{0} & \tilde{1} \end{bmatrix} = \begin{bmatrix} \tilde{R} & \tilde{R}\tilde{r}^{T} \\ \tilde{0} & \tilde{R} \end{bmatrix} \tag{A.11} $$

where $\tilde{1}$ and $\tilde{0}$ are 3×3 identity and zero matrices, respectively, and the matrix $\tilde{r}$ is the 3×3 skew symmetric matrix defined by the coordinates of $\vec{r}$ (where $\underline{\vec{r}} = [r_x,\ r_y,\ r_z]^T$):

$$ \tilde{r} = Skew(\vec{r}) = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{A.12} $$

Based on these equations, and the properties of rotation and skew symmetric matrices, the following conversion between force and motion transformation matrices holds:

$$ {}^{B}\tilde{\boldsymbol{X}}^{*}_{A} = \left({}^{B}\tilde{\boldsymbol{X}}_{A}\right)^{-T} = \left({}^{A}\tilde{\boldsymbol{X}}_{B}\right)^{T} = \left({}^{A}\tilde{\boldsymbol{X}}^{*}_{B}\right)^{-1} \tag{A.13} $$

Differentiation

The derivative $\dot{\vec{\boldsymbol{s}}}(t)$ of a generalized spatial vector $\vec{\boldsymbol{s}}(t)$ is also a spatial vector in the same space, given by the usual differentiation equation for vectors

$$ \dot{\vec{\boldsymbol{s}}}(t) = \frac{d}{dt}\vec{\boldsymbol{s}}(t) = \lim_{\delta t \to 0} \frac{\vec{\boldsymbol{s}}(t+\delta t) - \vec{\boldsymbol{s}}(t)}{\delta t} \tag{A.14} $$

This standard formulation leads to two key equations for finding the derivatives of different instances of spatial vectors. First, the spatial coordinate vector of $\dot{\vec{\boldsymbol{s}}}$ in a fixed frame O is simply the component-wise derivative of ${}^{O}\underline{\vec{\boldsymbol{s}}}$ (in O coordinates):

$$ {}^{O}\dot{\underline{\vec{\boldsymbol{s}}}} = \frac{d}{dt}({}^{O}\underline{\vec{\boldsymbol{s}}}) = \frac{d}{dt}\begin{bmatrix} {}^{O}s_1 & {}^{O}s_2 & \dots & {}^{O}s_6 \end{bmatrix}^{T} = \begin{bmatrix} {}^{O}\dot{s}_1 & {}^{O}\dot{s}_2 & \dots & {}^{O}\dot{s}_6 \end{bmatrix}^{T} \tag{A.15} $$

Second, the derivate $\dot{\vec{\boldsymbol{s}}}$ of a spatial vector $\vec{\boldsymbol{s}}$ whose coordinate vector is constant in a moving frame, where the frame has a velocity of $\vec{\boldsymbol{v}}$, is given by

$$\dot{\vec{\boldsymbol{s}}} = \begin{cases} \vec{\boldsymbol{v}} \times \vec{\boldsymbol{s}} & \text{if } \vec{\boldsymbol{s}} \in \mathbb{M}^6 \\ \vec{\boldsymbol{v}} \times^* \vec{\boldsymbol{s}} & \text{if } \vec{\boldsymbol{s}} \in \mathbb{F}^6 \end{cases} \tag{A.16}$$

As seen above, a pair of spatial cross products are required to perform differentiation due to the dual basis: one between motion vectors, which uses the standard operator notation ($\times$), and one between a motion vector and a force vector ($\times^*$). As shown, the first spatial vector in these operations is always a velocity (e.g., $\vec{\boldsymbol{v}} \times \vec{\boldsymbol{m}}$ or $\vec{\boldsymbol{v}} \times^* \vec{\boldsymbol{f}}$).

