



ISTITUTO ITALIANO
DI TECNOLOGIA



JIMI: A Hopping Monopod Robot Incorporating Nonlinear Series Elastic Actuators, Fiber-Reinforced Polymer Construction, and a Concurrent Asynchronous Dataflow-based Centroidal Momentum Balance Controller

by
Ivar L. Thorson

Istituto Italiano di Tecnologia

Università di Genova

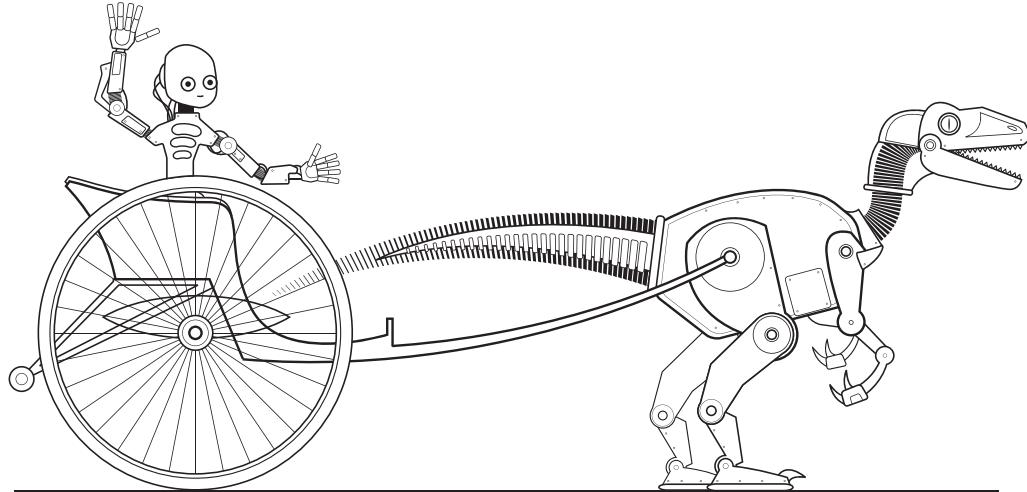
A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
Advanced Robotics

2012

Thesis Supervisors: Dr. Nikos Tsagarakis
Advanced Robotics Department
Italian Institute of Technology

Prof. Darwin Caldwell
Advanced Robotics Department
Italian Institute of Technology



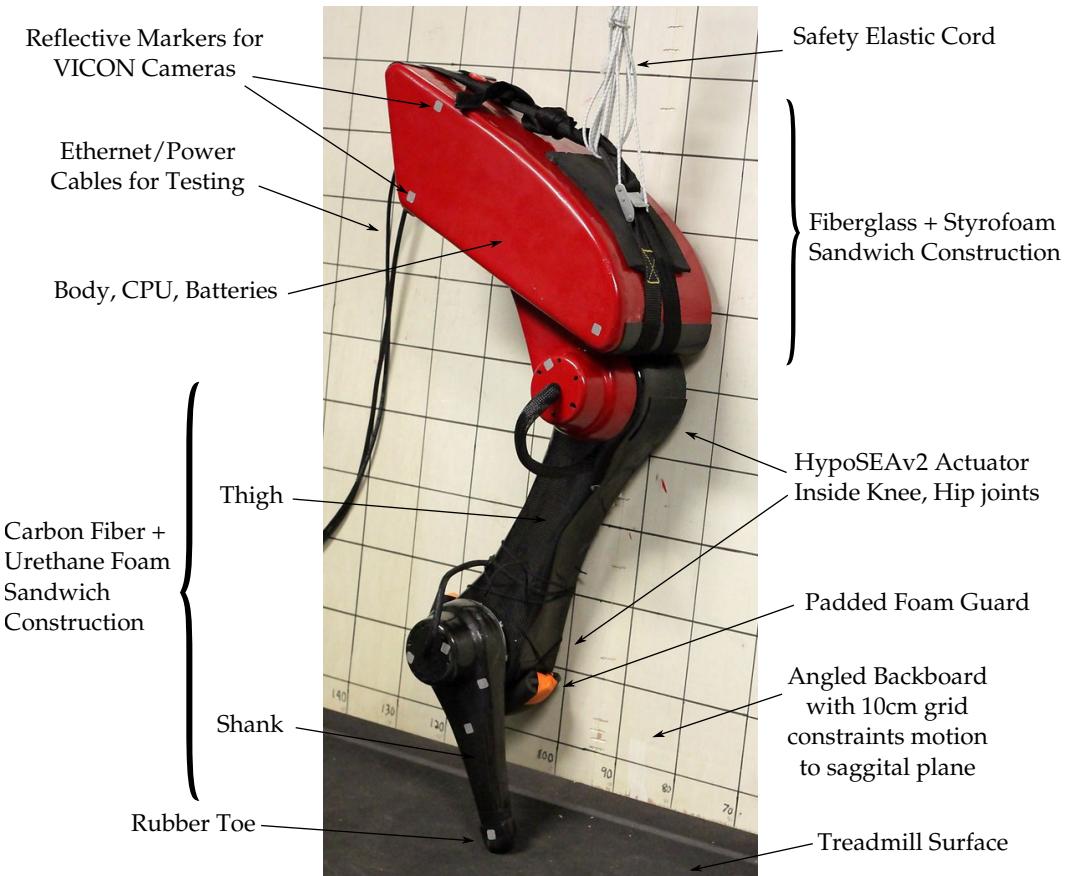
COPYRIGHT ©2012 IVAR L. THORSON. ALL RIGHTS RESERVED.
THE HYPOSEA ACTUATORS PRESENTED IN THIS THESIS ARE PROTECTED BY
ITALIAN PATENT #TO2011A000848: "ATTUATORE ROTANTE ELASTICO"
US PATENT #20130074635: "ELASTIC ROTARY ACTUATOR"

Abstract

This thesis presents three improvements to recently-developed technologies under-utilized in the field of legged robotics and employs these technologies to advance the current state of the art. Specifically, the improvements presented are:

1. Two different implementations of a *novel nonlinear series elastic actuator* that possesses superior energy storage, maximum elastic deflection, and force fidelity as compared to existing compliant actuators;
2. An *inverse-dynamics-based centroidal momentum balancing controller* that stabilizes a monopod robot via ground reaction forces and was developed in a rigid body dynamics simulation for the purpose of matching control dynamics with the mechanical dynamics of the robot;
3. A *novel asynchronous, concurrent soft real-time control system* that uses functional dataflow programming and immutable data structures for lock-free distributed computation and contained fault tolerance.

To provide a real-world demonstration of the capabilities of these three technologies, they are integrated into the construction of a planar monopod hopping robot called JIMI. The robot features carbon fiber-reinforced polymer composite monocoque structures, two compliant nonlinear actuators, and can be fully power autonomous. It weighs 8.3kg without batteries or PC, 10.9kg when fully loaded, stands 120cm tall, and can leap vertically a maximum of 40cm.



Acknowledgments

“Art is never finished, only abandoned. Doubly so for my thesis.”

– Leonardo da Vinci

“The trouble with quotes on the Internet is that you can never know if they are genuine.”

– Abraham Lincoln

In no particular order, I would like to express my thanks to Bram Vanderborght, for inviting me to Italy in the first place and helping me settle in; thanks to Prof. Darwin Caldwell and Nikos Tsagarakis, for patiently tolerating my inevitably irregular requests and allowing me to follow my curiosity unimpeded down strange paths for nearly three years; thanks to Irene Sardellitti, Claudio Semini, and Jonas Buchli for their advice as researchers just a few years ahead of me; thanks to Gianluca Pane and Phil Hudson, for spending so much time helping me when I had troubles with the mechanical and electrical details of this project; thanks to Roy Featherstone for so patiently answering many questions about rigid body dynamics via email; thanks especially to Daniela for marrying me and making me the luckiest man alive...and perhaps more apropos to this thesis, for taking the initiative to find me some two-part expanding polyurethane foam on short notice!

Preface

In 1997, at the age of 15, I attended my first meeting of the Seattle Robotics Society (SRS). At the time the SRS was a hobbyist-level club populated almost exclusively by enthusiastic male engineers wanting to build something neat in their spare time – most participants were slightly balding, geeky, and extremely generous to young people with their advice, time, and spare parts. Even to a kid, the profession of each participant was obvious from a single glance at their robots. The electrical engineers made stacks of printed circuit boards that formed the vague shape of a robot. The software engineers strapped a laptop and camera to the top of a round platform and began doing image processing or solving mazes. The mechanical engineers made beautiful but vaguely threatening polished metal sculpture that possessed exceptionally strong motors and a joystick for remote-control.

Besides strengthening my interest in robotics, the lesson that I took away from the experience was that we all subconsciously play to our engineering strengths. Yet our natural tendency towards specialization makes designing functioning robots even more difficult because it such a broad discipline; the proper operation of a robot requires harmonious interaction between its sub-systems in a manner that is more tightly coupled than most engineering disciplines. Especially for dynamic robots, mechanical, electrical, and control engineers cannot remain ignorant of each others' work. It may be for this reason that the Japanese coined the word “Mechatronics” to describe the principles of codesign they wisely espouse.

This thesis therefore runs the gamut of robotic engineering and touches on several issues in an exploratory way rather than focusing in depth on any one subject, an approach of which I am both proud and regretful. Were I to work towards this Ph.D. a second time, I would have restricted the set of problems that I approached and explored them in more depth, so that my thesis might serve as a more useful manual of expertise to others. Still, it is my deep hope that despite the many warts and flaws in this thesis, the reader may find in this thesis some useful ideas or small kernels of value, and be inspired to try their own hand at robot-building. Thank you for reading.

Contents

Abstract	iii
Acknowledgments and Preface	iv
Nomenclature	x
1 Overview	1
1.1 Changing Priorities in Robotics	2
1.2 Motivation for Legged Robots	3
1.3 Thesis Outline: The Road Ahead	4
1.4 The Robot JIMI	5
1.5 Contributions of this Thesis	5
2 State of the Art	7
2.1 Legged Robots	7
2.2 Actuation Systems	11
2.2.1 Ideal Force-Domain Actuators	11
2.2.2 Real Force-Domain Actuators	13
2.2.3 Series Elastic Actuators	15
2.3 Centroid Dynamics Control	17
2.4 Estimation and System Identification	19
3 Simultaneous Design of Mechanical and Controller Dynamics	21
3.1 Design Goals for JIMI	22
3.2 Sagittal Plane Model of JIMI	23
3.3 Rough Estimates of Mechanical Dynamics	24
3.3.1 What are the rough vertical motions of a hopping robot?	24
3.3.2 How much energy is stored and released each hop?	26
3.3.3 What are the joint torque and velocity requirements during stance?	27

3.3.4	Rough Analytical Conclusions	30
3.4	Numerical Simulation of Mechanical Dynamics	31
3.4.1	Equations of Motion	31
3.4.2	Numerical Integration	33
3.4.3	Joint-Link Connection Incidence Matrix	34
3.4.4	Types of Joint Torques	34
3.4.5	Contact with Ground	35
3.4.6	Transitions between Flight and Stance Phases	36
3.4.7	Ground Force Friction Constraint Verification	37
3.4.8	Simulation Parameters	37
3.5	Description of the Simulator	38
3.6	Simulation Results	39
3.7	Discussion of Hopping Dynamics	42
3.8	Conclusion	44
4	Nonlinear Series Elastic Actuators	45
4.1	Existing Series Elastic Actuators	45
4.2	Overview of the HypoSEA	46
4.2.1	Electric Motor and Transmission	46
4.2.2	Nonlinear Elasticity	47
4.2.3	Hypocycloids	48
4.2.4	Hypocycloid Spring-Stretching Mechanism	48
4.2.5	Mathematics of Planetary Gear Systems	50
4.2.6	HypoSEA Design Advantages	51
4.3	The HypoSEA-v1	52
4.3.1	Assemblies	52
4.3.2	Sensors	54
4.3.3	HypoSEA-v1 Transmission	54
4.4	Detailed Description of HypoSEA-v1 Components	57
4.4.1	Motor Assembly (M)	57
4.4.2	Compound Planetary Gear Assembly (G)	59
4.4.3	Planetary Differential Gear System (D)	60
4.4.4	Hypocycloid Assembly (H)	62
4.4.5	Elastic Element Assembly (E)	64
4.5	HypoSEA Torque Controller	65
4.6	HypoSEA-v1 Experimental Results	66
4.6.1	Mechanical Parameters	66
4.6.2	Maximum Torque	68
4.6.3	Bandwidth	69
4.6.4	Energy Recovery with the HypoSEA-v1	71
4.6.5	Motivation for Redesign	72
4.7	The HypoSEA-v2	73
4.7.1	Component Design Changes	75
4.7.2	HypoSEA-v2 Transmission	75
4.7.3	Stress Analysis	77
4.8	HypoSEA-v2 Experimental Results	77

4.8.1	HypoSEA-v2 Bandwidth	77
4.8.2	Mechanical improvements to the HypoSEA v2	78
4.9	Conclusions and Future Work	78
5	Centroidal Momentum Control of Legged Robots	80
5.1	A Condemnation of Trajectory Tracking	81
5.2	Choosing Nominal Hopping Dynamics	82
5.2.1	Sources of Inspiration	82
5.2.2	Essential Ideas in Hopping Dynamics	83
5.2.3	Dynamic Balancing : The Art of Not Falling Down	84
5.2.4	Constraints	85
5.3	Inverse Dynamics	85
5.3.1	Stance Phase Inverse Dynamics	86
5.3.2	Flight Phase Inverse Dynamics	87
5.3.3	Virtual Leg Task Space	87
5.3.4	Centroidal Momenta	87
5.4	Centroidal Momentum Balancing Controller	87
5.4.1	Controller Overview	87
5.4.2	Vertical GRF Controller	88
5.4.3	Centroidal Angular Momentum Controller	89
5.4.4	Centroid Ballistic Motion in Flight	90
5.4.5	Leg Swing	91
5.5	Conclusion	91
6	Model-Based Estimation	93
6.1	A Brief Review of Inference Techniques	95
6.1.1	Bayes' Theorem	95
6.1.2	Gaussian Distributions	96
6.1.3	Maximum Likelihood Gaussian Inference	97
6.1.4	System Model for Estimation Tasks	99
6.1.5	Least Squares (LS)	99
6.1.6	Maximum Likelihood (ML)	102
6.1.7	Maximum a Posteriori (MAP)	103
6.1.8	Marginalization	103
6.1.9	Nonlinear Estimation	103
6.2	HypoSEA Observer Design	103
6.2.1	Selecting Model Complexity	104
6.2.2	Derivation of HypoSEA System Matrices	105
6.2.3	State Measurement	108
6.2.4	Inference of Model Parameters	109
6.2.5	Real Time Inference	109
6.2.6	Fourth Order Fixed Lag Smoother	109
6.3	Estimation Performance Comparison	110
6.4	Power Flow Estimation	110
6.4.1	Power when consumption	110
6.4.2	Power Generation	110

6.4.3	Approximating Derivatives Using Finite Differences	110
6.4.4	Backward Finite Differences	110
6.4.5	Central Finite Differences	111
6.4.6	Approximate Derivatives with Irregular Sampling	111
6.5	Least Squares	112
6.5.1	Model Verification	112
6.6	Weighted Least Squares	112
6.7	Recursive, Weighted Least Squares	112
6.7.1	The Kalman Filter	112
6.7.2	Mixing Gaussians	113
6.7.3	Operation of the Kalman Filter	114
6.8	Extended Kalman Filtering on Instantaneous Powers	114
6.9	Interpretation	114
6.10	Real World Problems Encountered	114
6.11	Performance Comparison of Numerical Differentiation	116
7	Asynchronous, Concurrent, Functional Programming	117
7.1	Motivation	117
7.1.1	Better Abstractions	117
7.2	Backboard Compensation	121
8	JIMI: A Planar Hopper	122
8.1	Design Details	122
8.2	Mass Budget	129
8.3	Performance Comparison	129
8.4	Engineering Improvements to JIMI	131
8.4.1	Mass Reduction	131
8.4.2	Revised Composite Structure	131
8.4.3	Assembly and Manufacturing Costs	131
8.4.4	Electrical Improvements	132
9	Conclusion and Future Work	133
Bibliography		134
References		134
A	Bond Graphs	142
A.1	Constitutive Relations and Causality	144
A.2	Effort and Flow Sources (\mathbb{S})	145
A.3	Dissipative Components (\mathbb{R})	146
A.4	Conservative Components (\mathbb{I} and \mathbb{C})	147
A.5	Connectivity (0- and 1-junctions)	151
A.6	Gyrators ($\mathbb{G}\mathbb{Y}$) and Transformers ($\mathbb{T}\mathbb{F}$)	151
A.7	Drawing Bond Graphs	153
A.8	Multidimensional Bonds	154
A.9	Multi-Port Passive Components	155
A.10	From Bond Graphs to Hamiltonian Dynamics	155

A.10.1 Typical PHS Equations	156
A.10.2 Power Flow Constraints	156
A.10.3 Interpretation of a PD Controller	157
A.11 Examples	157
A.11.1 Example: Series RLC Circuit	158
A.11.2 Example: Spring-Mass-Damper System	159

Nomenclature

Abbreviations

BLDC Brushless Direct-Current Motor.

CoM Center of Mass. The mass centroid of a collection of one or more rigid bodies.

CoP Center of Pressure. Specifically for legged robots, it is the point on the robot's foot at which the sum of all the forces from the foot on the ground results in zero moment about that point.

CPG Central Pattern Generator.

DOF Degree of Freedom.

EKF Extended Kalman Filter

GRF Ground Reaction Force.

HF High Frequency.

HypoSEA Hypocycloid-based Series Elastic Actuator, the novel actuator presented in Chapter 4.

JIMI JIMI Is My Invention, a recursive acronym and nickname for the hopping robot presented in Chapter 8.

KF Kalman Filter.

LF Low Frequency.

LO Lift-Off, the moment when the foot lifts off the ground, the beginning of the flight phase.

LS Least Squares.

MAP Maximum *a posteriori* estimate.

ML Maximum Likelihood estimate.

PHS Port-controlled Hamiltonian System.

PID Proportional Integral Derivative control.

PWM Pulse Width Modulation.

SLIP Spring-Loaded Inverted Pendulum model of running.

TD Touch-Down, the moment when the foot strikes the ground, the beginning of the ground support phase.

UKF Unscented Kalman Filter.

Mathematical Notation

Equations in this book are written in approximately standard mathematical notation for engineers. Lowercase variables such as a, b, c indicate scalar quantities, such as velocities, forces, or lengths. Bold lowercase variables like $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are column vectors, and their uppercase cousins $\mathbf{A}, \mathbf{B}, \mathbf{C}$ denote matrices. Mathematical spaces use blackboard fonts, like \mathbb{R}, \mathbb{S} or \mathbb{SO} .

Variables are decorated in various ways. Undecorated variables like x are the *ground truth* or theoretically perfect values of the system. Dots indicate time derivatives: $\dot{x} = \frac{dx}{dt}$, $\ddot{x} = \frac{d^2x}{dt^2}$. The mean or average of a group of samples are marked with an overbar: $\bar{x} = \overline{\sum_{i=1}^N x_i}$. Groupings of samples are marked with an underbar: $\underline{x} = [\underline{x}_0^T \quad \underline{x}_1^T \quad \underline{x}_2^T]^T$. Estimated quantities (values conditioned from some other distribution) are marked with a hat (\hat{x}), and are usually maximum likelihood estimators. Stars indicate dual quantities: x^* . Probabilities follow convention; the marginal probability of A is written $P(A)$, the joint probability of A and B written $P(A, B)$, and conditional probabilities are written $P(A|B, C)$.

The mathematics in this book spans several engineering domains; without adding new letters to the Roman or Greek alphabets some overlap of symbols will be unfortunately inevitable. That said, the letter itself will usually indicate the type or units of the quantity and the subscript the particular engineering component to which it refers. The following table may help you navigate any ambiguities that may arise.

Symbol	Normally denotes	As a subscript
1	The identity matrix; vector of all 1's	
ϵ	Process noise	
ε	Measurement noise	
η	Efficiency	
θ	Angle, measured in radians counter-clockwise from the vertical axis	Virtual leg angle from vertical
Λ	Impulse vector	
λ	Force vector; Lagrange multipliers; Eigenvalues	
μ	Mean	
σ	Standard Deviation; Hop cycle completion (0 to 2π)	
τ	Torque	
Φ	Magnetic flux	
φ	Model parameters	

Symbol	Normally denotes	As a subscript
Ψ	Constitutive relation	
ψ	Duty cycle	
ω	Angular velocity (same as $\dot{\theta}$)	
A	PHS conservative components matrix	Annulus gear
a	Distance from CoM of limb to previous joint	
B	PHS external connection matrix	Battery; Backboard
b	Distance from CoM of limb to next joint	Body/torso of robot
C	Coriolis and centrifugal term matrix; Capacitance	Planet gear carrier
c	Viscous damping constant	System mass centroid
D	Incidence matrix	Derivative gain
d	Displacement	Desired quantity
E	Energy; Expected value	Back-EMF on motor
e	Effort; error	Elastic element
F	Force	Flight phase
f	Flow	Foot/CoP; Friction
G	Gear velocity ratio	Ground
g	Gravity	Gravitational
H	Hamiltonian; total system energy	
h	Height; angular momentum	Hip
I	Impulses; Integral gain term	
i	Current	A numerical index in a sequence
J	Jacobian	
j	Imaginary number ($\sqrt{-1}$)	Joint quantity
K	Coefficient (often linear), or user-selected parameter	
k		Knee
l	Length	Virtual leg from CoM to CoP
L	Moment of inertia; Inductance; Loss function	
M	Inertial matrix	
m	Mass	
N	Number, count, or quantity	
n	Index number	
o		Output link of the HypoSEA in test configuration
P	Probability; Power	Proportional gain term; Planet gear
p	Linear momentum	Pretension distance
Q	Covariance Matrix	
q	State vector	

Symbol	Normally denotes	As a subscript
R	Resistance; PHS dissipative matrix	
r	Radius	Rotor
S	Selection matrix	Sun gear
s	S-domain complex frequency	Shank
T	Duration of time; Kinetic energy	
t	Moment in time	Thigh
U	Potential energy	
u	PHS external effort (<i>i.e.</i> control) vector	
V	Linear velocity	
v	Voltage	Viscous friction
x	X-axis, horizontal position; State vector	“Forward” for a robot on the sagittal plane
y	Y-axis, vertical position; measurement vector	“Up” for a robot on the sagittal plane
z	Z-axis	“Out” of sagittal plane; PHS external flow vector

CHAPTER 1

Overview

This thesis presents the design, construction, assembly, modeling, programming, and control of a planar monopod hopping robot shown on the abstract (page iii). The robot is an experiment in the practical application of several conventional technologies that are underexploited in the field of robotics: centroidal momentum control, extreme series elasticity, functional dataflow programming, and fiber-reinforced polymer composite construction.

More broadly, the emphasis of this thesis rests primarily on applied engineering and mathematical techniques used to build a hopping robot. Although perhaps the technological aspects of this thesis will become more quickly dated than pure theory, the development of practical applications still represents a contribution to science and broader society. For example, the development of practical legged robots is often linked to the study of human biomechanics and leg prostheses. Further improvements to legged robot technology will undoubtedly lead to mobile robot applications in commercial, medical, aerospace, military, and consumer markets.

The development of robots is made challenging by the interdisciplinary engineering challenge they represent – not only are there simply many details to consider, it is hard to be an expert in all of them, and nearly everything is critical to successful operation. The cumulative effect of many small components significantly determines the resulting behavior of a complex robot, even if every individual component is well understood in isolation. This is particularly true for highly dynamic robots, in which components interact so strongly as to be very difficult to consider independently. The design of a controller for a robot is influenced by the dynamics of its mechanics, which are influenced by the actuators, whose requirements are determined by the controller and its interaction with the environment; see Figure 1.1.

Therefore, the cornerstone of this thesis is to maintain a relatively holistic viewpoint and not forsake one subtopic of robotics for another. One of the most elegant and unifying discoveries about the dynamics of physical systems was to connect everything via the flow of energy. We should respect this flow across domains, and embrace the fundamentally interdisciplinary problems presented by highly dynamic robots.

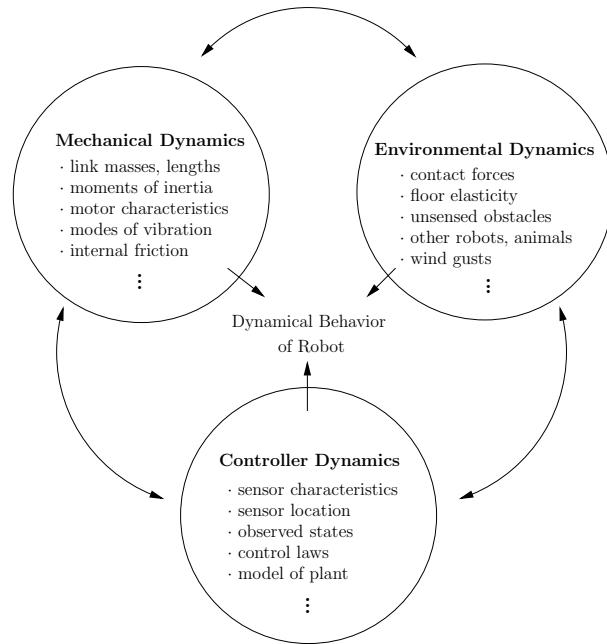


Figure 1.1: End-result robot dynamics can never be specified only by a control system, because dynamics depends critically on interaction forces between the mechanical structure, controller, and environment. Ideally, mechanical and control dynamics are considered simultaneously during the design process so that the natural mechanical motion of the system is not treated as a control disturbance to be eliminated.

1.1 Changing Priorities in Robotics

Despite a widespread popular enthusiasm for robots, the state of modern day robotics leaves a lot to be desired. It takes years to develop the mechanical structures of sophisticated robots. It takes years to program robots to do what appear to the layman as relatively simple tasks. There is an unfortunate gap between industry and academics – industry makes conservative, reliable designs with new technology and old theory, while academics are frequently inept engineers whose technical mistakes fundamentally limit their theoretical results. Despite the obvious benefits cooperation would bring, collaboration between research robotics and industry is unfortunately limited.

Over the years the driving economic force behind robotics has been the field of industrial manufacturing, so perhaps it is not surprising that most robot engineering techniques are adopted from common industry practice. Characteristics of manufacturing robots include:

1. Most mechanical structures are heavy and extremely stiff, in order to resist positional disturbances regardless of the forces on them.
2. A single trajectory is repeated over and over again.
3. The environment in which a robot operates is static, completely known, and generally unchanging.
4. The control software is run in tight, deterministic, synchronous control loops.
5. The dynamics of the system are modeled in order that they may be canceled out and the machine operated in arbitrary ways by people ignorant of or uninterested in its dynamics.
6. The robot typically cannot sense the forces being transmitted, nor can its transmission be easily back-driven.
7. Constraints on the system are typically formulated in a hard way – deadlines are absolutely guaranteed, constraints are absolute, and position trajectories are desired to be tracked as perfectly as possible.

8. Failure of any component ideally leads to a full system stop.
9. Only professionals are allowed to be near the robot, and even then it is often extremely dangerous.

If robots are to become truly widespread in general society as the personal computer, we must make robots of a fundamentally different quality. To safely interact with people, robots must become softer, lighter, and capable of more finesse. Rather than just tracking a single trajectory, they must begin to quickly plan new trajectories in response to every tiny disturbance. They must continually adapt to noise, time-varying dynamics, and even changes to their structure. Their interfaces and software must become simple enough that anyone can program them through simple demonstration techniques.

Researchers in the field of robotics understand these problems, and new developments are proceeding in a direction almost polar opposite to standard industry practice:

1. Stiff transmissions are being replaced by ones with compliant elements.
2. Heavy, dangerous steel robots are becoming lightweight through the use of advanced materials.
3. The mechanical dynamics of a robot are being considered by its controller to minimize energetic cost or actively control its impedance.
4. Advances in control software and programming by demonstration techniques are reducing the skill level required to teach robots to perform a task.
5. Hard, deterministic control loops are being replaced by softer, fault-tolerant deadlines.
6. Trajectory tracking controllers are being replaced with planned management of system dynamics.
7. There is a growing awareness of scalability, complexity, and integration issues.
8. Off-line algorithms are being replaced by incremental, online versions.
9. Fault tolerance is becoming critical, as sophisticated designs with more components result in a higher probability of one or more components failing.

Many of these trends present both theoretical challenges for the academic and technological challenges for the engineer that are active subjects of research.

1.2 Motivation for Legged Robots

Legged robots show great promise in navigating rough terrain and environments designed for humans, yet at the moment typically have inferior performance and efficiency as compared to their wheeled counterparts. These are not an intrinsic theoretical limitation of legged robots – indeed, on rough terrain it is easy to conceive of situations where legged machines could smoothly step from safe foothold to safe foothold and outperform any wheeled robot of similar size. Passive dynamic mechanical walkers have already demonstrated purely mechanical walking motions with gait efficiencies approaching that of humans (Collins, Ruina, Tedrake, & Wisse, 2005).

Rather, the inefficiency of legged robots appears to come from the enormous and often unfamiliar engineering challenges that they present to the researcher. Besides being theoretically difficult to analyze, running robots must generally have sufficient power-to-weight ratios that ground forces of three times body weight can be created. They must have accurate sensors and state estimators to calculate their spatial position during ballistic portions of their stride. Their balancing controllers must be low latency and robust to disturbances or the robot may

fall. Few suitable robot components are available off the shelf. These are non-trivial engineering problems that complicate the construction of successful legged robots.

Additionally, the can also be surprisingly difficult to analyze theoretically when treated as a general dynamic system. Running robots have many degrees of freedom, are underactuated, nonlinear, discretely switch between two types of operational states (flight/stance phases), and must satisfy equality and inequality contact constraints during their interaction with the ground. Even measuring their performance is often subjective due to a lack of widely used metrics for gait stability, although attempts have been made from a mechanical dynamics perspective (Hobbelin & Wisse, 2007) (Thorson, Svinin, Hosoe, Asano, & Taji, 2007), a control perspective (J. Pratt & Tedrake, 2006), and a biological one (Kubow, Schmitt, Holmes, Koditschek, et al., 2002).

One important objective of this thesis is to improve the art of legged robot construction. By addressing these engineering concerns, perhaps legged robots can be brought out of the laboratory and into wider public life. Legged robots with high levels of mobility could not only navigate spaces designed for humans with relative ease, they could open new avenues of automation in dangerous occupations like forestry, coal mining, landmine removal, disaster recovery, and inspection of other difficult-to-access areas like contaminated nuclear reactors.

The construction of legged robots is not just of interest to engineers, however. Building legged robots with specific characteristics is also one way to test theories on the biomechanics of animals (Griffin, Main, & Farley, 2004) – we cannot yet build new animals, after all. Experiments with walking and running robots can yield insight as to why animals move the way they do. Knowledge of the nominal locomotion of animals has led to improved prosthetics (Blaya & Herr, 2004) which greatly improve the quality of life of amputees. Analysis of springy running has led to better, safer shoes for athletes (Shorten, 1993). Studying legged robots improves our ability to develop useful balance-assisting machines that help stroke victims and even paraplegics to walk and run again (Costa & Caldwell, 2006; Hyon, Morimoto, Matsubara, Noda, & Kawato, 2011).

1.3 Thesis Outline: The Road Ahead

Research must begin with questions, with pointed directions of curiosity that later blossom into creative activity and experimentation. To succinctly summarize the focus of subsequent chapters of this thesis, the questions addressed are as follows:

Chapter 2 What is the current state of the art in legged robotics?

Chapter 3 How can we study the interaction between a robot's mechanical and control systems during the design process, and use this information to ensure that each system complements the others?

Chapter 4 What kind of actuator would best suit a highly dynamic, revolute-jointed legged robot?

Chapter 5 What is the essential nature of balancing dynamically, and how can it be controlled in terms of the momenta of the mass centroid?

Chapter 6 How can we continuously improve our estimation of the state and parameters of a system with mixed linear and nonlinear components?

Chapter 7 How can we write control software for a robot in such a way that a great number of operations may be performed concurrently on a computer with a large number of processing cores?

Chapter 8 is the capstone of this thesis. It represents the technical integration of all previous chapters into a planar monopod hopping robot named "JIMI". It also summarizes the preliminary performance of the robot and looks toward avenues of future research.

Additionally, since the author has received many questions about how he constructed the fiber-reinforced polymer (FRP) composite structure of JIMI, Appendix ?? includes a tutorial on how the average robotics researcher can make strong, stiff, and lightweight robot structures with a limited number of tools. While not particularly scientific, it provides useful know-how for those unfamiliar with FRP construction processes.

For those unfamiliar with system dynamics, a brief introduction to bond graphs and the Port Hamiltonian modeling approach is included in Appendix A.

1.4 The Robot JIMI

The robot JIMI will be described more fully in Chapter 8, but several things should immediately be highlighted about its design:

- The dynamical interaction between JIMI's mechanical system, actuation system, control system, and environment were modeled and considered in simulation as the design process unfolded. Mechanical and control dynamics were selected simultaneously. In contrast, the design of the mechanics and control systems for most robots are treated as largely independent engineering tasks.
- The dynamic balancing controller for JIMI does not rely on torque or position trajectory tracking as most robots do, but instead uses virtual dynamic fields to move in family of trajectories that satisfy the holistic dynamic requirements of hopping.
- JIMI is electrically actuated, is self-contained and can be power autonomous. Many hopping robots use pneumatics or hydraulics with external power sources to improve the robot's power-to-weight ratio.
- The actuators used in JIMI can not only store a tremendous amount of energy in an elastic spring, they can also efficiently recover mechanical energy into electrical energy. Most robots cannot recover mechanical energy at all.
- The load bearing structure is a very lightweight monocoque carbon fiber composite with a urethane foam core. Most robots use heavy endoskeletons of steel, aluminum, or plastic.
- Most robots use a hard real time language, a controller written in a low-level language like C, and a single-threaded control loop. The control system for JIMI is not hard real-time or guaranteed to run synchronously, is written in a highly dynamic functional programming language in the dataflow style, and can be executed on dozens of cores in an extremely concurrent fashion.
- System identification was performed with a recursive, maximum-likelihood model-based estimator that uses power constraints derived from the bond graph model of the system to ensure all inter-component power flows are accurately estimated.

These results will be expanded upon in later chapters.

1.5 Contributions of this Thesis

This thesis contributes to the body of academic knowledge in several ways:

1. It presents the design and construction of a novel high-performance, self-contained, electrically-actuated hopping monopod robot.

2. It presents the design and performance characteristics of two novel, patented nonlinear series elastic electric actuators with low impedance, significant amounts of energy storage, a high maximum elastic deflection, and the ability to generate electrical energy from mechanical motion.
3. It applies modern techniques of least-squares model-based system identification and state estimation such that power constraints are used to maintain model self-consistency.
4. It provides perhaps the world's first demonstration of a robot being controlled with asynchronous, concurrent dataflow techniques and software-transactional memory.
5. It presents a centroidal momentum task-space dynamic balancing controller and describes its implementation in detail.
6. It provides simple tutorials on the laboratory manufacture of high strength parts from fiber reinforced polymers.

To better understand these contributions in perspective, the next chapter will look at what amazing work has already been done by previous researchers in the field.

CHAPTER 2

State of the Art

This chapter presents a brief literature review of legged robots, their dynamics, series elastic actuation, and their control systems.

2.1 Legged Robots

An extensive review of the long history of legged robotics is beyond the scope of this thesis, and we will instead consider just the important milestones that provide context. As the focus of this thesis is on dynamic balancing and hopping, statically-balancing robots in particular will be completely ignored. Drawing a distinction between static and dynamic balancing can be difficult in some cases, but for our purposes the definition of “static” will refer to robots which move slowly, have limited momentum, and keep the center of mass over the support polygon at all times.

Without a doubt, the pioneering work of Marc Raibert (Raibert, 1985) at the MIT Leg Lab in the 1980’s inspired a generation of researchers and ushered in the era of dynamic running machines. In the span of less than a decade, the designs of the Leg Lab progressed from a simple planar monopod to a full 3D biped robot (Figure 2.1). More complex designs were also built, including a revolute-jointed monopod (W. Lee & Raibert, 1991), kangaroo-like hopper (Zeglin, 1991), and a quadruped (Raibert, 1985). The controllers of these robots were simple, effective, and not based upon tracking a single trajectory.

The work of Raibert in some ways remains unmatched – the author is not aware of *any* other robots to this day capable of a front-flip, as was demonstrated by the Leg Lab’s 3D biped robot (Playter & Raibert, 1992). The relevance of these robots to this thesis is that they did not generally use trajectories to balance, a trait shared by the controller of Chapter 5. Instead of tracking a single trajectory, the philosophical focus of the running controllers was upon the energetics and whole-body dynamics of the robot. Simple control laws were used to exert forces that would result in stable whole-system dynamics. Also in common with this thesis is that the Leg Lab robots exploited natural mechanical dynamics in their operation, perhaps none more explicitly than the Spring Flamingo robot (J. Pratt, 2000), shown in Figure 2.2, right.

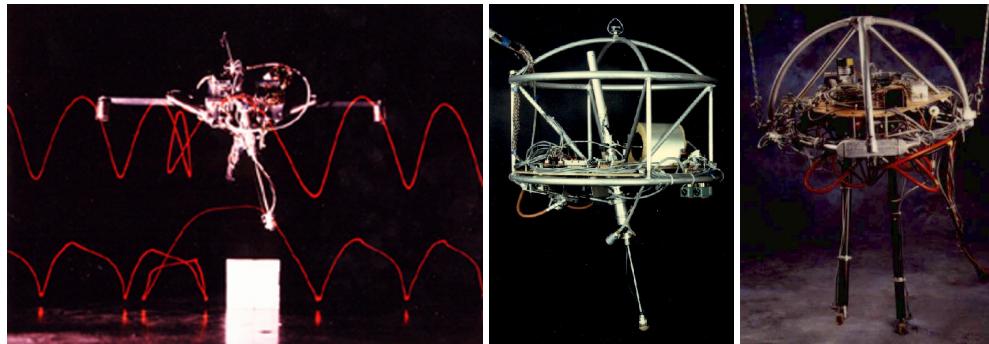


Figure 2.1: Telescoping-leg running robots from the MIT Leg Lab under direction of Marc Raibert. From left to right: the planar one-leg hopper, 3d hopper, and 3D biped.

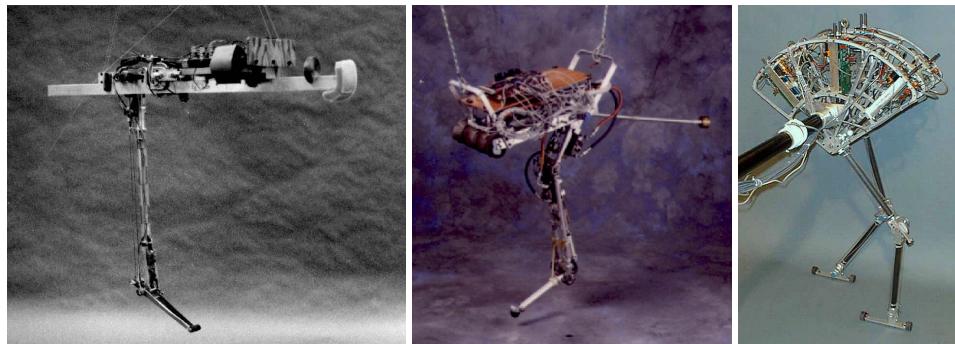


Figure 2.2: Three revolute-jointed robots also from the MIT Leg Lab. From left to right: a monopod, the “Uniroo”, and Spring Flamingo.

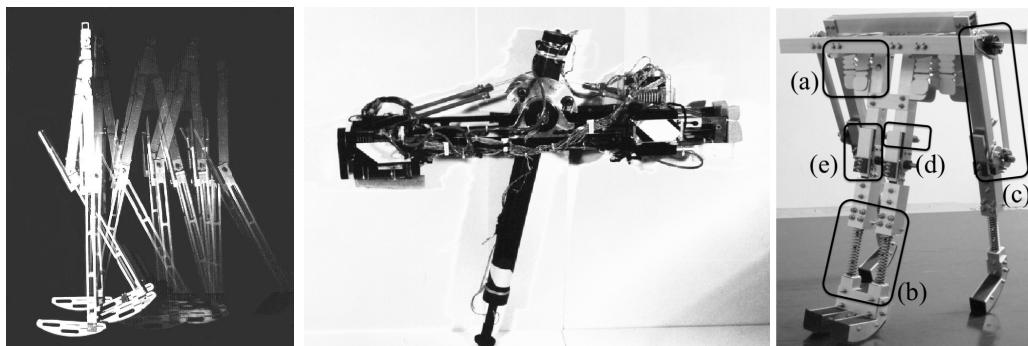


Figure 2.3: Left: A Cornell University duplicate of McGeer’s original passive dynamic walker. Center: The ARL-2 Monopod hopping robot, which has passive mechanical dynamics similar to hopping and is controlled to ensure stability. It is quite energetically efficient as a result. Right: The purely passive dynamic running robot PDR-400.

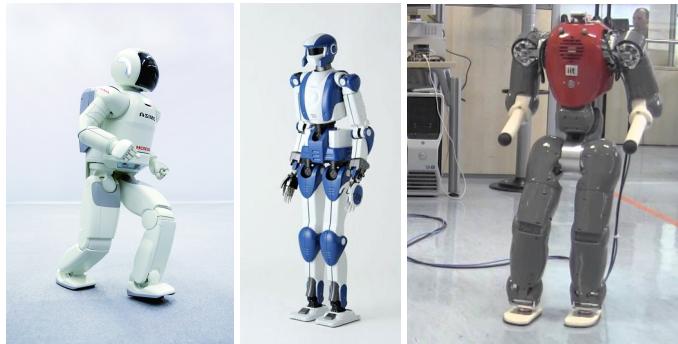


Figure 2.4: Left: The advanced ASIMO robot, built by Honda Motors Inc. Center: The HRP-4 Robot developed at Kawada Industries, Tsukuba, Japan. Right: COMAN is a compliant humanoid robot built at the Istituto Italiano di Tecnologia that incorporates compliant elements at certain joints.

Although perhaps less visually impressive, the work of Tad McGeer (McGeer, 1990) in passive dynamic walking has gradually become a full-fledged branch of study in the field of legged locomotion. Passive dynamic robots are purely mechanical machines yet exhibit strikingly realistic-looking walking strides – without any control or actuation at all. They are stabilized by the impacts of their feet with the ground, and are powered by gravitational potential energy from the slight downhill gradient on which they walk. More recently, robots have even been made to run completely passively (Owaki, Koyama, Yamaguchi, Kubo, & Ishiguro, 2010). Some robots with good passive dynamics are shown in Figure 2.3.

Perhaps the important lesson to be learned from such experiments is that some structures are just *ready* to walk or run – they already have the proper mechanical dynamics and lack merely a power supply and stabilization. If this is true, proper mechanical design can bring a robot 90% of the way to successful locomotion, provided that the controller does not attempt to overwhelm these beneficial natural dynamics. This lesson was taken to heart in this thesis, and the balance controller of Chapter 5 is designed to complement the mechanical dynamics presented in Chapter 3 as well as the actuators of Chapter 4.