To gain some insight into these different spatial cross products, the standard cross product between 3D vectors is examined. Given two arbitrary 3D vectors, $\vec{a}$ and $\vec{b}$, and their coordinate vectors in Cartesian space, $\underline{\vec{a}} = [a_x,\ a_y,\ a_z]^T$ and $\underline{\vec{b}} = [b_x,\ b_y,\ b_z]^T$, the cross product of the vectors can be written as a matrix multiplication:

$$\vec{a} \times \vec{b} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \tilde{a}\vec{b} \tag{A.17}$$

A similar method exists for defining 6×6 matrices to replace both forms of spatial cross products, using the coordinate vectors ${}^O\underline{\vec{\boldsymbol{m}}}$, ${}^O\underline{\vec{\boldsymbol{f}}}$, and $\underline{\vec{\boldsymbol{v}}}_O$ (the velocity of O in O coordinates) where O is the moving frame. Each 6×6 matrix contains three 3×3 skew symmetric matrices (two of them identical) and $\tilde{0}$, a 3×3 zero matrix:

$$\vec{\boldsymbol{v}}_O \times \vec{\boldsymbol{m}} = \begin{bmatrix} \vec{\omega} \\ \vec{v}_O \end{bmatrix} \times \begin{bmatrix} \vec{m} \\ \vec{m}_O \end{bmatrix} = \begin{bmatrix} \tilde{\omega}\vec{m} \\ \tilde{\omega}\vec{m}_O + \tilde{v}_O\vec{m} \end{bmatrix} = \begin{bmatrix} \tilde{\omega} & \tilde{0} \\ \tilde{v}_O & \tilde{\omega} \end{bmatrix} \begin{bmatrix} \vec{m} \\ \vec{m}_O \end{bmatrix} = \tilde{\boldsymbol{v}}_O\vec{\boldsymbol{m}} \tag{A.18}$$

$$\vec{\boldsymbol{v}}_O \times^* \vec{\boldsymbol{f}} = \begin{bmatrix} \vec{\omega} \\ \vec{v}_O \end{bmatrix} \times^* \begin{bmatrix} \vec{f}_O \\ \vec{f} \end{bmatrix} = \begin{bmatrix} \tilde{\omega}\vec{f}_O + \tilde{v}_O\vec{f} \\ \tilde{\omega}\vec{f} \end{bmatrix} = \begin{bmatrix} \tilde{\omega} & \tilde{v}_O \\ \tilde{0} & \tilde{\omega} \end{bmatrix} \begin{bmatrix} \vec{f}_O \\ \vec{f} \end{bmatrix} = \tilde{\boldsymbol{v}}_O^*\vec{\boldsymbol{f}} \tag{A.19}$$

where the matrix $\tilde{\boldsymbol{v}}^*$ can easily be shown to be equal to $-\tilde{\boldsymbol{v}}^T$ using the inherent properties of skew symmetric matrices.

Finally, the spatial coordinate vector ${}^O\dot{\underline{\vec{\boldsymbol{s}}}}$ of the derivative $\dot{\vec{\boldsymbol{s}}}$ of a general spatial vector $\vec{\boldsymbol{s}}$, in a moving frame O with velocity $\vec{\boldsymbol{v}}_O$, is found by adding together the first two types of derivatives, one for $\vec{\boldsymbol{s}}$ changing relative to O and one for the movement of O itself:

$${}^O\dot{\underline{\vec{\boldsymbol{s}}}} = \begin{bmatrix} {}^O\dot{s}_1 & {}^O\dot{s}_2 & \dots & {}^O\dot{s}_6 \end{bmatrix}^T + \begin{cases} \tilde{\boldsymbol{v}}_O\vec{\boldsymbol{s}} & \text{if } \vec{\boldsymbol{s}} \in \mathbb{M}^6 \\ \tilde{\boldsymbol{v}}_O^*\vec{\boldsymbol{s}} & \text{if } \vec{\boldsymbol{s}} \in \mathbb{F}^6 \end{cases} \tag{A.20}$$