The development of humanoid robots with up to 50 degrees of actuation represents another milestone in the engineering sophistication of legged robots. Particularly popular in Japan, humanoid robots that balance with Zero-Moment-Point (ZMP) control began to appear in increasing numbers and sophistication after the unveiling of the Honda P3 prototype (Hirose & Ogawa, 2007). Although Asimo (Sakagami et al., 2002) is probably the reigning king in terms of capability at this moment, there is considerable competition in the arena of humanoid robots and development is fast and furious (Kaneko, Harada, Kanehiro, Miyamori, & Akachi, 2008; Tajima, Honda, & Suga, 2009; Metta, Sandini, Vernon, Natale, & Nori, 2008). Figure 2.4 shows several advanced humanoid robot designs.

These humanoid robots are amazing machines, but they are not really very dynamic because they rarely balance well in all but the simplest environments. Many of these robots often naively follow a single, planned trajectory when walking. Small deviations from this single trajectory will be compensated for by the controller, but there is no real guarantee that following a joint-level trajectory will actually keep the robot balanced. Not only can a single significant disturbance topple them, even just angling the floor by a degree or two will often be enough to topple them.

More developed robots may perform more sophisticated online replanning techniques (Sakagami et al., 2002) to adapt to their environment. This begins to move them from being “just” tracking ZMP-constraint satisfying trajectories to being more dynamic and able to respond to the environment. Only a few use controllers that

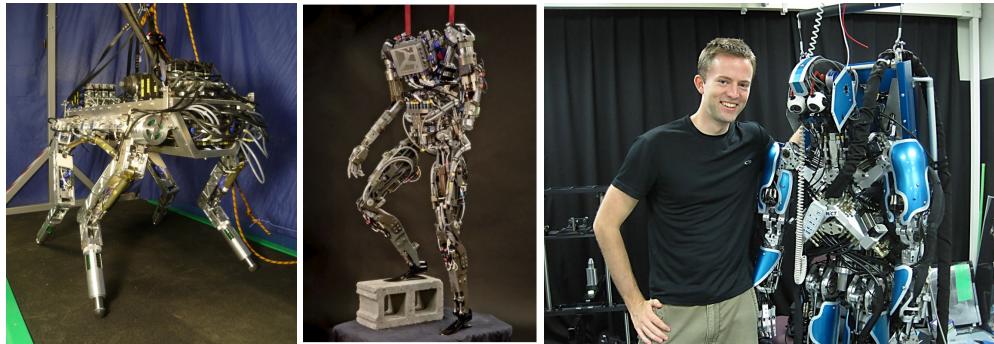


Figure 2.5: Three hydraulic robots with force control. Left: The HyQ hydraulic quadruped robot. Center: The PetMan robot built by Boston Dynamics. Right: The author with “CBI”, a hydraulically actuated humanoid robot built by Sarcos.

truly balancing dynamically (Hyon & Cheng, 2006). A few humanoids, such as COMAN (Li, Vanderborght, Tsagarakis, Colasanto, & Caldwell, 2012), include compliant elements at key joints, so that small deviations from the desired trajectory only generate small differences in anticipated forces, with the intent of using limited mechanical compliance to enhance the region of stability around the tracked trajectory.

Besides the controllers, a more relevant point regarding mechanical dynamics is this: *the natural dynamics of most humanoids more closely resembles a stiff manufacturing robot arm than anything else*. These robots are extremely stiff when turned off and certainly far from the elegant simplicity of passive dynamic walking.

For example, Asimo is said to be capable of running yet its flight time is just 40 milliseconds and its feet are never more than 3.2cm from the ground (Cho & Oh, 2008). Although some of the running dynamics are being captured by Asimo’s motion, it does not constitute the true internal dynamics of running in the sense that we humans know it. The illusion is dispelled the moment the robot experiences an unpredicted impact with a hard surface – the impact energy of an impulsive collision contains very high frequency energy, and this high frequency energy may be above the actuation bandwidth of the robot. It thus seems unlikely that extremely stiff robots such as Asimo will ever walk on unstructured, unknown rocky terrain.

That is not to say that stiff devices cannot mimic soft behavior despite having fundamentally different dynamics. With the addition of force-sensors and a force control loop, even very stiff actuators can be made to have low apparent impedance at bandwidths below their control frequency. We shall consider this technique more in the next section, as it is particularly popular with hydraulically powered robots such as BigDog (Raibert, Blanke-spoor, Nelson, Playter, et al., 2008), PetMan, the Sarcos humanoids (Cheng et al., 2006), and HyQ (Semini, 2010). The versatility of being able to change the effective physical impedance remains very attractive for robotics researchers because it avoids the time-consuming and costly step of mechanical redesign to physically change the passive mechanical impedance.

Although it is an impressive technique, there are problems with using stiff actuators to mimic soft ones. The largest problem is that of energy consumption, which is why such robots are typically hydraulically powered. It takes a lot of power to wave a heavy mass around as if it were a small one, since any difference between the programmed dynamics and natural dynamics must be filled by the control and actuation system.

If a robot has legs, it is already being purpose-built for walking or running. This suggests that the potential for mechanical specialization should be exploited, not avoided by using general purpose control techniques. Designing the structure of a robot to specifically suit legged locomotion could yield huge energetic benefits when compared to control techniques that artificially make a programmable impedance.

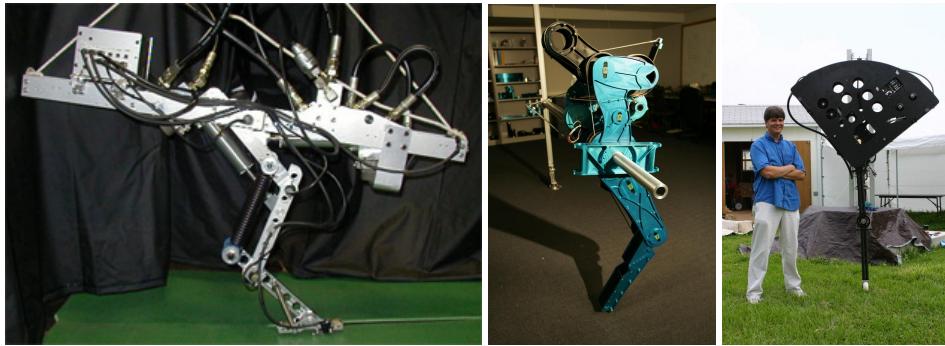


Figure 2.6: Robots with characteristics similar to the robot of this thesis. From left to right: The hydraulic KenKen, the electrically-actuated cable differential leg “Thumper”, and the IHMC monopod. All have series elasticity in their actuators and controllers that act on the momentum of the robot.

Finally, we turn to robots of Figure 2.6, which are close in spirit to the robot presented in this thesis. These robots include the Uniroo (Zeglin, 1991), (Ahmadi & Buehler, 2006), ECD leg (Poulakakis & Grizzle, 2009; Hurst, 2011), Kenken (Hyon & Mita, 2002), and the IHMC monopod (Krupp & Pratt, 2006). It is sufficient merely to list their names now, as they will be studied in more detail for comparison purposes in Section 8.3. All of these robots are monopedal, use force control techniques, have series elastic actuation and effectively point-contact feet. Many of these robots control system energy or momenta explicitly. Most use a task-space controller to regulate the motion of the system’s center of mass. Few use any type of trajectory tracking control. All of these robots also use some sort of elasticity – a crucial innovation that will be discussed in the next section.

2.2 Actuation Systems

Before discussing the work of other researchers, it will be instructive to review a little theory on actuation. Let us begin by asking what properties ideal actuators would have.

2.2.1 Ideal Force-Domain Actuators

Actuators are generally designed to either control quantities in the force domain (force, torque, momenta, etc) or position domain (position, velocity, acceleration, etc). These two spaces are dual to each other, meaning that they are in some sense two sides of the same problem; theoretically, proper control of the position or force can yield equivalent results. In practice, an actuator designed for force control looks somewhat different from one designed for position control, and the rules of causality mean that they are not perfectly equivalent.

A perfect force-domain actuator would exert any desired force regardless of its position. Conversely, a perfect position-domain actuator would hold any desired position regardless of the forces on it. We will might say that a perfect force-domain actuator is “compliant” in some sense – a device able to accurately track a predefined force trajectory in the presence of positional disturbances. In contrast to this we might call a perfect position-domain actuator a “stiff” actuator – a device able to accurately track a predefined position trajectory in the presence of force disturbances.

For manufacturing tasks, it is important to hold the position of the robot tip as fixed as possible so that the cutting bit, welding tip, or paint spray does not move too far from the desired target point. Naturally the focus in the design of such robots is on good position control and extremely stiff structures. We sometimes say that

Table 2.1: Effort and Flow Variables of Dual Domains

Dual Domains	Effort e	Flow f	State $x = \int f dt$	Impedance $e(x, f)$	Admittance $f(x, e)$
Electric	voltage	current	electric charge	Capacitor	
Magnetic	current	voltage	flux linkage		Inductor
Mechanical potential	force	velocity	displacement	Spring	
Mechanical kinetic	velocity	force	momentum		Mass
Hydraulic potential	flow volume	pressure	fluid volume	Bulk modulus	
Hydraulic kinetic	pressure	flow volume	fluid momentum		Fluid mass

such robots have very high mechanical impedance, meaning that they resist disturbances in a stiff way; the ideal position control actuator would in fact have *infinite* mechanical impedance. Conversely, compliant robots have low mechanical impedance; the ideal force control actuator would have *zero* impedance.

In the language of duality, impedance is defined as the ratio of an “effort” variable to its corresponding “flow” variable (Wellstead, 1979). This is such an important statement that it deserves to be the first equation in this thesis:

$$\text{impedance} = \frac{\text{effort}}{\text{flow}} \quad (2.1)$$

In the electric domain, the effort and flow variables are voltage and current; the impedance of a passive component with no energy storage is therefore equal to its resistance. In the fluid domain, the effort and flow variables are the pressure and fluid flow; we can thus logically talk about the impedance of a particular orifice through which fluid flows. In the mechanical domain, the effort and flow variables are force and velocity; mechanical impedance is thus similar to viscous friction in that it expresses how resistant the system is to being driven by an external force. The relationships between components in various domains is summarized in Table 2.1.

If one pushes on an actuator when it is programmed to provide zero torque, the mechanical impedance can be felt as the torque that actually results from that disturbance. If you push it faster, this torque will be larger, but some systems have different levels of impedance depending on the velocity at which you push them, so impedance is often specified for linear systems as being a value at a particular frequency.

Admittance is the reciprocal of impedance. There isn’t anything that we can say about impedance that isn’t true in a dual way about admittances. So let’s charge ahead and discuss which components can be thought of as being mostly “admittances” or “impedances.”

When we talk about an “admittance”, we are discussing components that have state that stores “effort”, and admittances are components that store “flow”. Each pair of physical domains will have both impedance components and admittance components with causal directions opposite to each other. In the electrical domain, they are components that are mostly capacitive (which resist the change of effort) or mostly inductive (which resist the change of flow). In the mechanical domain the impedances are springs and the admittances are masses. Springs “store” a force in the sense that when they continue to exert a force (an effort) when compressed. Conversely, masses “store” up momentum and hold a velocity (a flow) until disturbed further.

For a continuous mathematical treatment of their physics, impedances need always to be connected to admittances, and *vice versa*. Trying to simulate the collision of two masses connected by a spring is a fair thing to do; the spring’s impedance mediates the interaction forces between the two masses and a smooth oscillation results.

But if two perfectly rigid masses are smashed together without any mediating spring, the interaction forces go to infinity for an infinitely short amount of time – a discontinuity through which continuous simulation or control is impossible.

For this reason, we may say that positional actuators (flow control) are great for controlling impedances like springs, and that force/torque sources (effort control) are the better means to control admittances like masses. Most actuation in robotics connects two rigid bodies (admittances) together, so most actuation in robotics should use good torque actuators. This is particularly for jumping robots because they interact powerfully with the floor, a very large admittance indeed!

Unfortunately, present-day actuation systems have significant limitations that make them unsuitable for this kind of control paradigm. Before discussing the advantages limitations of specific technologies, let us consider the characteristics that an ideal force/torque actuator would have:

1. High maximum torque
2. High maximum angular velocity
3. High power-to-weight ratio
4. High energetic efficiency when performing work
5. High force-generating bandwidth
6. Low control latency
7. Energy recovery capability
8. Clean and safe for use around humans
9. Zero mechanical impedance

Finally, we should mention the concept of “backdrivability” as being a desirable characteristic for force control actuators to have. Backdrivability refers to how easily an actuator’s motor can be driven through the transmission. A transmission such as worm drive is completely non-backdrivable because its motor *cannot* be turned by pushing axially on the screw. Conversely, a motor with a single stage planetary gear system can be easily rotated by twisting on the output. Therefore, “backdrivability” means essentially the same thing as “low passive mechanical impedance.” Backdrivability is desirable because it allows the possibility of converting mechanical energy into electrical energy. Series elasticity allows a small amount of energy recovery in the potential energy of the spring, although the timing of the release of this energy is in general not controllable.¹

The future of robotics will undoubtedly rely heavily on control algorithms that work in the force and torque control domain, as it enables applications that are impossible to achieve using only positional control and actuation. For example, without accurate control of the forces being generated by a robot, it is impossible to precisely manipulate delicate objects of unmodeled composition or to safely interact with humans in an unstructured environment.

¹One possible exception to this statement would be the V2E2 design, which can vary the transmission ratio to the point where energy can be locked into the elastic element and released at arbitrary later times (Gerelli, Carloni, & Stramigioli, 2009).

2.2.2 Real Force-Domain Actuators

One huge obstacle that every roboticist faces is the lack of “good” robot force actuators. At present, there exists no type of robotic actuator – electric, hydraulic, pneumatic, piezoelectric, dielectric, shape-memory alloys, magnetostrictive, and polymeric actuators, or otherwise (Hollerbach, Hunter, & Ballantyne, 1992) – with properties that approach the sophistication and grace of animal muscle tissue. Muscles are lightweight, flexible, self-healing, control force accurately, have low friction, have good power density, and even possess limited onboard fuel storage. Robot motions tend to look robotic precisely because they have stiff actuators and large mechanical impedances at high frequencies. Animal motions tend to look natural precisely because they are compliant and have low impedance at high frequencies. Marc Raibert (Raibert & Hodgins, 1991) noted when presenting a controller for a simulated kangaroo for the purposes of computer animation that even physically plausible motions may not look particularly realistic, but lower stiffnesses generally tended to make motions look more natural.

Robots most often use electric, hydraulic, or pneumatic actuators. How do these compare to the ideal actuator outlined in the previous section?

Hydraulic systems can achieve power densities in excess of 10,000 W/kg, while human muscle is in general only 50W/kg or perhaps 200W/kg for short times (Hollerbach et al., 1992; McMahon, 1984). Hydraulic actuation systems can also produce large torques and can be quite fast, depending on their geometry. Yet despite these performance advantages, for safety and cleanliness reasons, it seems unlikely that high pressure hydraulic systems will find their way into the home. Hydraulic systems which operate using valves and a single pump also have very high mechanical impedance due to their minimally compressible fluids.

Pneumatic actuators have power densities in the thousands of W/kg and intrinsic compliance due to the compressible gas inside them. On the other hand, they often suffer from efficiency and control problems due to the inherent flow-rate limitations of their valves and nonlinear compressibility of the air. Compressing a gas will heat it, and this heat energy cannot be recovered. Pneumatic systems can be even more dangerous than hydraulic systems because of their potential to release stored energy very quickly. However, they do have a much lower mechanical impedance than hydraulic systems.

Electric actuation systems are very fast, clean and can have power densities of a few hundred watts per kilogram (Hollerbach et al., 1992). Unfortunately, they generally have such low torque capability that a gear speed reduction systems must be used to provide a mechanical advantage. The highest torque-to-weight ratio for a direct drive electric motor suitable for robotics is the $\sim 15\text{Nm/kg}$ direct drive motor developed at McGill University (Hollerbach et al., 1993; Aghili, 1998; Aghili, Buehler, & Hollerbach, 2002). Furthermore, the motor’s pancake geometry is not suitable for most applications, and in practice robots inevitably require gear systems to develop useful torques. A gear reduction of $N:1$ increases the reflected inertia and friction of the actuator by a factor of N^2 , meaning most electric actuators tend to feel very stiff through the gearbox.

What can we do about this problem of high impedance electric and hydraulic actuators? As mentioned in the previous section, one possible solution is to implement an impedance controller (Hogan, 1985) using a force sensor. With fast control loops, we may move the actuator in a way that increases or decreases the apparent mechanical impedance seen by the load – at least for speeds below the control loop frequency. Above this control frequency only mechanical effects are in play, and only the passive mechanical impedance matters. Very fast impacts with stiff objects can damage robots with simulated mechanical impedance because the impact introduces a great deal of energy at frequencies above the controllable bandwidth. For this reason, such techniques are generally not suitable for a highly dynamic robot.

Despite this criticism of controlled impedance, it must be acknowledged that the HyQ robot (Semini, 2010) provides an excellent demonstration of the utility of this technique – internally the actuation and structure is quite rigid, yet the motion appears beautifully fluid and animal-like because of very fast impedance controller. However, there are two problems with this simulated compliance. The first is that there are no energetically “free”

motions for the system since the robot must be controlled at all times; a result of the inherent properties of stiff actuators is that the natural, unforced motions of the mechanical system's rigid bodies will always be dominated by actuator dynamics. The second problem is that if the system is turned off it will immediately return to its passive behavior, becoming extremely stiff and vulnerable to damage – from personal discussions HyQ's designer Claudio Semini, the unexpected rigor mortis of HyQ resulting from a mid-flight controller failure has broken a few expensive load cells upon landing.

A second technique to reduce the impedance of a robot's joint or end effector is to introduce an elasticity between the load and the actuator (G. Pratt, Williamson, Dillworth, Pratt, & Wright, 1997). A spring is a perfect impedance and acts as a low-pass filter for the stiff system. If the position of load mass is perturbed by a disturbance, the force in the spring will begin to increase, but because the inertia of the spring is very low this force changes very gradually and proportionally to the size of the deflection. It is certainly a smaller force than would be created from trying to turn the high inertia of an unsprung gear reduction system or hydraulic cylinder. In this way, the addition of a series elastic element decouples the motion of the output mass and the rotor to some extent. Such so-called compliant actuators (Mennitto, 1995) or series-elastic actuators (G. Pratt & Williamson, 1995), are becoming increasingly popular.

As with most engineering decisions, adding an elastic element like a spring is a compromise. Resistance to breaking under sudden impacts and the fidelity of force control at low bandwidths are obviously improved. Often the torque at each joint can be accurately measured by measuring the deflection of the elastic element. On the other hand, the ability of such a system to quickly generate large forces is reduced compared to actuators without compliant elements, because the spring must first be compressed substantially before a large force may be generated. Such bandwidth limitations are well understood (Eppinger & Seering, 1987).

A more difficult problem is created by the increased control complexity: each additional elastic element increases the number of degrees of freedom of the system and makes feedback control more difficult. Naïve feedback to control the position of the load may actually use *more* energy than a stiff actuation system, because work must be done compressing and decompressing the spring to develop control forces. On the other hand, it has also been shown that an appropriate series elastic element can increase the energy that a motor can deliver to a mass by a factor of four for catapulting, throwing, and hopping tasks (Paluska & Herr, 2006). So although the ability of an actuator to regulate high frequency forces may be diminished by the addition of a series spring, the ability of the system to regulate the total energy of the system may not be quite so negatively affected—any energy stored in the spring will inevitably affect the dynamics of the system, although it may not happen immediately.

There is another argument in favor of series elasticity that must be mentioned: in nature elasticity is found everywhere. Muscle fibers and animal tissues have significant elasticity that plays an important role during locomotion (Alexander, 1988, 1990). Elastic models of walking (Geyer, Seyfarth, & Blickhan, 2006) and running (Blickhan, 1989) concisely describe the motion of an animal's center of mass by computing forces generated by a virtual elastic element.

The actuator presented in Chapter 4 uses an electric motor and a nonlinear elastic element in its construction. Electric actuation is uncommon in hopping robots but not unprecedented; it is simply easier for most hopping robots to use pneumatics (Hosoda, Sakaguchi, Takayama, & Takuma, 2010) (Niiyama, Nagakubo, & Kuniyoshi, 2007) or hydraulics (Raibert, 1985) (Zeglin, 1991) (Hyon & Mita, 2002) to achieve the necessary power-to-weight ratios for sustained hopping motions. Relatively few hopping robots incorporate both electric actuation and series elasticity (Ahmadi & Buehler, 2006) (Hurst, 2008) (Rummel, Iida, Smith, & Seyfarth, 2008) (Curran & Orin, 2008). Let us now review some of the many series elastic actuator designs to see where the work of this thesis fits in.

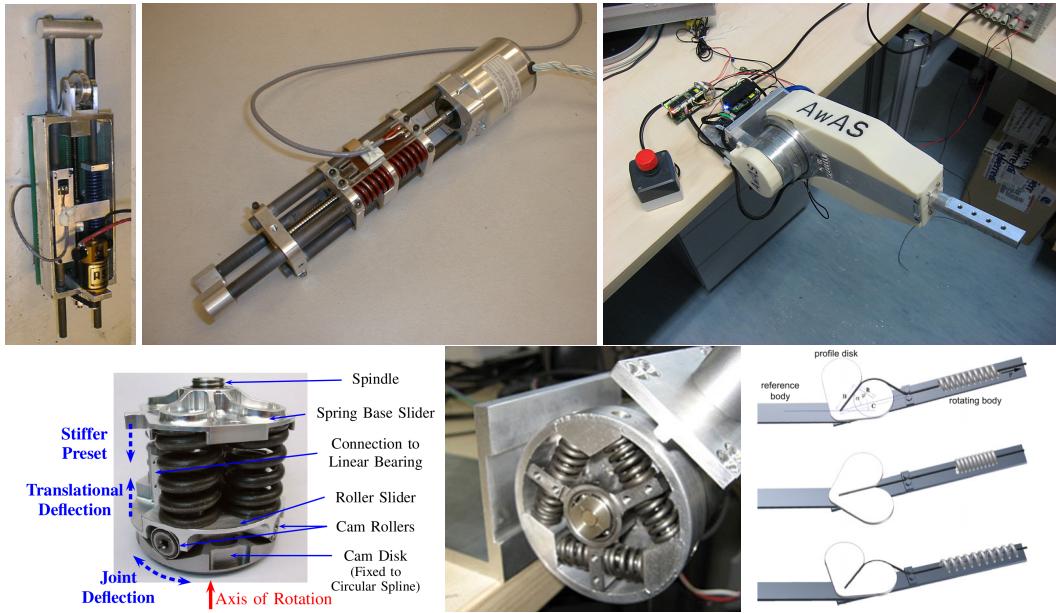


Figure 2.7: Six actuators with series elastic elements. Clockwise from the upper left: The MIT SEA, an electric SEA sold by Yobotics, the AWAS, MACCEPA 2.0, the cCub compliant element configuration, and the DLR VS module.

2.2.3 Series Elastic Actuators

Series elastic actuators have a variety of morphologies for a variety of different purposes (Figure 2.7). For some researchers, the reason to add series elasticity is simply to create a good force-domain actuators. Toward this end, a stiff linear spring is often sufficient. Pratt and Williamson (G. Pratt & Williamson, 1995) used linear-rate compression springs and a ballscrew to precisely control linear forces. The actuators for the robot Twendy-One (Iwata & Sugano, 2009) use “gummetal” torsion bars passing through the rotor to measure torques, an extremely compact geometry. The cCub robot actuators (Ugurlu, Tsagarakis, Spyros-Papastavridis, & Caldwell, 2011) are also very compact and use six small compression springs arranged in a triangle. Although none of these designs allow significant energy storage or would be suitable for a jumping robot, they are small enough for integration in complex robots and the minimal compliance allows improved force control without risking too far a departure from classical control methods.

In medical applications, a very low apparent impedance is important to avoid injuring patients. Accordingly, large, soft springs are used. An example, Sulzer *et al.* developed a compliant knee brace that uses a large torsional spring located at the knee and a power supply mounted on the user’s back (Sulzer, Roiz, Peshkin, & Patton, 2009). This reduced the mass of the mechanism at the knee to just 1.2kg and allowed for a very soft impedance through which to gently interact with the patient.

In the last decade, there has been an explosion of actuators designed to create variable compliance. Many of these designs use two nonlinear elements in an antagonistic configuration to create an adjustable, effectively linear spring (Figure 2.8). Some researchers use the nonlinear stiffness from a nonlinear elastic material such as ball-shaped pieces of rubber (Palli, Melchiorri, Wimböck, Grebenstein, & Hirzinger, 2007) or silicon rubber sponges (Haiya, Komada, & Hirai, 2010). Other researchers use nonlinear elasticities which come from the nature of the actuator itself, such as McKibben air muscles (Tonietti & Bicchi, 2002). Other researchers create

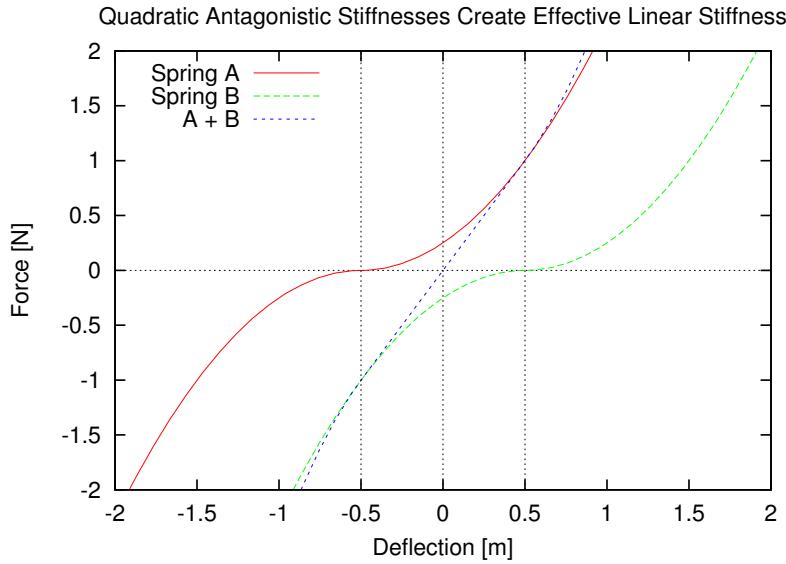


Figure 2.8: Two antagonistic quadratic springs with unit stiffnesses pretensioned by a unit from each other produce an effectively linear spring for deflections between -0.5 and 0.5.

a nonlinear mechanism that stretches a spring at an increasingly fast rate, for example, by using a cable under tension pulled orthogonally to the direction in which it lies (Yamaguchi & Takanishi, 1997; Hyodo & Kobayashi, 1992).

Another way to create an effective variable compliance is to change the geometry of the device. This can be done by using a variable length lever arm to effect different stiffnesses (Schiavi, Grioli, Sen, & Bicchi, 2008; Jafari, Tsagarakis, Vanderborght, & Caldwell, 2010). Another technique is to rely on the orientation of the robot's link itself to create different lever arms, such as was used in MACCEPA (Vanderborght, Tsagarakis, Semini, Van Ham, & Caldwell, 2009). Pulleys of increasing radius such as in the ECL leg (Hurst, 2011) or the MACCEPA II (Vanderborght et al., 2009) are another way to provide a nonlinear stiffness if used in pairs. Still another technique followed by the DLR VS-joint is to use a revolute mechanism and several cams to create an effectively nonlinear spring (Wolf & Hirzinger, 2008). By changing the cam profiles, its nonlinearity may also be easily changed.

Maximizing energetic efficiency is the goal of other researchers. The V2E2 actuator (Gerelli et al., 2009) has few mechanical details published at the moment, although it is described as using an infinitely variable transmission with clutched and unclutched configurations, allowing energy to be stored indefinitely in a mechanical spring and released as needed. They also mention the potential of regenerating energy electrically by using the motor as a generator in conjunction with a reversible amplifier, a capability already present in the actuator to be described in Chapter 4.

Other researchers focus on how compliance relates to safety concerns (Bicchi & Tonietti, 2004). The DM² approach uses a series elastic actuator to produce large, low frequency torques, and a high frequency but lower torque actuator is used in parallel to cover higher bandwidth signals (Zinn, Roth, Khatib, & Salisbury, 2004). The result is a low-impedance actuator which retains good force bandwidth and control of the end effector position.

Actuators with programmable damping have also been developed in the past few decades. Kirsten Laurin-Kovitz created a programmable damping unit using multi-channel fluid valves as well as antagonistically configured nonlinear springs to create a variable effective elasticity (Laurin-Kovitz, Colgate, & Carnes, 1991). More recently, researchers at IIT created a compact design with a variable damping mechanism but a non-adjustable elasticity (Laffranchi, Tsagarakis, & Caldwell, 2010). That controllable damping can provide improved distur-

bance rejection is well known, even in series elastic actuators (Hurst, Rizzi, & Hobbelen, 2004).

Interestingly, with the present focus on variability, few designs appear to be designed specifically towards a particular task's nonlinearity at the moment. The MACCEPA's stiffness curves do not appear to be optimal for the torques produced during hopping, although it is used in the hopping robot CHOBINO (Vanderborght et al., 2009). Similarly, although the actuator presented in (Hutter, Remy, & Siegwart, 2009) possesses nonlinear characteristics, these nonlinearities are purposely non-smooth and contain internal collisions beneficial for stability but not for energetic efficiency or control. For the case of linear actuators, task-specific mechanical design may exist in limited form: (Curran & Orin, 2008) used genetic algorithms to optimize the control and actuator parameters of a series elastic actuator to produce the highest jumping motions for a vertically-constrained robot.

In this thesis, task-appropriate elastic nonlinearity is an explicit goal. The series elastic actuator presented in Chapter 4 was designed specifically for suitability in hopping robots. For this reason, it has superior energy storage capability among comparably sized actuators.

2.3 Centroid Dynamics Control

Successful controllers for dynamic walking, running, and hopping robots all seem to have one thing in common: task space control of the center of mass (CoM). For those unfamiliar with the term, a “task space” is simply some mathematical space in which it is convenient to express the mathematics of a controller or a solution to a particular problem. The kinematic mapping between the joint coordinate space and the arbitrary task space is usually expressed with a Jacobian, which provides an instantaneous relation between the task and joint spaces. If the task space contains fewer dimensions than the joint space, one of the many possible pseudo-inverses can be used to partition the effect of forces in the task space amongst the extra joints (Sentis, 2007).

The fact that the same task space (the CoM space) is given different names by different researchers and obscures the commonality of the work being done. For example:

- The Spring Flamingo walker (J. Pratt, Chew, Torres, Dilworth, & Pratt, 2001) uses a virtual leg model between the CoM and the foot, and computes joint torques from the effect of virtual components on the CoM using Jacobians. Virtual components are used to generate joint torques in an intuitive way, and can be “springs, dampers, dashpots, masses, latches, bearings, potential fields, dissipative fields” – but the most important virtual components often work on the center of mass.
- Resolved Momentum Control (Kajita et al., 2003) explicitly regulates the whole-body angular momentum such that it remains near zero, a form of CoM control.
- Control via centroidal momentum matrix (Orin & Goswami, 2008; Ankarali, Saranli, & Saranli, 2010) is another way to regulate the angular and linear momentum of the CoM.
- The reaction mass pendulum model (Goswami, 2008; S. Lee & Goswami, 2007) maps the structure of any robot onto just a two quantities: the position of the center of mass and the moment of inertia about that mass. A robot of any geometry may thus be analyzed or controlled with only a few quantities of interest.
- The operational space formulation (Khatib, 1987) has been applied to the balancing tasks of humanoid robots (Khatib, Sentis, Park, & Warren, 2004) using the center of mass of the system as the task space. It also focuses on the use of a pseudo-inverse that minimizes the systems kinetic energy to create “dynamically consistent” motion.
- Quadratic programming techniques can be used to take a desired centroidal motion and map it to control torques for the robot (Stephens & Atkeson, 2010), selecting contact forces that do not violate friction

constraints and minimize internal torques when multiple feet contact the ground. A Spring-Loaded Inverted Pendulum (SLIP) model is often used to compute the desired motion of the CoM (Hutter, Remy, Hopflinger, & Siegwart, 2010).

- The controller of the robot “Thumper” (Poulakakis & Grizzle, 2009) specifies the motion of the motor’s rotors using Hybrid Zero Dynamics (HZD). Unfortunately, the unusual notation and terminology used when discussing HZD is unfamiliar to many engineers and perhaps explains why the idea has not gained wider support. It seems to be best described as a task space controller that forces the motion in the task space to follow B-splines, yet the mathematics seem to be getting in the way of this simple statement. The bipedal controllers presented in (Chevallereau et al., 2003) and (Chevallereau, Westervelt, & Grizzle, 2005) also use HZD and are even less accessible.
- Even Marc Raibert’s decades-old 3-part decomposed controller (Raibert, 1985) computes torques for a lightweight leg based on the motion of the heavy body. The body is assumed to be massive enough that the CoM of the whole system can be approximated by considering just this body, so it may be classified as CoM control with some particularly simplifying assumptions about the leg.

In this thesis, we will simply call these “Centroidal Momentum Controllers” because each uses the motion of CoM to generate joint torques that in turn regulate the momentum of the CoM. The hopping robot controller presented in Chapter 5 is of this type of dynamic balancing controller.

2.4 Estimation and System Identification

Estimation is the word used to encompass the topics of filtering, smoothing, and prediction. The problem is well understood by naval navigators and roboticists alike: given some noisy sensor data, we wish to know where we *were* (smoothing), where we *are now* (filtering), and where we *will be* (prediction). By using a parametrized model, we can improve our estimations by selecting the most likely states given our model. *System identification* is essentially the same problem but in reverse: given some noisy sensor data, we try to find the model parameters that would most likely have generated the data that we have seen.

These are both clearly inference problems; we wish to infer what the states of the robot are at different times based on a collection of data and our robot model. Conversely, based on the data we have seen, we also want to infer how we might correct our model parameters to better fit the data that we have seen.

David MacKay argues persuasively in his textbook that nearly every inference problem should be tackled using a Bayesian methodology (MacKay, 2003). Yet for engineering problems with many parameters, a Bayesian analysis techniques that infer model parameters or marginalize across different models can be a daunting task. Not surprisingly, most engineers prefer simpler and computationally efficient methods.

In the classic paper on linear filtering, R. Kalman presented what came to be known as the Kalman Filter, a landmark development in filtering, smoothing, and state estimation problems (Kalman, 1960). Indeed, introductions to the Kalman Filter (KF) may be found in a wide variety of textbooks and reference works (Welch & Bishop, 2001; Levine, 1996; Siciliano & Khatib, 2008) because for almost every sense relevant to linear dynamic systems, the KF can be considered optimal. In a classic paper, Rauch *et al.* showed that for linear dynamic systems with additive gaussian noise, the maximum likelihood estimate is equivalent to least squares, the underlying method used in clever recursive form by the Kalman Filter (Rauch, Tung, & Striebel, 1965). However, the Kalman Filter is not theoretically applicable to nonlinear systems, an important limitation.

For this reason, the Extended Kalman Filter (EKF) was developed, and it remains the most common nonlinear state estimation technique despite an important limitation of the EKF: the first order linearization used in the

EKF poorly estimates the posterior mean and covariance of a gaussian random variable propagated through a nonlinear system (Wan & Van Der Merwe, 2000). The so-called Unscented Kalman Filter (UKF) avoids this assumption by using the “unscented transformation” to better estimate the statistics of a gaussian random variable that undergoes a nonlinear transformation (Wan, Van Der Merwe, & Nelson, 2000). In essence, by sampling five carefully chosen points on the posterior distribution, the UKF can produce an estimate of the posterior mean and covariance that are accurate to the third order, for *any* nonlinearity. For modern computers, the additional sampling yields sufficiently better estimates that it is worth the slightly increased computational burden – although the EKF and UKF are of the same order of computation, the UKF will of course have a larger coefficient.

But perhaps this is overstepping the current state of the art as applied to legged robots. Despite such interesting advances in the theory of estimation and system identification, the mechanical parameters of even advanced bipedal robots such as MABEL are still identified using somewhat ad-hoc, offline techniques (Park, Sreenath, Hurst, & Grizzle, 2011). As MABEL lacks force, torque, and acceleration sensors at many locations, determining inertial parameters and friction is not a trivial task. However, the deep link between linear filtering and forward and inverse dynamics in spatial vector notation should make the task tractable (Rodriguez, 1987).

At present, robot researchers performing the most detailed system identification are typically studying robot collision detection. Without a good model of the robot, it is impossible to detect small discrepancies between predicted and actual motion that signal the presence of an external force or impact (De Luca, Albu-Schaffer, Haddadin, & Hirzinger, 2006).

Use of model-based inference techniques for state and maximum-likelihood model parameter estimation methods are the subject of Chapter 6. There is one additional detail that must be mentioned regarding state estimation: ensuring conservation of energy. In a system without full state estimation, it is easy to create an observer that appears to give reasonable values for the state of the system. Yet, when these same values are used to compute the total energy of the system, it may vary significantly from one state to the next. Without the addition of power constraints, it can be difficult to account for all instantaneous power flows in a complex system with both electrical and mechanical components. Thankfully, engineering analysis techniques are already available for performing multi-domain modeling of power flows: bond graphs and a port-Hamiltonian formulation of the system are an excellent way to discover power constraints and monitor the flow of energy between components. Excellent introductions to the field are widely available (Schaft, 2006; Visser, Carloni, Unal, & Stramigioli, 2010).

CHAPTER 3

Simultaneous Design of Mechanical and Controller Dynamics

One hallmark of a good scientific theory is consistency; the best theories predict the same results regardless of the observer or reference frame. Similarly, a trait of good engineering is harmony; the best works of engineering have an elegant compromise of functionality in which components work together to accomplish a task rather than fight against each other. But how can we arrive at such a design?

This chapter establishes the framework for the rest of the book by describing the design process used for the robot JIMI, a novel, planar monopod hopping robot. The name JIMI is a recursive acronym that playfully stands for “JIMI Is My Invention”, but was mostly chosen to give the author a good excuse to listen to Jimi Hendrix’s music while he works.

The most important design goal for JIMI was that *the end-result dynamic behavior of the robot be a result of harmoniously integrated systems*; there should be an appropriate balance between dynamics achieved through passive mechanical properties and dynamics achieved through controlled actuation. Since a hopping robot is a single-purpose machine, the mechanical design can be optimized such that it hops in a nominally stable way and energy must only be expended by the controller to resist destabilizing forces on the robot.

In other words, the design of the mechanical system should be such that the unforced dynamics of the robot in the environment will be sufficiently similar to the dynamics of jumping that only minor control actions will be required to stabilize the robot. The controller must of course be aware of these dynamics, and not attempt to fully control the mechanical motion to stabilize the robot. We are *guiding* a robot’s natural motion in such a case, not *proscribing* it with a single trajectory.

For passive dynamic walkers, an analogy is often made to aircraft: one way to design efficient powered flying machines is to focus first on making efficient gliders and only later add a propulsion system. In the case of JIMI, the goal was for the mechanical dynamics of the robot to be close to hopping, and for the controller to complement these natural dynamics but provide stability. In practice, making the mechanical system work harmoniously with the proposed controller involved some guessing, model-based simulation, and lots of iteration.

For this reason, it is difficult to organize the presentation of the mechanical dynamics, actuator dynamics, and control dynamics, since the design decisions presented in Chapters 3, 4, and 5 were all performed at roughly the same time. However, a line must be drawn somewhere, and so this chapter focuses upon the mechanical dy-

namics of hopping, its implications on the actuation system, and initial dynamic estimates taken from numerical simulations.

The design of the a custom actuation system for JIMI requires more discussion than can be fit comfortably in this chapter, and will be presented in Chapter 4. This custom actuator is called the “HypoSEA”, short for “Hypocycloid-based Series Elastic Actuator”, which describes the particular mechanism used to stretch a series elasticity in a nonlinear manner but unfortunately does not provide an excuse to listen to any particular genre of music.

The design of the balance controller dynamics was complex enough that it also deserves more space than just this chapter, and so is the subject of Chapter 5.

We will now turn to the design goals for JIMI to elucidate why certain decisions were made, then later study its model in depth via simple analysis, and finally view numerical simulation results used to develop harmony between actuator, controller, and mechanical dynamics. For comparison purposes, the reader may later to compare the contents of this chapter with Chapter 8, which describe the robot as it was subsequently realized.

3.1 Design Goals for JIMI

Somewhat arbitrarily, the initial design of JIMI began with a list of researcher desires:

1. The robot should be a planar monopod with revolute joints, and the inertia of the leg should be non-negligible compared to the mass of the robot body. This is should prevent the use of controllers which assume that leg inertia is a small fraction of the torso’s inertia, and is deliberate in order to further the development of hopping controllers for robots of more general morphologies.
2. The robot should stand approximately 120cm tall, be largely self-contained, and weigh roughly 10kg including batteries and on-board computer.
3. The construction of the robot should be robust enough that it will survive many uncontrolled falls from a height of 50cm.
4. For simplicity, it should have just two actuated revolute joints (hip, knee), and three links (torso, thigh, and shank).
5. For energy efficiency, there should be elasticity in each joint capable of regularly storing and releasing enough energy for the robot to hop in a purely mechanical manner.
6. It should use electric actuation for clean and safe indoor operation.
7. The passive mechanical dynamic behavior of the robot should be similar to hopping.
8. The actuation system should be very backdrivable, for good force control and easy modeling.
9. It should feature a biarticulate actuator to provide either stiffening effects or to transfer energy from one joint to the other.
10. The test platform used to planarize the motion of the robot should not interfere significantly with the robot’s dynamics.

In retrospect several of these goals seem naively chosen. Making a robot strong enough to survive many uncontrolled falls from 50cm is a terrifyingly difficult challenge – it would be better to catch the robot or avoid

falling entirely. The idea to use biarticulate actuation was in fact dropped completely to reduce system complexity, although there is evidence that biarticulate actuators can improve jumping heights (Babic, Lim, Omrcen, Lenarcic, & Park, 2009). A heavy leg is bad for efficiency and also for control – even if it is more academically interesting, it also makes life more difficult for the researcher. Most regrettably, the author wishes that an additional design goal had been given consideration: to reduce the mass and inertia of not just the link immediately contacting the ground (say, the “foot”) but also its immediate neighbor (the “shank”), in order to better decouple the destabilizing effect of shocks on the more massive parts of the robot. Another regret is not fully considering the energetic requirements of holding torques on the rotor – even holding the rotor motionless consumes electrical energy if the transmission is easily backdriven – although in Section 4.6.4 we will see that energy regeneration techniques may alleviate the significance of this problem.

The decision to use revolute joints is also important. There is evidence that a revolute jointed leg may have larger margins of passive mechanical stability than a spring loaded inverted pendulum, especially if nonlinear springs are used (Rummel & Seyfarth, 2008). It has also been found that nonlinear springs improve the disturbance rejection of a mechanical running system (Karssen & Wisse, 2011). On the other hand, simple models of hopping suggest that a decreasing effective leg stiffness stabilizes a hopping motion, so using a nonlinear spring perfectly suited to creating a linear ground reaction force may not be ideal (Riese & Seyfarth, 2012). The debate is clearly not settled yet.

Returning to the narrative, the above design goals naturally lead to decisions about what tradeoffs would be made for the robot. Although we might desire our robot be arbitrarily light or powerful, when faced with a realistic engineering problem we must prioritize certain quantities at the expense of others.