Acceleration

The spatial acceleration of a body is found by simply differentiating its spatial velocity: $\vec{\boldsymbol{a}} = \dot{\vec{\boldsymbol{v}}}$. For a simple example of this, consider two bodies attached by a joint along an axis $\vec{\boldsymbol{s}} \in \mathbb{M}^6$ (fixed in the second body), which constrains their movement such that $\vec{\boldsymbol{v}}_2 = \vec{\boldsymbol{v}}_1 + \vec{\boldsymbol{s}}\dot{q}$, where $\dot{q}$ is the joint velocity. The acceleration of the second body is found by simply differentiating the spatial velocity equation: $\vec{\boldsymbol{a}}_2 = \vec{\boldsymbol{a}}_1 + \dot{\vec{\boldsymbol{s}}}\dot{q} + \vec{\boldsymbol{s}}\ddot{q}$.

The second term in this equation can then be simplified, using the above methods for differentiation of spatial vectors which are fixed in a given frame, the equation for $\vec{\boldsymbol{v}}_2$, and a few spatial cross product identities (i.e., $\tilde{\boldsymbol{v}}(\vec{\boldsymbol{s}}a) = (\tilde{\boldsymbol{v}}\vec{\boldsymbol{s}})a$, $\tilde{\boldsymbol{v}}\vec{\boldsymbol{v}} = 0$, and $\tilde{\boldsymbol{v}}_i\vec{\boldsymbol{v}}_j = -\tilde{\boldsymbol{v}}_j\vec{\boldsymbol{v}}_i$):

$$\dot{\vec{\boldsymbol{s}}}\dot{q} = (\tilde{\boldsymbol{v}}_2\vec{\boldsymbol{s}})\dot{q} = \tilde{\boldsymbol{v}}_2(\vec{\boldsymbol{s}}\dot{q}) = \tilde{\boldsymbol{v}}_2(\vec{\boldsymbol{v}}_2 - \vec{\boldsymbol{v}}_1) = \tilde{\boldsymbol{v}}_2(-\vec{\boldsymbol{v}}_1) = \tilde{\boldsymbol{v}}_1\vec{\boldsymbol{v}}_2 \tag{A.21}$$

Momentum and Inertia

The spatial momentum $\vec{\boldsymbol{h}} \in \mathbb{F}^6$ of a rigid body is given by $\vec{\boldsymbol{h}} = \tilde{\boldsymbol{I}}\vec{\boldsymbol{v}}$, where $\tilde{\boldsymbol{I}}$ is the body's spatial inertia. The spatial inertia also provides a simple formulation of the kinetic energy T of a rigid body: $T = \frac{1}{2}\vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{h}} = \frac{1}{2}\vec{\boldsymbol{v}} \cdot \tilde{\boldsymbol{I}}\vec{\boldsymbol{v}}$. Since momentum is conservative, the total momentum of a system of bodies is simply the sum of the momenta of each of the bodies. If the bodies are rigidly connected, their spatial velocities will all be equal, which directly leads to their total spatial inertia being equal to the sum of their individual inertias.

The spatial inertia matrix of a body in any dual coordinate system is a 6×6 symmetric, positive definite matrix (positive semi-definite for massless bodies) which combines three rigid body inertial properties: its mass m, a 3D vector $\vec{c}$ from O to the body's Center of Mass (COM) C, and the 3×3 symmetric matrix $\tilde{I}_C$: the rotational inertia of the body around C. Therefore, the inertia matrix for any body in the O frame is given by

$${}^{O}\underline{\tilde{\boldsymbol{I}}} = \begin{bmatrix} \tilde{I}_C + m\tilde{c}\tilde{c}^T & m\tilde{c} \\ m\tilde{c}^T & m\tilde{1} \end{bmatrix} = \begin{bmatrix} \tilde{I}_O & m\tilde{c} \\ m\tilde{c}^T & m\tilde{1} \end{bmatrix} \tag{A.22}$$

where $\tilde{I}_O$ is the rotational inertia of the body around O and $\tilde{1}$ is a 3×3 identity matrix.