For JIMI, the choice of electric actuation became extremely significant because it limited the total power-to-weight ratio achievable in the robot. The high torque frameless Brushless Direct Current (BLDC) electric motors used in JIMI (See Chapter 4) weigh $\sim 900\text{g}$ with a simple enclosure and are rated for only 111W of continuous power. Although human muscle is of comparable power density at roughly 100W/kg (McMahon, 1984), an electric motor requires a transmission to develop useful levels of torque. The resulting low power density meant that maximizing energetic efficiency would be crucial to the operation of the robot. It also meant that minimizing the structural weight of the robot would be important and lead to the use of fiber reinforced composite materials, more fully described in Appendix ??.

We now turn to a simple theoretical model of JIMI which was developed to more precisely estimate mechanical design criteria needed to satisfy the engineering goals outline above.

3.2 Sagittal Plane Model of JIMI

The planar robot model used to analyze JIMI consists of three links, two actuated revolute joints, and a point contact foot assumed to be an unactuated revolute joint. We use the convention that angles are measured counterclockwise from the vertical, and that subscripts s, t, b, f, c refer to the shin, thigh, body, foot tip, and mass centroid, where applicable. The model is shown in Figure 3.1, and displays three rigid bodies (called the “body/torso”, “thigh” and “shank”), their masses (m_b, m_t, m_s), moments of inertia (I_b, I_t, I_s), joint angles ($\theta_b, \theta_t, \theta_s$), and some lengths (a, b) for each body expressing the total length of each link ($l_t = a_t + b_t$) as well as the position of the CoM along each link, which are represented by black dots. For modeling simplicity, the center of mass of each body is assumed to be on the line connecting its two joints. When the robot is in the air, the position of the foot must also be specified with two cartesian coordinates (x_f, y_f) .

Mathematical details of this model will be described in section 3.4 when its forward dynamics are simulated, impact impulses are calculated, and the effects of friction, elasticity, and motor torques are considered. For now, it is sufficient to note that this is an extremely simple model with only two degrees of actuation, yet three degrees

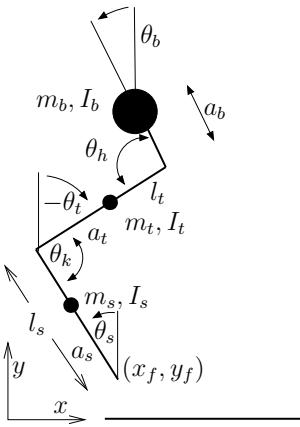


Figure 3.1: Model of the robot JIMI, showing the torso, thigh, and shank links. A black dot indicates the center of mass for each link.

of freedom during stance phase. The robot is therefore underactuated during stance. During the flight phase the robot will move ballistically and be completely unable to regulate its centroidal angular or linear momentum.

The primary source of energy loss in a jumping robot is typically from the foot's impact with the ground, which we assume will be nearly instantaneous and inelastic. To reduce ground impact energy losses, it is desirable to minimize the “unsprung” weight of the robot, defined as any mass whose kinetic energy during flight cannot be stored in an elastic energy storage device during stance phase. Minimizing the unsprung weight of a leg improves efficiency, as unsprung mass directly experiences the impact with the ground and some fraction of its kinetic energy is lost forever. On the other hand, the kinetic energy of a sprung weight may be converted into potential energy and released later in a stride with minimal net energy loss.

3.3 Rough Estimates of Mechanical Dynamics

This section documents the process by which the dynamics JIMI were selected, and what actuation requirements were entailed by these dynamics. Using the simple model described in the previous section and some basic assumptions, we can begin to estimate the mechanical engineering requirements for a hopping robot. This is not intended to be extensive or particularly accurate analysis, merely documentation of the first steps taken during the engineering process.

Let us begin by asking several simple questions to get a rough idea of what the dynamics of a hopping robot will be. Considerations of energetics will be at the forefront, as the power density that an electric motor can deliver is extremely limited when compared to hydraulic or pneumatic systems.

3.3.1 What are the rough vertical motions of a hopping robot?

Running humans have a step frequency of roughly two hertz (Alexander, 2003). If we arbitrarily accept this value as typical for a dimensionally similar robot and assume that the robot spends about half its time in the air, the duration of the stance and flight phases of the robot will be $T_F = T_S = 0.25$ seconds. This corresponds to approximately eight centimeters of vertical displacement of the center of mass (CoM) during flight phase.

$$y_{cF} = \frac{1}{2}g \left(\frac{T_F}{2} \right)^2 = 0.0765\text{m} \quad (3.1)$$

From animal running studies we find that the vertical component of the ground reaction force profile (GRF) resembles half a sine wave (Alexander, 1990). This corresponds to a perfect spring-mass system and has led to the development of many spring-based models of running (Blickhan, 1989).

If we assume such a sinusoidal GRF is typical of the dynamics of hopping and should be used as a template, how much will the center of mass move vertically during stance? Assuming that the velocity at which the robot touches down v_{TD} is equal to the speed at which it lifts off the ground again $T_S = 0.25$ seconds later, and that the force on the ground is sinusoidal, we may write the force the foot exerts on the ground¹ as

$$F_f(t) = F_{peak} \sin\left(\frac{\pi}{T_S} t\right) \quad (3.2)$$

where F_{peak} is the peak ground reaction force. This peak force must be large enough so that the total impulse applied to the ground during stance I_S is equal to the impulse due to gravity I_g over the entire hopping cycle. The latter is easily written:

$$I_g = mg(T_S + T_F) \quad (3.3)$$

A sinusoidal force profile during stance creates an impulse

$$I_S = \int_0^{T_S} F_{peak} \sin\left(\frac{\pi}{T_S} t\right) t dt = \frac{2F_{peak} T_S}{\pi} \quad (3.4)$$

Equating (3.3) and (3.4), we can solve for the maximum force F_{peak} as a function of the stance and flight times:

$$F_{peak}(T_S, T_F) = \frac{I_g \pi}{2T_S} = \frac{\pi mg(T_S + T_F)}{2T_S} \quad (3.5)$$

For a 10kg robot hopping at 2Hz with 50% duty cycle ($T_S = T_F = 0.25$ s), equation (3.5) predicts 308N of peak force. This force may also be normalized by mg and expressed as π bodyweights (Gees) of force on the foot. A peak force of about three times body weight is in fact a reasonable value for a moderately sized animal running at slow speeds (Alexander, 2003).

If we normalize (3.5) by body weight mg and hopping period ($T_S + T_F$), the expression for the peak vertical GRF as a function of stance duty cycle $\psi = \frac{T_S}{T_S + T_F}$ becomes simple:

$$F_{peak}(\psi) = \frac{\pi}{2\psi} \quad (3.6)$$

Figure 3.2 shows this graphically. As the duty cycle becomes shorter and shorter, the peak forces get increasingly large, an intuitive result. Conversely, as the stance duty cycle approaches 1.0 and the flight phase just barely disappears, the peak force required is just $\pi/2$ times the body weight.

Let us now return to the non-normalized motion of the CoM during stance. We may find its vertical motion by summing the forces on it, integrating twice, and using suitable integration constants.

¹The interaction force between the foot and ground only has one physical solution, so the sign of the GRF is disregarded in this thesis. Forces on the foot tip and the GRF are therefore equivalent.

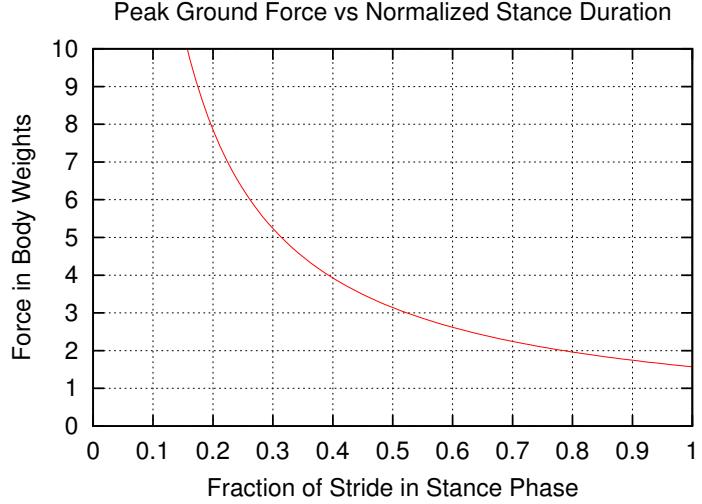


Figure 3.2: Peak ground reaction forces of a normalized springy hopping robot as a function of the stance duty cycle.

$$\begin{aligned}
 \ddot{y}_{cs}(t) &= \frac{F}{m} = \frac{F_e + F_g}{m} \\
 &= \frac{F_{peak}}{m} \sin\left(\frac{\pi}{T_s} t\right) - \frac{mg}{m} \\
 \dot{y}_{cs}(t) &= \frac{F_{peak}}{m} \int \sin\left(\frac{\pi}{T_s} t\right) dt - \int g dt \\
 &= -\frac{F_{peak} T_s}{m\pi} \cos\left(\frac{\pi}{T_s} t\right) - gt + \left(v_{td} + \frac{F_{peak} T_s}{m\pi}\right) \\
 y_{cs} &= -\frac{F_{peak} T_s^2}{m\pi^2} \sin\left(\frac{\pi}{T_s} t\right) - \frac{1}{2}gt^2 + \left(v_{td} + \frac{F_{peak} T_s}{m\pi}\right) t
 \end{aligned} \tag{3.7}$$

Assuming that $m = 10\text{kg}$, $T_s = T_F = 0.25$ and $v_{LO} = v_{TD} = \frac{1}{2}g T_F = 1.23\text{m/s}$, the total vertical travel of the robot during stance can be found to be

$$y_{cs}(0) - y_{cs}\left(\frac{T_s}{2}\right) = -0.119$$

In conclusion, there is roughly 12cm of vertical travel during a $T_s = 0.25\text{s}$ stance phase, and roughly 8cm of vertical travel during a $T_F = 0.25\text{s}$ flight phase, for a total of slightly less than 20cm of vertical motion. This is true for any robot or animal exhibiting a sinusoidal GRF and hopping at 2Hz with 50% stance duty cycle.

3.3.2 How much energy is stored and released each hop?

If the robot weighs $m = 10\text{kg}$ and the CoM moves up and down $h = 20\text{cm}$ each hop, the change in potential energy is

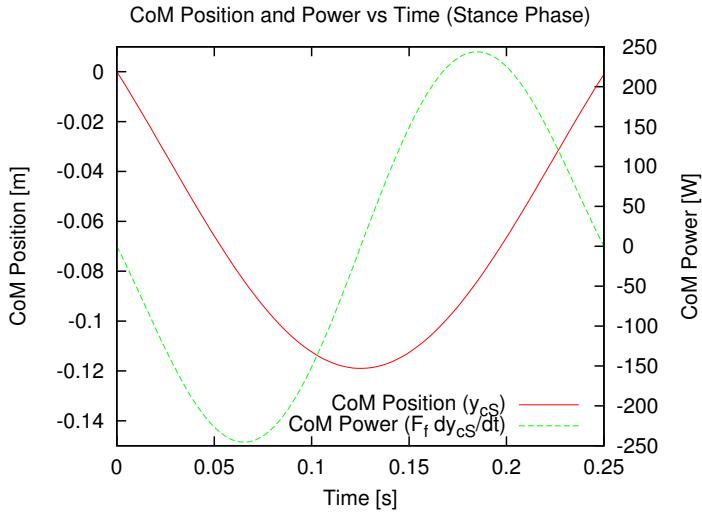


Figure 3.3: Sinking motion of a 10kg robot's CoM during a 0.25s stance and the instantaneous power required to follow this motion.

$$\Delta E = mgh = 10(9.8)0.2 = 19.6J \quad (3.8)$$

Thus approximately 20J of energy needs to be stored and released twice per second during hopping. There is more to say about energetics than just this, however.

The true mechanical power acting on the robot to bounce it up and down is given by the velocity \dot{d} of the robot's CoM multiplied by the ground reaction force F_f . This quantity is plotted in Figure 3.3 along with the CoM motion during stance computed from (3.7) in the previous section. We can see that the actual instantaneous rates of work are between -250W and +250W.

The instantaneous power requirements, when averaged out, suggest that the motors of the robot must be able to dissipate the heat energy created when performing 158W of mechanical work continuously. However, it is obvious that a large amount of negative work is done on the robot's CoM during the first half of stance, and large amount of positive work in the second half. If we had a way of mechanically storing and releasing 20J of energy, the amount of mechanical work that would need to be done by the rotor could be dramatically reduced.

The reader may anticipate that large springs will indeed be added to JIMI's actuators for precisely this reason.

3.3.3 What are the joint torque and velocity requirements during stance?

If all the joints of the robot will use the same type of actuator for engineering simplicity, we must design the actuators at the joints to satisfy the worst case constraints. This means that we must consider only the most distal joint, the "knee" joint in the case of JIMI, because it will experience the fastest impacts and largest torques of all the joints in the robot. With the stereotypical vertical motion of the CoM established previously by equation (3.7), we should be able to compute the torque and angular velocity at the knee.

Unfortunately, precise estimates of joint torques are greatly dependent upon the robot posture and morphology, so we will simplify our hopping robot model further by assuming that the hip is directly above the foot and the thigh and shank are the same length l , such that the knee joint does all the work. This configuration is shown in Figure 3.4 and is assumed to be the worst case scenario for computing knee torques or velocity.

We can write the torque at the knee τ_k as a function of the knee's interior angle θ_k :

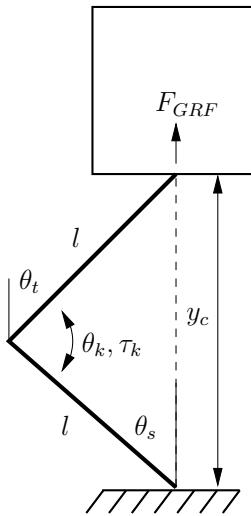


Figure 3.4: Simplified two-link leg model for estimating knee torques in a worst-case jumping configuration. The leg is assumed to be massless and the body positioned directly above the foot.

$$\tau_k(\theta_k) = F_f l \cos\left(\frac{\theta_k(t)}{2}\right) \quad (3.9)$$

Describing how the knee will bend requires that we define the initial posture of the robot. If the robot starts with a relatively straight leg, the angle traced out by the knee will be different than for the case with a bent knee – even assuming that the GRFs are identical. If we use the expression for the motion of the CoM from equation (3.7), the knee angle θ_k is a function of the CoM height $y_c(t)$ and the CoM height at touchdown y_{cTD} . The knee angular velocity $\dot{\theta}_k$ is also a function of the touchdown velocity \dot{y}_{cTD} .

$$\theta_k(y_{cS}(t), y_{cTD}) = 2 \sin^{-1} \left(\frac{y_{cS}(t) + y_{cTD}}{l} \right) \quad (3.10)$$

$$\dot{\theta}_k(y_{cS}(t), y_{cTD}, \dot{y}_{cTD}) = \frac{2 \dot{y}_{cTD}(t)}{\sqrt{1 - \left(\frac{y_{cS}(t) + y_{cTD}}{l} \right)^2}} \quad (3.11)$$

For some knee angle at touchdown θ_{kTD} , the height of the CoM relative to the floor is:

$$y_{cTD} = 2l \sin\left(\frac{\theta_{kTD}}{2}\right) \quad (3.12)$$

Of course, $0 < y_c < 2l$ must always be true for all this to make physical sense; we can't have a leg lifting the CoM higher than the leg length or allow the CoM to go below ground.

Now let $m = 10\text{kg}$ and $l = 0.4\text{m}$. A graph of the torque-angle relationship for various values of θ_{kTD} are shown in Figure 3.5, and the torque-velocity curves in Figure 3.6.

Looking at Figure 3.5, we can see that the total deflection of the leg changes depending on the straightness of the knee at impact. For an almost straight leg ($\theta_{kTD} = 3.0$), there is a deflection of slightly more than a radian and the torques required to create the desired GRF are nonlinear enough that they resemble a quadratic curve. For a knee that starts more bent at impact ($\theta_{kTD} = 1.0$), the necessary deflection is perhaps only half a radian,

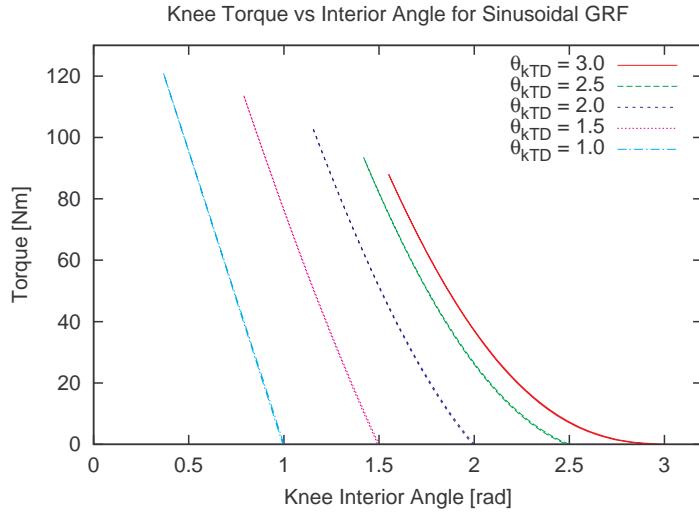


Figure 3.5: The torque-angle relation required for a hopping robot to create a sinusoidal F_f , starting from various touchdown knee angles. The areas under each curve are equal.

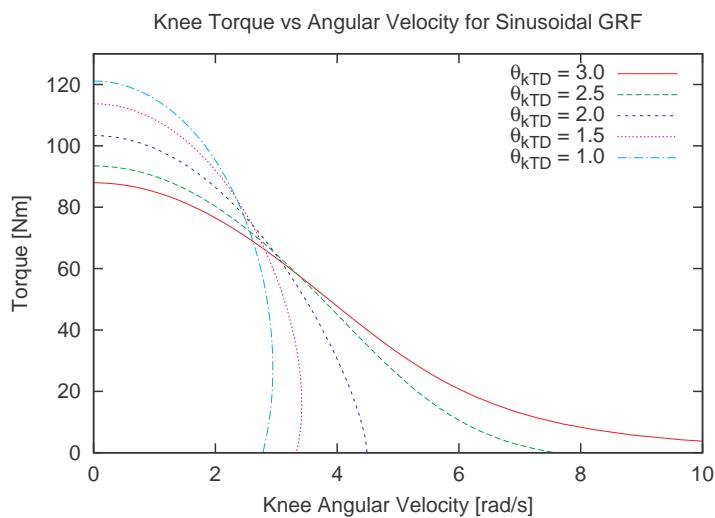


Figure 3.6: Torque-velocity relation for a hopping robot to create a sinusoidal F_f , starting from various touchdown knee angles..

with correspondingly higher torque requirements that are more linear in nature. We also observe that to produce identical ground reaction forces and CoM motion, landing with a straighter leg ($\theta_{kTD} = 3.0$) results in lower peak torques but more nonlinearity. Conversely, landing with a bent leg results in less nonlinearity but higher necessary torques. Keep in mind that all of these starting postures produce the same CoM motion referenced from the CoM height at the instant of touchdown.

From Figure 3.6, we arrive at the intuitive conclusion that a straighter leg will result in initially very high angular velocities during stance, due to the nature of the mechanical advantage a jointed leg such as this provides. It is easy to see from this figure that the maximum joint torques will be between about 90 and 120Nm, and the maximum angular velocity may exceed 10rad/s.

Choosing the 'best' landing posture will depend on other considerations and cannot be made at this time. The choice of leg straightness and its effect on torque-velocity requirements will need to be carefully balanced against the effect of a straight leg on inelastic collision energy loss considerations, as well as the spring stiffness and (non-)linearity. Later, using a numerical simulation of the robot, we will revisit these angle-velocity requirements and their feasibility with the chosen motors. We wish to ensure that the motors are capable of sufficient acceleration to achieve the angular velocities and torques computed above.

Interestingly, in the last figure of paper by Hutter *et al.*, a torque-deflection curve is drawn for the knee joint that closely resembles Figure 3.5 of this thesis (Hutter *et al.*, 2010). They note also that an appropriate nonlinear spring could be used to reduce energy consumption.

3.3.4 Rough Analytical Conclusions

The proceeding rough analysis has in part specified the centroidal dynamics of a hopping robot or animal, as well as a rough estimate of the actuation requirements. Let's review what we have assumed or computed, based on the assumptions of a 10kg robot with a thigh and shank 0.4m long, hopping at about two hertz, with 50% duty cycle and a controller which produces approximately sinusoidal ground reaction forces.

- The robot's CoM will travel about 8cm vertically in the air, and about 12cm during stance.
- The peak ground reaction force will be slightly more than 3 times body weight.
- The worst-case motor torques will be between 80-120Nm, depending on landing configuration.
- The worst-case angular velocities will be greater than 10rad/s.
- The instantaneous mechanical power of the entire system must be greater than 250W.
- A large elastic spring capable of storing 20J of energy at a joint could perform the energy storage and release with high efficiency.
- The linearity of the torque-angle relationship at the joints ranges from approximately linear ($\theta_{kTD} = 1.0$ case) to approximately quadratic ($\theta_{kTD} = 3.0$ case).

Conservative engineering practice demands that each actuator be capable handling the worst case. Our goal should be to select an actuator capable of 120Nm peak torque, 250W peak power, 150W continuous power, and greater than 10rad/s angular velocity. If possible, we should also incorporate large springs capable of storing >20J. In order that the springs not bottom out if the robot accidentally lands too hard, and also because a hopping robot is expected to bounce hundred of thousands of times during its life, it seems reasonable to estimate that the elastic elements in each joint are able to store 30J of energy at maximum.

This actuation specification led to the development of the HypoSEA actuators presented in Chapter 4, which were specifically designed to provide 120Nm torque, store 30J of energy in a series spring, and possess an adjustable nonlinearity ranging from approximately linear to approximately quadratic.