It should be noted that although a general spatial inertia matrix potentially has 21 independent parameters, the spatial inertia matrix for a single rigid body is a function of only 10 parameters. The full 21 parameters are only needed to define the spatial inertia of a system of rigid bodies which are not rigidly connected [30].

The construction of the ${}^{O}\underline{\tilde{\boldsymbol{I}}}$ matrix can be demonstrated easily, using equations from classic momentum calculations. In classic 3D dynamics, the linear momentum $\vec{l}$ of a rigid

body is found by multiplying the mass of the body by the linear velocity of its COM: $\vec{l} = m\vec{v}_C$. The angular momentum $\vec{k}_C$ of a rigid body around its COM is given by $\vec{k}_C = \tilde{I}_C\vec{\omega}$, which leads to a simple method for defining the spatial momentum and inertia at C:

$$ {}^{C}\underline{\vec{\boldsymbol{h}}} = \begin{bmatrix} \underline{\vec{k}}_C \\ \underline{\vec{l}} \end{bmatrix} = \begin{bmatrix} \tilde{I}_C\underline{\vec{\omega}} \\ m\underline{\vec{v}}_C \end{bmatrix} = \begin{bmatrix} \tilde{I}_C & \tilde{0} \\ \tilde{0} & m\tilde{1} \end{bmatrix} \begin{bmatrix} \underline{\vec{\omega}} \\ \underline{\vec{v}}_C \end{bmatrix} = {}^{C}\underline{\tilde{\boldsymbol{I}}}\,{}^{C}\underline{\vec{\boldsymbol{v}}} \tag{A.23} $$

To find ${}^{O}\underline{\vec{\boldsymbol{h}}}$, the spatial momentum in O coordinates, the angular momentum $\vec{k}_C$ must be converted to $\vec{k}_O$, in a similar fashion to the couple in a force: $\vec{k}_O = \vec{k}_C + \vec{r}_{OC} \times \vec{l}$, where $\vec{r}_{OC} = \vec{c}$ by the definition above. This leads to a formulation for ${}^{O}\underline{\vec{\boldsymbol{h}}}$ as follows

$$ {}^{O}\underline{\vec{\boldsymbol{h}}} = \begin{bmatrix} \underline{\vec{k}}_O \\ \underline{\vec{l}} \end{bmatrix} = \begin{bmatrix} \underline{\vec{k}}_C + \tilde{c}\underline{\vec{l}} \\ \underline{\vec{l}} \end{bmatrix} = \begin{bmatrix} \tilde{1} & \tilde{c} \\ \tilde{0} & \tilde{1} \end{bmatrix} {}^{C}\underline{\vec{\boldsymbol{h}}} = {}^{O}\tilde{\boldsymbol{X}}^{*}_{C}\,{}^{C}\underline{\vec{\boldsymbol{h}}} \tag{A.24} $$

It is easy to verify this relationship in terms of the $\tilde{R}$ and $\tilde{r}$ variables from equation (A.11) by recalling that the frame of reference at the COM does not rotate, making $\tilde{R}$ a 3×3 identity matrix, and that the transformation matrix ${}^{O}\tilde{\boldsymbol{X}}^{*}_{C}$ is translating from C to O coordinates, so $\vec{r} = -\vec{c}$ and therefore $\tilde{c} = \tilde{r}^T$. Building on this understanding, a mapping between ${}^{O}\underline{\vec{\boldsymbol{h}}}$ and ${}^{O}\underline{\vec{\boldsymbol{v}}}$ can be used to generate the equality: ${}^{O}\underline{\tilde{\boldsymbol{I}}} = {}^{O}\tilde{\boldsymbol{X}}^{*}_{C}\,{}^{C}\underline{\tilde{\boldsymbol{I}}}\,{}^{C}\tilde{\boldsymbol{X}}_{O}$:

$$ {}^{O}\underline{\vec{\boldsymbol{h}}} = {}^{O}\tilde{\boldsymbol{X}}^{*}_{C}\,{}^{C}\underline{\vec{\boldsymbol{h}}} = {}^{O}\tilde{\boldsymbol{X}}^{*}_{C}({}^{C}\underline{\tilde{\boldsymbol{I}}}\,{}^{C}\underline{\vec{\boldsymbol{v}}}) = {}^{O}\tilde{\boldsymbol{X}}^{*}_{C}\,{}^{C}\underline{\tilde{\boldsymbol{I}}}({}^{C}\tilde{\boldsymbol{X}}_{O}\,{}^{O}\underline{\vec{\boldsymbol{v}}}) = {}^{O}\underline{\tilde{\boldsymbol{I}}}\,{}^{O}\underline{\vec{\boldsymbol{v}}} \tag{A.25} $$

where the construction of ${}^{O}\underline{\tilde{\boldsymbol{I}}}$ can easily be verified by recalling that ${}^{C}\tilde{\boldsymbol{X}}_{O} = {}^{O}\tilde{\boldsymbol{X}}^{*\,T}_{C}$. This also leads to a more general equation for transforming a spatial inertia matrix from one coordinate frame to another:

$$ {}^{B}\underline{\tilde{\boldsymbol{I}}} = {}^{B}\tilde{\boldsymbol{X}}^{*}_{A}\,{}^{A}\underline{\tilde{\boldsymbol{I}}}({}^{B}\tilde{\boldsymbol{X}}^{*}_{A})^{T} = {}^{B}\tilde{\boldsymbol{X}}^{*}_{A}\,{}^{A}\underline{\tilde{\boldsymbol{I}}}\,{}^{A}\tilde{\boldsymbol{X}}_{B} \tag{A.26} $$

Equation of Motion

The derivative of the spatial inertia of a rigid body, which is moving with velocity $\vec{\boldsymbol{v}}$, is

$$ \tfrac{d}{dt}\tilde{\boldsymbol{I}} = \tilde{\boldsymbol{v}}^{*}\tilde{\boldsymbol{I}} - \tilde{\boldsymbol{I}}\tilde{\boldsymbol{v}} \tag{A.27} $$

which directly leads to the equation of motion for a rigid body (recall $\vec{\boldsymbol{h}} = \tilde{\boldsymbol{I}}\vec{\boldsymbol{v}}$, $\tilde{\boldsymbol{v}}\vec{\boldsymbol{v}} = 0$):

$$ \vec{\boldsymbol{f}} = \dot{\vec{\boldsymbol{h}}} = \tfrac{d}{dt}(\tilde{\boldsymbol{I}}\vec{\boldsymbol{v}}) = \tilde{\boldsymbol{I}}\dot{\vec{\boldsymbol{v}}} + (\tfrac{d}{dt}\tilde{\boldsymbol{I}})\vec{\boldsymbol{v}} = \tilde{\boldsymbol{I}}\vec{\boldsymbol{a}} + \tilde{\boldsymbol{v}}^{*}\tilde{\boldsymbol{I}}\vec{\boldsymbol{v}} - \tilde{\boldsymbol{I}}\tilde{\boldsymbol{v}}\vec{\boldsymbol{v}} = \tilde{\boldsymbol{I}}\vec{\boldsymbol{a}} + \tilde{\boldsymbol{v}}^{*}\vec{\boldsymbol{h}} \tag{A.28} $$

where $\vec{\boldsymbol{f}}$ is the total spatial force acting on the body and $\vec{\boldsymbol{h}}$ is its spatial momentum.

For a more in depth treatment of spatial vectors as they relate to the dynamics of robots in general, readers are invited to consult [28] or [29].