3.4 Numerical Simulation of Mechanical Dynamics

We now turn from the simplified analytic equations used in the previous section to a brute-force computational approach. By numerically simulating the mechanical dynamics of JIMI and the effect of its controller, we may not only visually inspect the behavior of the controller and robot, but also estimate energy losses to friction or collisional effects. The goal is to take the design estimates a step farther and develop an engineering coherence between the mechanical design estimated from CAD models, the controller behavior, and expected dynamics of the robot. By being able to monitor various physical quantities before anything is built, the mechanics, controller, and desired dynamics may be adjusted until everything seems physically coherent and plausible.

While a numerical simulation does not guarantee the robot will behave exactly as desired, it can be an invaluable tool during the design process that illuminates many of the important design issues before a prototype is built. Successful controllers developed in simulation may also be adapted from simulation to the real robot with only minor adjustments.

3.4.1 Equations of Motion

The equations of motion for the three-link rigid body model of Figure 3.1 were derived using the standard Euler-Lagrange formulation.² If we select the generalized coordinates $\mathbf{q} = [\theta_s \quad \theta_t \quad \theta_b \quad x_f \quad y_f]^T$, we can express the kinetic energy T and potential energy V of the n links as functions of the generalized coordinates and velocities.

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \sum_{i=1}^n (m_i v_i^2 + I_i \omega_i^2) \quad (3.13)$$

$$V(\mathbf{q}) = \sum_{i=1}^n m_i g y_i \quad (3.14)$$

where m is the mass of the i th rigid body, I its moment of inertia, v the velocity vector of its center of mass, y the vertical height of its center of mass, and ω its angular velocity about its center of mass. The Lagrangian $L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q})$ can then be used to derive the equations of motion using

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = 0 \quad (3.15)$$

The resulting 2nd order system of equations is rather inconvenient to directly use with numerical integration techniques, so we rearrange the resulting terms into the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = 0 \quad (3.16)$$

²More sophisticated ways of simulating the motion of rigid body systems will be presented in Chapter 7. However, when this preliminary simulation was done the author had not yet developed a familiarity with that material.

where $\mathbf{M}(\mathbf{q})$ is the inertial matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ contains the coriolis and centripetal terms, and $\mathbf{g}(\mathbf{q})$ the gravitational terms. We now list these matricies in full:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{12} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} \\ m_{14} & m_{24} & m_{34} & m_{44} & m_{45} \\ m_{15} & m_{25} & m_{35} & m_{45} & m_{55} \end{bmatrix} \quad (3.17)$$

$$\begin{aligned} m_{11} &= (b_s^2 + 2a_s b_s + a_s^2)m_t + a_s^2 m_s \\ &\quad + (b_s^2 + 2a_s b_s + a_s^2)m_b + I_s \\ m_{12} &= ((a_t b_s + a_s a_t)m_t \\ &\quad + ((b_s + a_s)b_t + a_t b_s + a_s a_t)m_b) \cos(\theta_t - \theta_s) \\ m_{13} &= (a_b b_s + a_b a_s)m_b \cos(\theta_s - \theta_b) \\ m_{14} &= (-b_s + a_s)m_t + a_s m_s - (b_s + a_s)m_b \cos\theta_s \\ m_{15} &= (-b_s + a_s)m_t + a_s m_s - (b_s + a_s)m_b \sin\theta_s \\ m_{22} &= a_t^2 m_t + (b_t^2 + 2a_t b_t + a_t^2)m_b + I_t \\ m_{23} &= (a_b b_t + a_b a_t)m_b \cos(\theta_t - \theta_b) \\ m_{24} &= (-b_t + a_t)m_b + a_t m_t \cos\theta_t \\ m_{25} &= (-b_t + a_t)m_b + a_t m_t \sin\theta_t \\ m_{33} &= a_b^2 m_b + I_b \\ m_{34} &= -a_b m_b \cos\theta_b \\ m_{35} &= -a_b m_b \sin\theta_b \\ m_{44} &= m_t + m_s + m_b \\ m_{45} &= 0 \\ m_{55} &= m_t + m_s + m_b \end{aligned} \quad (3.18)$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & c_{12} & c_{13} & 0 & 0 \\ -c_{12} & 0 & c_{23} & 0 & 0 \\ c_{13} & -c_{23} & 0 & 0 & 0 \\ c_{14} & c_{24} & c_{34} & 0 & 0 \\ c_{15} & c_{25} & c_{35} & 0 & 0 \end{bmatrix} \quad (3.19)$$

$$\begin{aligned}
c_{12} &= ((-a_t b_s - a_s a_t) m_t + ((-b_s - a_s) b_t \\
&\quad - a_t b_s - a_s a_t) m_b) (\dot{\theta}_t) \sin(\theta_t - \theta_s) \\
c_{13} &= (a_b b_s + a_b a_s) m_b (\dot{\theta}_b) \sin(\theta_s - \theta_b) \\
c_{21} &= ((a_t b_s + a_s a_t) m_t + ((b_s + a_s) b_t \\
&\quad + a_t b_s + a_s a_t) m_b) (\dot{\theta}_s) \sin(\theta_t - \theta_s) \\
c_{23} &= (a_b b_t + a_b a_t) m_b (\dot{\theta}_b) \sin(\theta_t - \theta_b) \\
c_{31} &= (-a_b b_s - a_b a_s) m_b \dot{\theta}_s \sin(\theta_s - \theta_b) \\
c_{32} &= (-a_b b_t - a_b a_t) m_b (\dot{\theta}_t) \sin(\theta_t - \theta_b) \\
c_{41} &= ((b_s + a_s) m_t + a_s m_s \\
&\quad + (b_s + a_s) m_b) \sin \theta_s (\dot{\theta}_s) \\
c_{42} &= (a_t m_t + (b_t + a_t) m_b) \sin \theta_t (\dot{\theta}_t) \\
c_{43} &= a_b m_b \sin \theta_b (\dot{\theta}_b) \\
c_{51} &= ((-b_s - a_s) m_t - a_s m_s \\
&\quad + (-b_s - a_s) m_b) \cos \theta_s (\dot{\theta}_s) \\
c_{52} &= ((-b_t - a_t) m_b - a_t m_t) \cos \theta_t (\dot{\theta}_t) \\
c_{53} &= -a_b m_b \cos \theta_b (\dot{\theta}_b)
\end{aligned} \tag{3.20}$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} ((-b_s - a_s) g (m_t + m_b) - a_s g m_s) \sin \theta_s \\ ((-b_t - a_t) g m_b - a_t g m_t) \sin \theta_t \\ -a_b g m_b \sin \theta_b \\ 0 \\ g m_t + g m_s + g m_b \end{bmatrix} \tag{3.21}$$

Equation (3.16) described the unforced motion of the system. To add controls and interaction forces, we now introduce a vector $\tau = [\tau_s \ \tau_t \ \tau_b \ F_{fx} \ F_{fy}]^T$ that expresses the sum of nonconservative torques on the robot, and write

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau \tag{3.22}$$

We can then solve for $\ddot{\mathbf{q}}$ using simple linear algebra to find the forward dynamics of the system.

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q}) [\tau - \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})] \tag{3.23}$$

3.4.2 Numerical Integration

To integrate equation 3.23 numerically with a computer, it will be most convenient if we express it in the form $\dot{\mathbf{x}} = f(\mathbf{x}, t)$, where $\mathbf{x} = [\mathbf{q} \ \dot{\mathbf{q}}]^T$ is the state vector. Writing $f(\mathbf{x}, t)$ is remarkably straightforward since we have done all the work already in the previous section. The following line should be self-explanatory.

$$\begin{aligned}
f(\mathbf{x}, t) &= \frac{d}{dt} \mathbf{x} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \\
&= \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{q}) [\tau - \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})] \end{bmatrix}
\end{aligned} \tag{3.24}$$

We can now apply a common numerical integration technique such as 4th order Runge-Kutta to find the next state \mathbf{x}_{n+1} at time Δt after the current state \mathbf{x}_n .

$$\begin{aligned} k_1 &= f(\mathbf{x}_n, t) \\ k_2 &= f\left(\mathbf{x}_n + \frac{k_1}{2}, t + \frac{\Delta t}{2}\right) \\ k_3 &= f\left(\mathbf{x}_n + \frac{k_2}{2}, t + \frac{\Delta t}{2}\right) \\ k_4 &= f(\mathbf{x}_n + k_3, t + \Delta t) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

This equation has an elegant symmetry to it; we now can easily see how the next state, \mathbf{x}_{n+1} is a sum of the current state \mathbf{x}_n plus some weighted averages of the derivatives, with heavier weights for derivatives computed at the midpoint of each timestep (*i.e.* at $t + \frac{\Delta t}{2}$).

One way to check that the equations are self-consistent is to perform the numerical integration and verify that the total energy $E = T + V$ is nearly constant when no nonconservative forces are used (*i.e.* $\tau = 0$). With a moderately small integration timestep of $\Delta t = 0.002$ s, the energy variations are negligible ($\leq 10^{-12}$ J per timestep) for the model we are studying.

3.4.3 Joint-Link Connection Incidence Matrix

Motors cannot apply torques such as τ_s directly to a link unless they are fixed to the world in some way. In other words, it is nonphysical for a motor to apply a torque just on the shank; real internal robot motors create a force in one link and a reaction force in the opposite direction in another link – that's just Newton's second law. We therefore define an incidence matrix \mathbf{D}_j to transform torques at the joints (τ_j) to torques on the links of the robot.

$$\tau = \mathbf{D}_j^T \tau_j + \dots \quad (3.25)$$

$$\mathbf{D}_j = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{bmatrix} \quad (3.26)$$

In equation (3.25) the “...” indicates that the equation isn't quite complete; there will be other nonconservative forces acting on the system. We will discuss ground contact forces in just a moment for precisely this reason.

3.4.4 Types of Joint Torques

Only two types of non-conservative torques were considered in this simulation: viscous joint friction torques (subscript v) and dynamic balancing controller joint torques τ_j . The method by which controller joint torques τ_j were computed requires a rather lengthy discussion, and so will be presented in Chapter 5. For the time being, the reader is encouraged to abstract away the joint torques as being whatever torques are necessary to create a vertical GRF that is sinusoidal, plus a little extra to stabilize the posture of the robot.

The effect of viscous friction may be described briefly. Viscous friction on the joints was modeled as a force proportional to the velocity of the joint and opposite in direction.

$$\tau_{kv} = -K_{kv}\dot{\theta}_k \quad (3.27)$$

$$\tau_{hv} = -K_{hv}\dot{\theta}_h \quad (3.28)$$

Typical low values of friction were $K_{kv} = K_{hv} = 0.08\text{Nm}(\text{rad})\text{s}^{-1}$. This coefficient of friction was used for most of the simulations until the actual value could be estimated from physical data of the HypoSEA-v1 (see Chapter 4).

In summary, the joint torques are merely the applied controller torques τ_k, τ_h summed with friction:

$$\tau_j = [(\tau_k + \tau_{kv}) \ (\tau_h + \tau_{hv})]^T \quad (3.29)$$

3.4.5 Contact with Ground

Besides torques applied by actuators at the joints, the other significant force exerted on the robot's links comes through contact with the ground. We can approach ground contact in two ways: by modeling it as a spring-damper system, or as a perfectly rigid geometric constraint. Let us consider each briefly.

Soft Contact Model

Although it was not extensively studied, a simulated experiment of the robot running on soft ground was performed. The ground interaction force for the case of a compliant ground was modeled as

$$F_{fx} = \begin{cases} K_{GP}x^n + K_{GD}\dot{x} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (3.30)$$

$$F_{fy} = \begin{cases} K_{GP}y^n + K_{GD}\dot{y} & y \geq 0 \\ 0 & y < 0 \end{cases} \quad (3.31)$$

These torques were added to the nonconservative torques τ :

$$\tau = \mathbf{D}_j^T \tau_j + \tau_f \quad (3.32)$$

$$\tau_f = [0 \ 0 \ 0 \ F_{fx} \ F_{fy}]^T \quad (3.33)$$

where $K_{GP} = 10000\text{N/m}$ is the ground stiffness coefficient, $K_{GD} = 1000\text{Ns/m}$ is the ground damping coefficient, $n = 1$ implies a linear spring, and x, y are the relative coordinates from the first contact point between the foot and ground to the current position of the foot.

The conclusion drawn from using a soft contact model of the ground is obvious in retrospect. A softer ground requires more energy to hop on because work is done on the ground as it is compressed. Exactly how much energy is lost depends on many factors and a realistic number is hard to estimate.

In fact, it was for this very reason that the soft contact ground model was quickly abandoned; it requires that two more unknown parameters (the stiffness and damping coefficients of the floor) be estimated and used in simulation. Another reason to abandon a soft contact model is that the simulation timestep must be greatly reduced to avoid numerical instability problems caused by stiff springs in naively integrated simulations. A variable timestep numerical integrator alleviates but does not eliminate this problem.

Rigid Ground Contact Model

If we switch from a soft contact ground model to a perfectly rigid constraint, our system will have fewer parameters and we may simulate faster by integrating numerically with larger timesteps. The addition of a perfectly rigid ground constraint is easily accomplished by writing the Jacobian of the point foot $\mathbf{J}_f(\mathbf{q})$, and enforcing a non-slipping condition $\mathbf{J}_f \dot{\mathbf{q}} + \dot{\mathbf{J}}_f \mathbf{q} = 0$. To numerically integrate during stance we must solve this constraint simultaneously with (3.23) using a Lagrange multiplier technique. The new equations of motion become

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_f^T \\ \mathbf{J}_f & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{D}_j^T \tau_j - \mathbf{C} \dot{\mathbf{q}} - \mathbf{g} \\ -\dot{\mathbf{J}}_f \dot{\mathbf{q}} \end{bmatrix} \quad (3.34)$$

where λ is a vector of Lagrange multipliers. Thankfully, the generalized coordinates we chose result in a constraint jacobian of the foot that is extremely simple, so during ground contact phase the constraint jacobian is merely

$$\mathbf{J}_f = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.35)$$

The physical meaning of λ suddenly becomes clear: $\lambda = [F_{fx} F_{fy}]^T$ are the forces of the foot on the ground. It is particularly useful to be able to compute these constraint forces when implementing ground reaction force control developed in Chapter 5.

3.4.6 Transitions between Flight and Stance Phases

Assuming a perfectly flat floor, the transition between flight phase and collision phase occurs when $y_f \leq 0, \dot{y}_f \leq 0$. We model this instantaneous collision as an impulse that completely stops the motion of the robot's foot. We derive this impulse from assumptions similar to those developed in the previous section.

We wish to find the impulse Λ that moves the robot's generalized velocities instantaneously to the point that satisfies the foot's new geometric constraint. If we integrate over a very small amount of time around the instant of impact, we can assume that \mathbf{q} and τ do not change, and that only the generalized velocities $\dot{\mathbf{q}}$ change instantaneously due to impulse Λ . We will use superscripts $-$ and $+$ to indicate the moments immediately before and after the impact, respectively.

$$\mathbf{M}\Delta\dot{\mathbf{q}} = -\mathbf{J}_f^T \Lambda \quad (3.36)$$

$$\mathbf{M}(\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-) = -\mathbf{J}_f^T \Lambda \quad (3.37)$$

$$\mathbf{M}\dot{\mathbf{q}}^+ = \mathbf{M}\dot{\mathbf{q}}^- - \mathbf{J}_f^T \Lambda \quad (3.38)$$

$$\dot{\mathbf{q}}^+ = \dot{\mathbf{q}}^- - \mathbf{M}^{-1} \mathbf{J}_f^T \Lambda \quad (3.39)$$

The post-collision foot must have zero velocity, meaning that $\mathbf{J}_f \dot{\mathbf{q}}^+ = 0$, and so we can now solve for the impact impulse Λ .

$$\mathbf{J}_f [\dot{\mathbf{q}}^- - \mathbf{M}^{-1} \mathbf{J}_f^T \Lambda] = 0 \quad (3.40)$$

$$\mathbf{J}_f \mathbf{M}^{-1} \mathbf{J}_f^T \Lambda = \mathbf{J}_f \dot{\mathbf{q}}^- \quad (3.41)$$

$$\Lambda = (\mathbf{J}_f \mathbf{M}^{-1} \mathbf{J}_f^T)^{-1} \mathbf{J}_f \dot{\mathbf{q}}^- \quad (3.42)$$

Therefore, the post-collision velocities $\dot{\mathbf{q}}^+$ can now be expressed using only the inertial matrix \mathbf{M} , foot constraint jacobian \mathbf{J}_f , and pre-collision velocity $\dot{\mathbf{q}}^-$:

$$\dot{\mathbf{q}}^+ = \dot{\mathbf{q}}^- - \mathbf{M}^{-1} \mathbf{J}_f^T (\mathbf{J}_f \mathbf{M}^{-1} \mathbf{J}_f^T)^{-1} \mathbf{J}_f \dot{\mathbf{q}}^- \quad (3.43)$$

Because the timing of the impact impulse and no-slip constraint release have a large effect on the dynamics of the system, it is important to use equation (3.43) only at the exact moment of collision, not after the foot has penetrated the ground or while the foot is still in the air. To accomplish this, it is recommended to use a root-finding technique – such as the secant method or Raphson-Newton – to find to the computer's numerical precision the exact instant in time when the foot contacts the ground, because this impulse can have a rather large effect on the dynamics. Such root-finding techniques can be found in any good textbook on numerical computation (Press, Flannery, Teukolsky, Vetterling, et al., 1986).

Finally, the transition from stance to flight phase occurs when the vertical component of the GRF on the foot is no longer pushing against the floor (*i.e.* when $F_y >= 0$). Nothing special needs to be done except the removal of the rigid contact constraint established in equation 3.34.

3.4.7 Ground Force Friction Constraint Verification

We have assumed up to this point that the direction of the impact impulse lies within the friction cone – however, it is not necessarily the case that the impact impulse or interaction forces between the foot and the ground actually are plausibly within the limits of friction.

The simulation was therefore programmed to signal an error if the tangential force F_{fx} and the normal force F_{fy} fail to satisfy the friction constraint $F_{fy} \leq K_f F_{fx}$ during stance. An error was likewise signaled if the impact with the foot and ground during the transition between flight and stance phases violated the friction cone.

3.4.8 Simulation Parameters

The physical parameters used in the simulation of the robot may be found in Table 3.1. CAD models were used to predict masses and moments of inertia for each component. Note that the moment of inertia of the body is

Table 3.1: Simulated Mechanical Model Parameters

	Description	Value	Unit
m_s	Shank mass	0.8	kg
m_t	Thigh mass	2.5	kg
m_b	Body mass	7.4	kg
I_s	Shank moment of inertia	0.078	kg·m
I_t	Thigh moment of inertia	0.06	kg·m
I_b	Body moment of inertia	0.23	kg·m
a_s	Shank CoM distance from foot	0.21	m
b_s	Shank CoM distance from knee	0.14	m
a_t	Thigh CoM distance from knee	0.18	m
b_t	Thigh CoM distance from hip	0.17	m
a_b	Body CoM distance from hip	0.19	m
K_{kv}	Knee joint viscous damping	0.08	N·s/m
K_{hv}	Hip joint viscous damping	0.08	N·s/m
Δt	Forward dynamics simulation time-step	0.002	s
Δt_{trn}	Forward dynamics impact accuracy	$< 10^{-11}$	s
K_f	Friction coefficient between foot and floor	0.8	-

only about twice that of the shank and thigh. These parameters are *not* the parameters of the robot described in Chapter 8, but rather the parameters of the simulated model during the design process.

3.5 Description of the Simulator

The simulator used to study the control and dynamics of hopping was written in Common Lisp and developed solely by the author. It runs on Linux Ubuntu version 8.4 or later, plots graphs with gnuplot, uses the Matlisp interface to LAPACK to perform linear algebra, and accesses OpenGL through the CL-OPENGL interface for visualization. Although not particularly optimized, performance was good enough for several hundred dynamics timestep computations per second on a relatively modest computer.³

A screenshot of the simulator is shown in Figure 3.7. Although CAD models of the preliminary hopping robot design are visible, the 3D environment was only for visualization purposes; the mathematics are roughly the same as presented in this chapter. See Chapter 7 for a similar-looking but internally different implementation of a simulator that actually uses 3D rigid body mechanics.

In anticipation of a later generalization to 3D, the OpenGL viewport camera may also be moved around freely in the 3D environment and set to track the motion of the robot in several ways for ease of viewing. Also, the simulator is capable of loading PNG texture files and STL model files so that CAD models of the robot may be directly visualized in the simulator. A fixed step size Runge-Kutta 4th order numerical integrator is used for computation of the equations of motion, and various zero-finding algorithms (Bisection, Newton-Raphson, Secant methods) were used to find collision instants and compute Lagrange multipliers.

A handheld dual-stick joystick similar to those used for the Sony Playstation™ can be used to control the camera or the robot during interactive real-time experiments. Multiple simulation objects may be run simultaneously and independently. Real-time monitoring and on-the-fly editing of simulation object parameters is possible

³The author used a single-core 2.4Ghz Pentium 4 processor with 1 gigabyte of RAM.

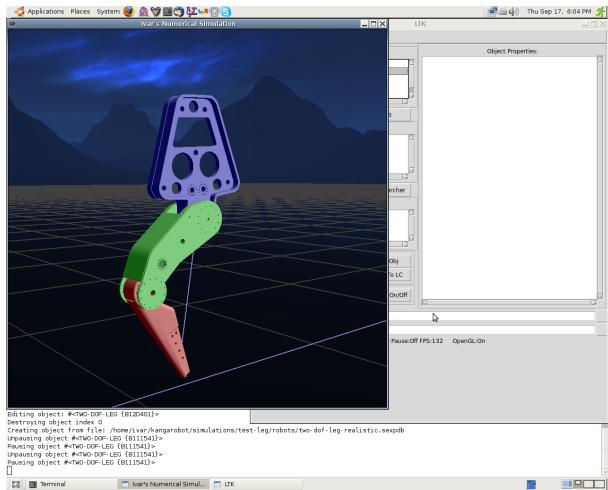


Figure 3.7: Screenshot of the simulator developed by the author to simulate the dynamics of a jumping robot, visualize its motion, and develop harmony between the actuators, controller, and mechanical dynamics.

through a GUI. Supported features include automatic data capture, screen capture, parameter vs parameter plots, OpenGL models and textures, and saving/loading of multiple controller and model parameters. Multiple simulation objects may be run simultaneously and controlled independently.

Despite all these features, this simulation was completely rewritten in another language and is now considered obsolete – the author now prefers to use the software presented in Chapter 7.

3.6 Simulation Results

Sample simulated dynamics of the robot hopping in place are shown in Figure 3.8. For convenience, about 50 different quantities are plotted simultaneously on the same figure so that the behavior of the robot can be easily understood. The simulation uses the controller developed in Chapter 5 as well as accurate models of the HypoSEA actuator presented in Chapter 4.

Starting from the leftmost column and describing each sub-figure from top to bottom, we will now explain the relevance of each graph. The focus of these graphs is on understanding the actuation requirements of the robot, so most quantities are described in terms of knee and hip motors, not on the absolute positions or velocities of the links in space.

Column 1: Joint and CoM Motion

Joint Angles The joint positions (θ_k, θ_h) show the angles of the knee and hip during stance, as well as the oscillation during flight phase. The foot position (x_f, y_f) is useful for measuring foot clearance and step length. The angle of the body θ_b is also shown.

Joint Velocities Joint velocities ($\dot{\theta}_k, \dot{\theta}_h$, a.k.a. ω_k, ω_h) show visible discontinuities when the robot collides inelastically with the ground. Foot velocities (\dot{x}_f, \dot{y}_f) allow us to measure the velocity of the foot at impact.

Joint Torques Joint torques (τ_k, τ_h) and ground reaction forces (F_{fx}, F_{fy}) help us understand the interaction forces between the robot and ground, and also the actuator torque requirements.

CoM Momenta By watching the horizontal, vertical, and angular momenta (p_{cx}, p_{cy}, L_c) of the system's mass centroid, we can characterize the dynamic behavior of the robot. The acceleration of gravity is easy to see

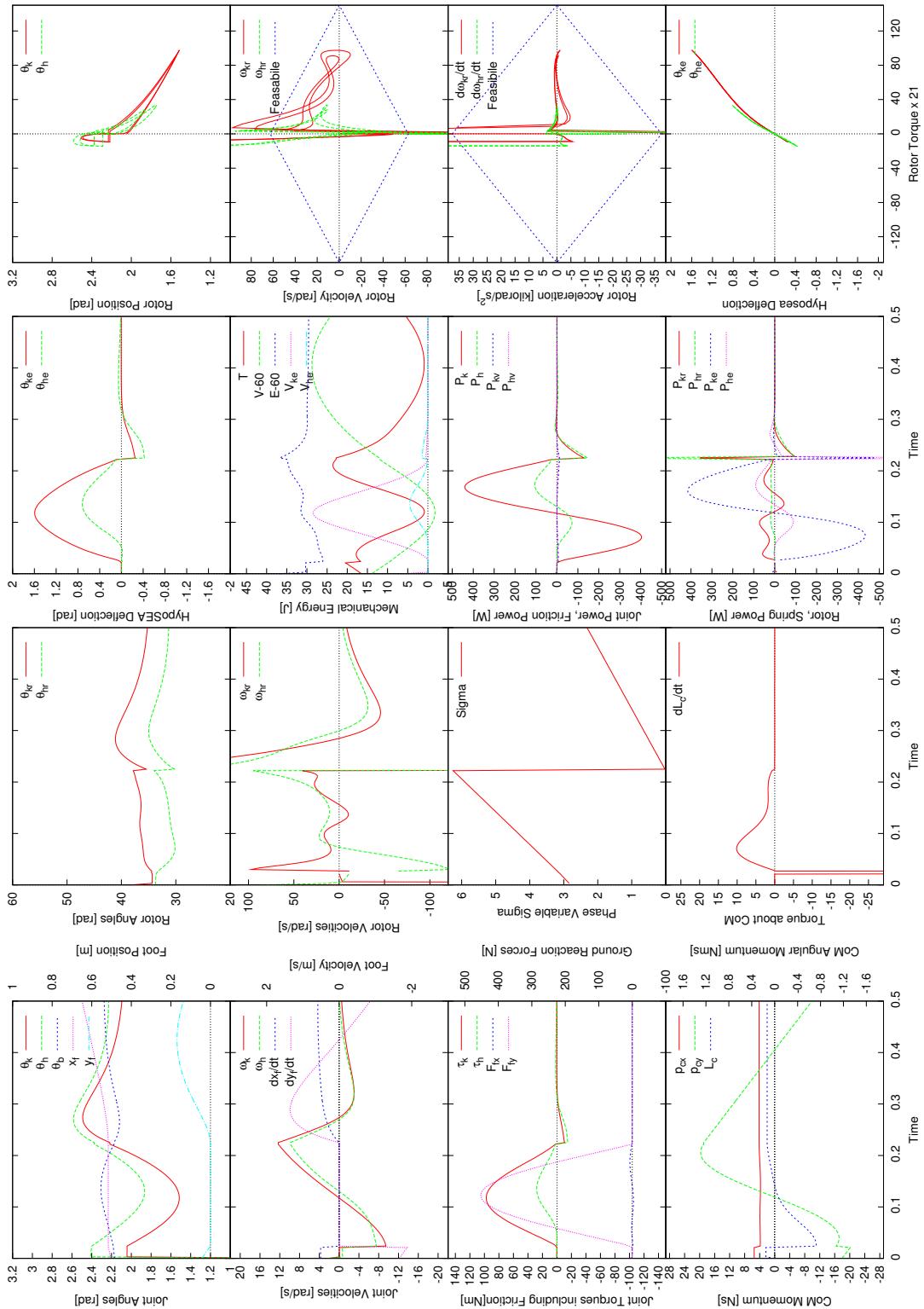


Figure 3.8: Simulation data of the mechanical and control dynamics of a hopping robot bouncing at steady state.

looking at p_{cy} , as are the impact impulses. We desire that L_c remain near zero for postural reasons. p_{cy} is an important quantity used to calculate control torques.

Column 2: Rotor Motion

Rotor Angles The position of the rotors (θ_{kr}, θ_{hr}) shows the trajectory they must follow to achieve proper joint torques through the HypoSEA mechanism. It is not a driving variable in the simulation, but is computed from the assumption that the rotors will be moved appropriately to create desired forces, which are primary. A small amount of rotor motion is desirable, as it reduces the mechanical work that needs to be done by the rotors.

Rotor Velocities The rotor velocities ($\dot{\theta}_{kr}, \dot{\theta}_{hr}$, a.k.a. ω_{kr}, ω_{hr}) show clearly that the rotors must often move much faster than the joints do during flight, to keep the passive oscillatory behavior of the springs under control.

Sigma This phase variable σ indicates the balance controller's estimate of the current progression through a hop cycle, measured from 0 to 2π . The instant of liftoff is defined as $\sigma = 0$, and touchdown is $\sigma = \pi$. This value is estimated from the orientation and momenta of the robot and is ideally linear at steady state.

CoM Torque The numerical derivative of the CoM angular momentum, it shows the rate of change of the angular momentum (\dot{L}_c) of the system. For postural reasons, this quantity should integrate to zero ($\int \dot{L}_c = 0$) during a cycle to keep the robot stable. It is an important control quantity.

Column 3: Energy and Power

HypoSEA Deflection The angular deflection (θ_{ke}, θ_{he}) of the HypoSEA spring-stretching mechanism is directly related to the joint torque, and can be used to compute the potential energy stored in the spring.

Mechanical Energy The total mechanical kinetic energy of the links T , potential energy of the links V , potential energies in the springs V_{ke}, V_{he} , and total robot system energy $E = T + V_g + V_{ke} + V_{he}$ are shown to help understand the flow of energy around the system. V and E are shifted downward by 60J so the y-axis scale may be smaller.

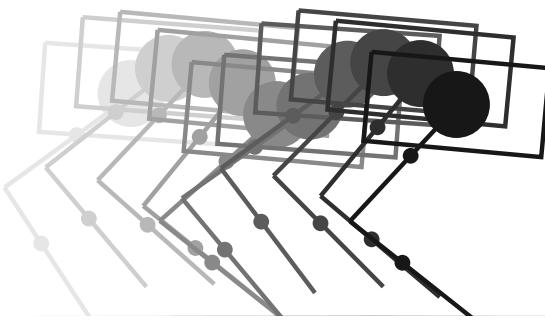
Joint Power The instantaneous mechanical power of the hip and knee joints (P_k, P_h) gives a rough estimate of the electrical power required. Also, we can see the energy loss to viscous friction is small by comparison, with only 10-15W instantaneous peaks. (P_{kv}, P_{hv})

Rotor/Spring Power The rotor powers (P_{kr}, P_{hr}) and spring powers (P_{ke}, P_{he}) show the distribution of instantaneous joint power. Ideally, we would like the rotor to do only positive work and for the spring to store and release much of the energy. Such behavior indicates a harmonious balance between mechanical dynamics and controller dynamics.

Column 4: Torque Relationships

Position-Torque By comparing joint position (θ_k, θ_h) vs controller torques (τ_{kn}, τ_{hn}), we can perhaps isolate energetically conservative trends in the controller behavior, and redesign our mechanical system to be more harmonious with this trend, or vice versa. The area contained in these loops indicates the amount of mechanical work done by the rotors.

Figure 3.9: Samples taken from an early hopping simulation at 0.1s intervals and drawn with postscript. The robot is moving from left to right. In this figure, the distance from the torso's CoM to the hip joint is very small (3cm) so that a Raibert-style controller could be applied.



Velocity-Torque The capability of a electric motor powered by a fixed-voltage motor controller is well approximated by a linear relationship between stall torque (no angular velocity) and maximum velocity (no torque). This creates a diamond-shaped operational area. If the torque (τ_k, τ_h) and desired rotor velocities ($\dot{\theta}_{kr}, \dot{\theta}_{hr}$) for each actuator stays in this area, we can assume that our motors have a good chance of reproducing the desired motion perfectly. Except for very quick accelerations after takeoff, motor torques seem largely feasible with the controller.

Acceleration-Torque The previous graph was not enough to ensure the actuators can reproduce a motion trajectory perfectly. Rotors have mass, and can not accelerate arbitrarily quickly. Again, if the desired torques (τ_{kn}, τ_{hn}) and rotor accelerations ($\ddot{\theta}_{kr}, \ddot{\theta}_{hr}$) stay within the diamond-shaped limits, we can be certain that our actuators will be able to reproduce the trajectory with good fidelity.

HypoSEA Deflection This data here is essentially the same as in Figure 3.5, but with the axes swapped. It lets us see the range of operation and similarity to the position-torque graph at the top of this column. Most of the operation will be in the first quadrant, associated with positive torques (and probably the stance phase).

Besides this data, it may be useful to look at Figure 3.9 to get an intuitive feel of what the hopping motion looked like. Unfortunately, this figure does not correspond precisely with the data just presented, but is for illustration purposes. Accompanying this thesis is a short video containing the actual hopping motion of the simulated robot – please refer to Appendix ?? for more information.

3.7 Discussion of Hopping Dynamics

In what direction should the robot hop?

It was initially assumed that having the point of the knee of the robot oriented in the direction of travel would be optimal, as that is how the knee bends in humans. However, simulation results showed the opposite: knees that bend “backward”, like chicken legs, result in lower knee and hip torques. This was somewhat surprising, but looking more closely at human and animal anatomy, we see that the ankle does indeed point “backwards”. Like JIMI’s knee, the first joint of most animals points in the direction opposite of normal travel.

What are joint flexions during normal operation?

The knee bends about $\Delta\theta_k = 0.4$ radians during stance (using a collision angle of $\theta_{kTD} = 1.7\text{rad}$), and the hip about $\Delta\theta_h = 0.5$ radians throughout the running phase. The knee flexion is slightly smaller and the GRF slightly higher than predicted by the simple analytic results because the robot is jumping slightly higher.

What are the torques on knee and hip during jumping?

The maximum torque on the knee predicted by simulation was $\tau_k = 120\text{Nm}$. This value is remarkably close to what was predicted analytically in section 3.3.4. By reducing the rate of leg swing so that it is not 'kicked' but 'swung' into position more gradually, the maximum hip torque could be reduced further.

How do link lengths affect joint torques?

Adjusting the length of the thigh and shank so that they are not equal changes the required torques at the knee and hip. The exact effect appears to depend on weight distribution, but one conclusion is easily made: all other things being equal, shortening the shank or lengthening the thigh increases the hip torque and reduces the knee torque.

What are the forces on the foot during stance?

The peak ground reaction force estimated from simulation during normal operation is approximately $F_{peak} = 350\text{N}$, slightly larger than was predicted by the analytical equations. The robot in simulation was also jumping slightly higher and landing with a fairly bent leg ($\theta_{kTD} = 2.0$).

How fast can the robot run?

Parameter and control settings that would let the robot run at about 1.1m/s were found. Given sufficient time and more tuning, this number could probably be increased. For the purposes of the simulation in this chapter, however, this number was sufficient.

What is the impact velocity with the ground?

In simulation the impact velocity of the foot with the ground was approximately 1.5m/s, meaning it is jumping higher in simulation than was analytically estimated in section 3.3.4. There is some collisional and frictional energy loss in the simulation so it needs to jump slightly harder than the analytical solution to overcome these losses.

How much energy is lost in an inelastic collision with the ground for various knee impact angles?

Deciding the posture of the robot during ground impact critically affects how much energy is lost. In general, we can say that the straighter the leg is, the more energy is lost, due to the increased transfer of momentum of the body to the ground. Bent legs allow for springier motions but require higher torques because of reduced leverage.

For relatively straight leg postures ($\theta_{kTD} = 2.0$), there appeared to be at least 4J of energy loss at impact. For more bent legs this was slightly smaller, but not significantly so. Very straight legs resulted in impact losses exceeding 10J, which could possibly damage the joints of the robot.

Does the robot always remain in the friction cone during normal operation?

By calculating the angle of the impact impulse and subsequent ground reaction forces, we may verify that the interaction forces between the foot and ground remain inside the friction cone.

All motions of the robot remained inside the cone created by a friction coefficient of $K = 0.8$, including horizontal accelerations. However, during steady state, the coefficient could be as low as $K = 0.2$ without

the robot falling over. This appears to be due to the decoupling of vertical GRF and horizontal GRF such that horizontal GRF is only used to regulate angular momentum. As a result, during steady state running, GRF vectors are very nearly vertical.

Impact posture appears to have the greatest effect on the necessary minimum coefficient of friction at steady state. At non-steady state, the faster the robot's accelerates horizontally, the higher the required friction coefficient.

3.8 Conclusion

The simulation presented in this chapter was extremely useful in studying the dynamics of the hopping robot before construction. It allowed forces, torques, velocities, and a variety of impact postures to be tried.

In particular, it became obvious how jumping robots are constrained by very strict power-to-weight requirements. Looking at the simulation data it is clear that the total joint power can momentarily exceed 300W of mechanical work for short periods of time. For a 10kg robot with two actuators capable of such power levels, this leads to a power-to-weight ratio of 60W/kg. Real world demands on the robot may be even higher than this.

For comparison, a popular lightplane such as the Cessna 172R weighs 1,111kg gross and has an 160HP engine (120 kW) – a ratio of 108W/kg. Although this is certainly not an apples-to-apples comparison, it may be fair to say that lightweight structures and powerful actuation are nearly as important for hopping robots as for aircraft.

In the case of JIMI, producing sufficient torque with the electric motors would prove to be a difficult problem. Driving the electric motors beyond their specification and actively cooling them would allow more torque to be produced by the motors, but this technique is best left as a last resort.

Instead, several solutions were used to reduce the predicted maximum necessary actuator torque:

- The legs were shortened, particularly the shank, reducing torque at the knee substantially.
- The stance time duty cycle was increased, reducing peak vertical GRF.
- The control law was subtly adjusted to widen and flatten the peak of the sinusoidal vertical GRF.
- The robot posture controller was adjusted to keep the leg comparatively straighter. This increases impact loss but reduced peak torques.
- The posture of the robot was changed so that both joints contribute energy more equally to the hopping motion, similar to how a person leans their torso forward when doing a squat-jump so as to use their back muscles in addition to leg muscles.

This concludes a discussion of the mechanical dynamics and actuation requirements of the hopping robot JIMI. The next chapter will discuss the actuators used in the robot and Chapter 5 will discuss the controller used to balance the robot in both simulation and in reality. The subjects of both of these chapters were modeled in the simulation presented in this chapter.

CHAPTER 4

Nonlinear Series Elastic Actuators

This chapter discusses the design and construction of two patented, nonlinear series elastic actuators called HypoSEA-v1 and HypoSEA-v2, shown in Figure 4.1. The HypoSEA-v1, short for Hypocycloid-based Series Elastic Actuator Version 1, was designed as a proof-of-concept device. It showed enough promise that it was revised into the HypoSEA-v2, which was designed specifically to match the dynamics simulated in Chapter 3 and the balance controller presented in Chapter 5. Two HypoSEA-v2 actuators were used in the construction of the robot JIMI, as will be presented in Chapter 8.

Both HypoSEA designs possess a great deal of mechanical energy storage in a series spring, can rotate continuously, have excellent mechanical efficiency, good backdrivability, good force-control fidelity, and low reflected impedance. However, neither actuator is particularly good by classical trajectory tracking control standards; force control bandwidth and power density are fairly low. Despite these disadvantages, the actuators are well-suited for highly dynamic motions such as hopping or running. Interestingly, the HypoSEA designs are sufficiently backdrivable that mechanical energy may be recovered into electrical energy with high efficiency.

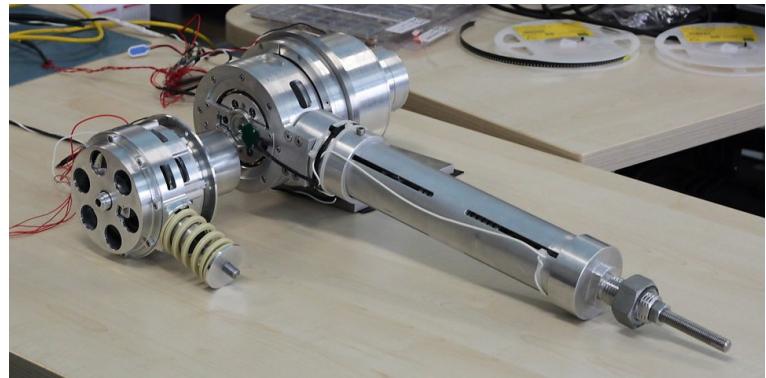
Before examining the designs, let us review what actuators have been developed by other researchers, and then proceed to examine the HypoSEA designs compared to them.

4.1 Existing Series Elastic Actuators

The first actuator specifically named “series elastic” was presented in “Series Elastic Actuators” by G. Pratt and M. Williamson, IROS 1995. However, the design is somewhat dated. The current state of the art in series-elastic actuation is briefly summarized in “Review of Actuators with Adjustable Compliance/Controllable Stiffness for Robotic Applications” by R. Van Ham, T. Sugar, B. Vanderborght, K. Hollander, and D. Lefeber.

Compared to existing designs the HypoSEA is superior in terms of maximum elastic energy storage capability and low mechanical impedance. The HypoSEA is uniquely able to stretch a spring a very large distance, and this large deflection not only stores significant energy but the large deflection of the spring allows very precise measurement of the force being transmitted to the output load, all other things being equal. The softness of this spring also gives the design exceptionally low mechanical impedance.

Figure 4.1: The HypoSEA-v1 (right) and HypoSEA-v2 (left), two low-impedance compliant actuators designed specifically for hopping tasks. The prototype HypoSEA-v1 creates peak torques in excess of 60Nm, stores \sim 40J of energy in a series spring, and weighs 8.2kg. The improved HypoSEA-v2 stores $>$ 25J of energy, has a peak torque of $>$ 55Nm, and weighs just 2.88kg.



In terms of power-to-weight ratio, size, maximum torque, or zero-force bandwidth, other designs currently surpass the HypoSEA as other researchers have chosen different performance quantities to maximize.

Although the mechanisms are substantially different, the stiffness characteristics of the HypoSEA most resemble those of the VS-joint being developed at DLR (?). The VS-joint module can withstand 160Nm torque, deflections of \pm 14 degrees, is 97mm in diameter, 106mm long, weighs 1.4kg, and stores up to 16J of energy. The total mass of the system with the 350W motor and harmonic drive is unfortunately not reported.

The LADD actuator is another nonlinear, high-efficiency transmissions based on the concept of twisted strings (Mennitto & Buehler, 1997). The function is similar to a ballscrew, in which the rotational motion of a motor is converted into a linear contraction, meaning that the HypoSEAs and LADD actuators fairly comparable (Mennitto, 1995). The LADD actuators weigh 1.3kg and produce 60Nm peak torque, but this figure does not include series elasticity or sensing, and significant stresses are placed upon the limb to support the cable tensions.

As mentioned in section 2.2.3, the MACCEPA and MACCEPA-II are nonlinear actuators with increasing incremental stiffnesses (Van Ham, Vanderborght, Van Damme, Verrelst, & Lefeber, 2007; Vanderborght et al., 2009). The torque-angle curves of the latter mechanism are stiffening up to about 45 degrees of deflection, meaning the geometry possesses an absolute maximum deflection limitation similar to the HypoSEA but for a lower level of deflection. The design is very lightweight, and a long deflection spring could also be used. In many ways, it is an excellent design for a lightweight jumping robot, although it places significant stresses on the surrounding structure required to use it. Also, the high stresses on the cables and small bending radius at one point suggest that the cables will break from fatigue after limited use, so modification of the mechanical geometry would be required.

4.2 Overview of the HypoSEA

In section 3.3.4, a rough specification of a compliant actuator was developed from a simple analysis of hopping dynamics. In review, the actuation performance goals used during the design of the HypoSEA were extensive energy storage (30J), very low mechanical impedance, and a harmony between the torques commanded by JIMI's centroidal momentum controller and the natural passive behavior of the series springs. We now provide an overview of the HypoSEA design and explain how it satisfies these goals.

4.2.1 Electric Motor and Transmission

Electric motors have fundamental limitations on the amount of torque that they can generate (Hollerbach et al., 1992). To increase the amount of torque available, a mechanical advantage such as a gear speed reduction unit

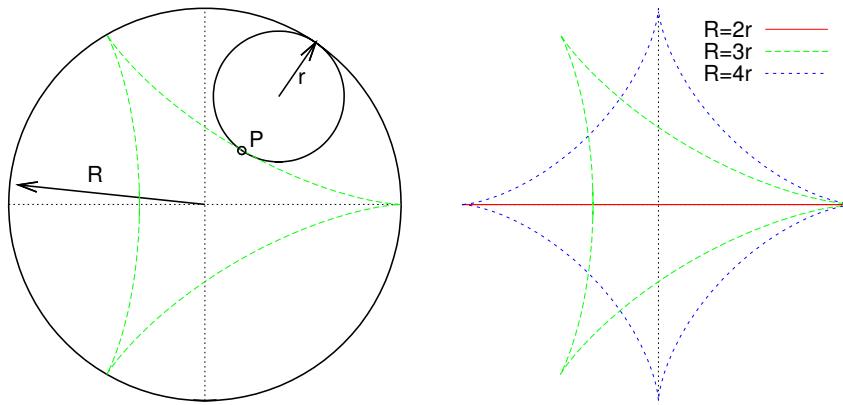


Figure 4.2: Left: A hypocycloid is a curve drawn by tracing a point on a smaller circle rolling within the interior of a larger circle. Right: Depending on the ratio of the inner circle to the outer circle, a different number of cusps will be drawn.

is generally used. However, it is easily shown that using reducing the speed of the output N times increases the mechanical impedance at the output by N^2 times (Hurst, 2008).

Therefore, to keep the mechanical impedance low, it was desirable for the HypoSEA to use a low gear ratio; to create a good torque actuator, it unsurprisingly helps to start with a good torque source. For this reason, EmoTec 2303 high torque brushless DC motor was selected. The motor is rated for a peak torque of 2.7Nm, 111W continuous, and has a mass of 640g – although when a rotor shaft, bearings, an enclosure, and position encoders are included it weighs much closer to a kilogram. Because the duty cycle of a hopping robot is less than 50%, using the motor at above its continuous duty specification is plausible. If the motor was driven beyond its specification, producing instantaneous torques of up to 5-6Nm is not be unrealistic; the motor is rated to demagnetize at 8-9Nm.

Even with an incredible 6Nm of torque at the rotor, this a factor of 20 times less than the design goal of 120Nm. Some type of speed reduction unit must be used. Although harmonic drives can create very high gear reductions in small packages, they have lower efficiencies and are less easily backdriven than planetary gears. For this reason a compound planetary gear system was used. This gear system will be described in section 4.2.5.

4.2.2 Nonlinear Elasticity

When we think of creating a powerful spring with a high energy storage to weight ratios, it is tempting to think immediately of high performance elastic materials. Fiberglass bow springs can store a great deal of energy in a light weight, and two successful examples of this approach are the ECD leg and Bow Leg hopper (Hurst, 2008; Zeglin, 1999). Elastomers are another material with great energy storage per unit weight (Mennitto, 1995) .

We will follow a different approach and start with the equation for the potential energy stored in a Hookean spring ($V_e = \frac{1}{2}K_e d^2$). Clearly, the fastest way to increase energy storage is to increase the displacement d of the spring, not its stiffness K_e . A softer spring displacing a large distance will store the same amount of energy as a stiff spring displacing a short distance, and the lower resulting forces will be easier to support. A lightweight structure around a heavier spring may save more weight in total than a lightweight spring with heavier supporting structure.

But how to create a nonlinearity that matches the torque-angle relationship needed by a hopping robot with revolute joints? Although custom nonlinear springs can be manufactured, it is also simple to create a mechanism which stretches a linear spring at a nonlinear rate. This approach is often followed in the literature (Wolf & Hirzinger, 2008). Let us now look at the nonlinear mechanism by which the HypoSEA creates a large amount of spring deflection.

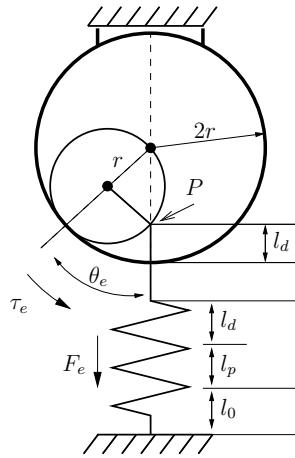


Figure 4.3: Model of the HypoSEA spring stretching mechanism, showing how the spring is stretched along the vertical line as the inner circle rotates without slipping on the inside of the fixed larger circle.

4.2.3 Hypocycloids

The HypoSEA actuators use a geometric shape called a *hypocycloid*; a *cycloid* is a plane curve traced out by a point attached to a circle as it rolls without slipping along another curve, and a *hypocycloid* is a curve traced out by a point P attached to a smaller circle of radius r as it rolls along the inside of a larger circle of radius R (Figure 4.2). If the ratio of the radii R and r is an integer, then cusps will appear in quantity equal to this integer.

If the hypocycloid only has two cusps (*i.e.* when $R = 2r$), the motion of point P is a perfectly straight line being drawn. This was discovered in the 19th century as one of the many ways to create a straight line motion from a rotational motion.

It is this particular curve that is used by the HypoSEA to stretch a spring. It enables a relatively large amount of spring travel, which meant that a large, soft spring can be used to store energy without excessive internal stresses. Also, it can produce a family of nonlinear torque-angle relations that resemble linear or quadratic curves, depending on the spring pretension. Let us look more closely at the mechanism now.

4.2.4 Hypocycloid Spring-Stretching Mechanism

The model of the nonlinear stretching mechanism under consideration is shown in Figure 4.3. The inner circle of radius r contacts the outer circle without slipping and point P traces a vertical straight line as θ_e is varied. A rod connected to point P stretches a spring by an amount l_d , and the force from the elastic spring creates a torque τ_e about the center of the larger circle. The spring is assumed to be prestretched from its natural, zero-force length l_0 by a pretension distance l_p so that there is no slack in rod or spring. In reality, to enforce the non-slipping condition of the rolling between the circles, gears are used instead of perfect circles, but such details do not change fundamental operation of the device.

Let us now study the mathematics of this mechanism. We can derive the torque-angle relation created by the spring and hypocycloid mechanism from three obvious geometric relations

$$l_d(\theta_e) = 2r(1 - \cos \theta_e) \quad (4.1)$$

$$\tau_e(\theta_e) = F_e 2r \sin \theta_e \quad (4.2)$$

$$F_e(\theta_e) = K_e(l_d + l_p) \quad (4.3)$$

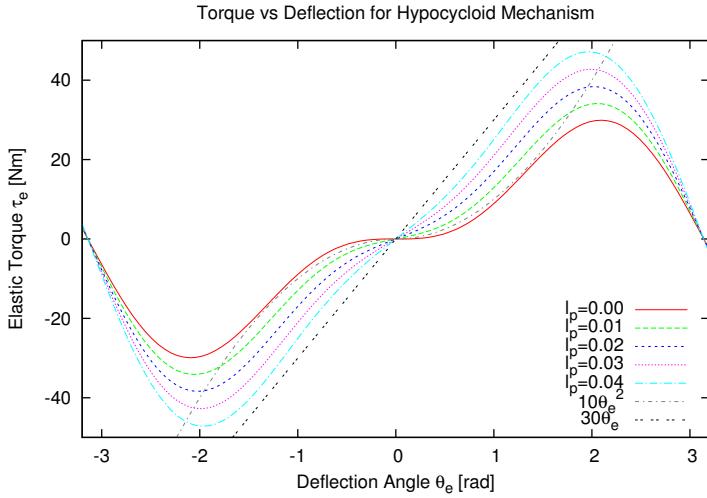


Figure 4.4: Torque-angle curve for the HypoSEA transmission for various pretension distances l_p . For low pretensions $l_p = 0.0$, the curve approximates the quadratic curve $10\theta_e^2$. For higher pretensions $l_p = 0.04$, the curve is much closer to the linear relation $30\theta_e$. This shows the versatility of the HypoSEA mechanism for the family of torque-angle relations between quadratic and linear.

From the geometric relations, we can write the torque explicitly as

$$\tau_e(\theta_e) = K_e(2r)(\sin \theta_e)(2r(1 - \cos \theta_e) + l_p) \quad (4.4)$$

where K_e is the stiffness of the spring, r is the radius of the inner circle, $l_d(\theta_e)$ is the distance from the intersection of the inner circle and the vertical axis with the bottom of the outer circle, and l_p is the pretension displacement of the spring at $\theta_e = 0$. Refer to Figure 4.3 as necessary and note that both the left and right sides of the figure are mathematically equivalent.

A graph of the torque-angle relationship for various pretension distances l_p is shown in Figure 4.4, using the values of $r = 0.024$, $K_e = 10\text{kN/m}$. Note that the torque-angle relation becomes more linear as pretension is increased.

The useful range of the mechanism appears to be about $\theta_e \approx \pm 2.0$ radians (± 120 degrees). Above this deflection, the incremental stiffness $\frac{d\tau_e}{d\theta_e}$ begins to decrease, which is not a desirable characteristic for reasons of stability.

The potential energy stored in the spring as a function of angle is

$$V_e(\theta_e) = \int_0^{\theta_e} \tau_e d\theta_e \quad (4.5)$$

$$= \int_0^{\theta_e} K_e(2r)(\sin \theta_e)(2r(1 - \cos \theta_e) + l_p) d\theta_e \quad (4.6)$$

$$= \int_0^{\theta_e} K_e(2r \cos \theta_e - 2r - l_p)(-2r \sin \theta_e) d\theta_e \quad (4.7)$$

$$= \frac{1}{2}K_e \left((2r \cos \theta_e - 2r - l_p)^2 - l_p^2 \right) \quad (4.8)$$

For intuitive reasons, we may also wish to simply consider the linear deflection of the spring. In such a case, the amount of energy stored in the hypocycloid mechanism's spring when it is displaced from its natural length

by some pretension distance l_p plus the deflection l_d from the hypocycloid mechanism is

$$V_e = \int_{l_p}^{l_p + l_d} K_e l_e dl_e = \frac{1}{2} K_e (l_d^2 + l_p l_d) \quad (4.9)$$

If $l_p = 0.02\text{m}$, $K_e = 10000\text{N/m}$, and the maximum displacement of the spring is $d = 0.08\text{m}$, then the spring can store approximately 40J of energy. Assuming that there is sufficient torque from the motor to resist the force from the maximum displacement of the spring, increasing the pretension will further increase the levels of energy storage.

The incremental stiffness – in other words, the instantaneous linear stiffness at a particular deflection – can be found from equation 4.4.

$$\frac{\partial \tau}{\partial \theta_e} = K_e 2r ((2r + l_p) \cos \theta_e - 4r \cos^2 \theta_e + 2r) \quad (4.10)$$

This equation can be useful for estimating small-force oscillation frequencies, using linearized models, or gain-scheduling the PID controller to maintain stability.

4.2.5 Mathematics of Planetary Gear Systems

The HypoSEA uses a combination of several planetary gear systems in both compound planetary and differential configurations. The locations of the sun, planet, annulus, and planet carrier are well known to most engineers, but the mathematics of these systems are more easily forgotten and will now be reviewed. We will use the subscripts \cdot_A , \cdot_P , \cdot_S , \cdot_C to indicate annulus, planet gear, sun gear, and planet carrier, respectively. When written as A , P , or S , it refers to the number of teeth on each gear.

For two spur gears P and S to properly mesh, they must have the same diametral pitch. When they mesh, the linear velocities on their edges are equal.

$$P\omega_P = -S\omega_S \quad (4.11)$$

If one of the gears is an annulus gear, the rotation of both gears will be in the same direction

$$P\omega_P = A\omega_A \quad (4.12)$$

For the case of a planetary gear P meshing with the sun gear S but rotating also on a planet carrier, we need to write this equation in a rotating frame attached to the carrier arm.

$$S\omega_S^C = -P\omega_P^C \quad (4.13)$$

This is equivalently written

$$S(\omega_S - \omega_C) = -P(\omega_P - \omega_C) \quad (4.14)$$

From equation (4.14), it is a few short algebraic steps to find the equation governing the motion of a planetary gear system:

$$\frac{\omega_C - \omega_A}{\omega_S - \omega_C} = \frac{S}{A} \quad (4.15)$$

The number of teeth of the planetary system must be $P = \frac{1}{2}(A - S)$ for the planetary system to mesh properly. From equation 4.15, we can work out three important special cases:

1. If the annulus is held fixed, then

$$\omega_C = \left(\frac{A}{S} + 1 \right)^{-1} \omega_S \quad (4.16)$$

2. If the carrier is held fixed, then

$$\omega_A = -\frac{S}{A} \omega_S \quad (4.17)$$

3. If the sun is held fixed, then

$$\omega_A = \left(\frac{S}{A} + 1 \right) \omega_C \quad (4.18)$$

We can see that for a simple planetary reduction system, using a fixed annulus provides the highest level of reduction in a single planetary stage. Not coincidentally, this is the arrangement used in the HypoSEA transmissions.

4.2.6 HypoSEA Design Advantages

The HypoSEA architecture has several advantages over conventional actuation systems:

- Reflected inertia of the actuator is minimized by using a low-reduction two-stage compound planetary gear system. Also, for small pretensions l_p , the incremental stiffness $\frac{d\tau_e}{d\theta_e}$ can be made extremely low and the impedance seen by the output mass will be exceptionally small.
- The output may rotate continuously an arbitrary amount without a problem (subject to cabling limitations).
- A single spring stores energy regardless of direction of deflection, saving weight and space.
- A large deflection of the spring is possible, efficiently storing a large amount of energy without excessively high forces.
- No expensive, high-friction harmonic drive is required.
- Stiffness characteristics are easily adjustable by pretensioning the spring. Although in this thesis this pretension is not automatically controllable, it would be trivial to add a second motor and acme screw to enable this capability.
- Alternatively, antagonistic or redundant configurations of these actuators would also allow online adjustable stiffness, because two nonlinear elastic elements can be combined with different antagonistic tensions to create a new stiffness profile.
- The design is highly integrated, combining the high-torque planetary gear reduction and a hypocycloid mechanism in a single, small package.

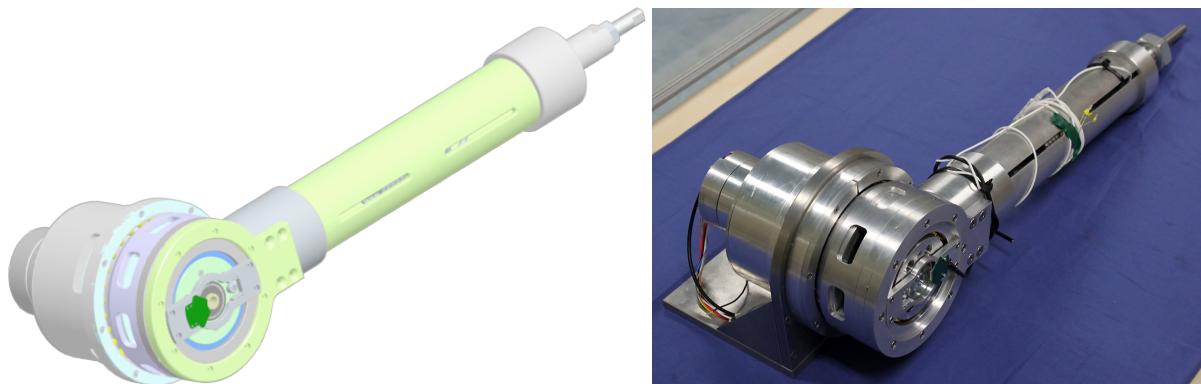


Figure 4.5: Perspective, exterior view of the HypoSEA-v1.

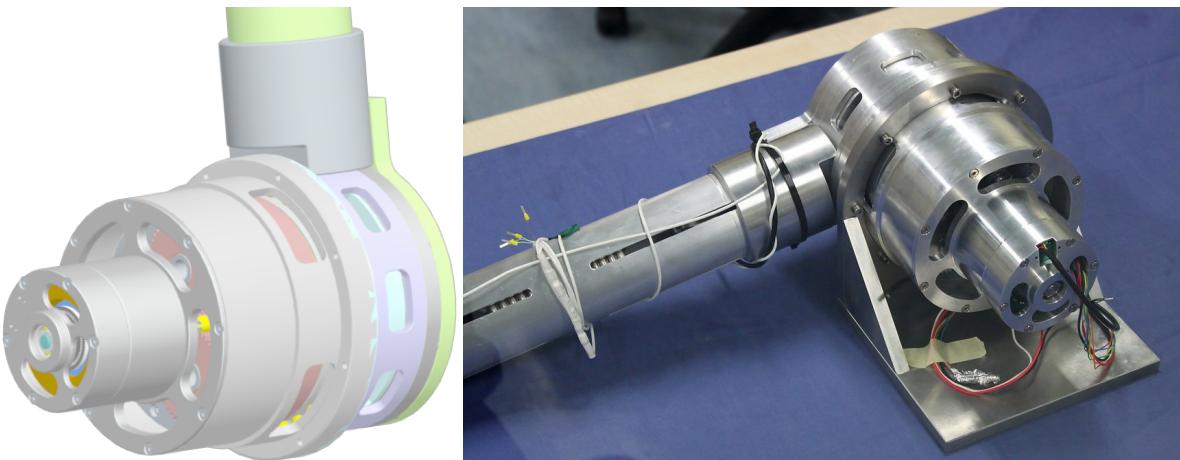


Figure 4.6: Exterior view of the HypoSEA-v1 invention. Motor assembly (M) is in the foreground.

4.3 The HypoSEA-v1

This section presents in detail the design of the HypoSEA-v1, describing its transmission and a detailed description of its components. Figures 4.5 and 4.6 show exterior views of the HypoSEA-v1 rendered by CAD software and also photographs taken after the prototype was assembled.

4.3.1 Assemblies

The HypoSEA may be roughly divided into five assemblies. The location of each of these assemblies is shown by color in figure 4.8. Referring to this figure as needed, let us name each of the assemblies as follows:

M A *motor assembly*

G A compound planetary *gear assembly*

D A *differential gear assembly*

H A *hypocycloid gear assembly*



Figure 4.7: **Left:** Prototype of the HypoSEA-v1 mounted on a table. **Right:** Components of the HypoSEA-v1 prototype before assembly.

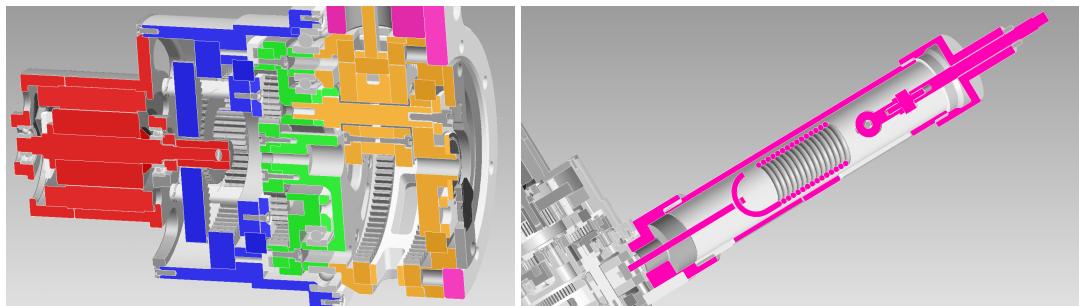


Figure 4.8: Motor assemblies labeled by color. **Red:** Motor Assembly R. **Green:** Compound planetary gear assembly G. **Blue:** Velocity-doubling gear assembly D. **Orange:** Hypocycloid gear assembly H. **Magenta:** Elastic element assembly E.

E A elastic element assembly

The letters M,G,D,H,E will be used as prefixes to indicate in which assembly a component may be categorized. For example, component M3 indicates a component in the motor assembly. Most major components may be seen by referring to figure 4.9 as needed for labeled cross-sections of the invention.

Because the HypoSEA design is highly integrated, to some extent the distinctions between assemblies is not perfectly strict. For example, component G5 functions as a planetary gear carrier for the both planetary gears in assembly G and also the planetary gears in assembly D, and could be categorized as belonging to either assembly.

4.3.2 Sensors

The HypoSEA-v1 contains three sensors:

- S1** An angular position sensor M7/M8 to measure the rotational orientation of the rotor M2.
- S2** An angular position sensor H18/H19 to measure the rotational orientation of the hypocycloid mechanism (H8 relative to H20)
- S3** A force/torque sensor E7 to measure the tension or compression force through the elastic element E5.

The latter of these three was not particularly useful – in practice, it was found to be much more accurate to compute the force based on the displacement of the spring. For this reason, the force/torque sensor was removed for the HypoSEA-v2.

One interesting thing about the sensor configuration is that the joint angle is not measured directly, and is instead computed from the measurements of S1 and S2. By using S2 to measure the angle of component H8 relative to E1 it is possible to more accurately compute joint torque than if it were measured relative to G1. This is because the gear assembly G has some unavoidable gear backlash that will yield some uncertainty as to the true position of D1 if it were computed from the position of M12. As the invention is a device intended to generate very accurate torques, it is preferable that any backlash in the device yield a small uncertainty in the absolute angular position of E1 as opposed to having uncertainty on the deflection of E1 relative to H8.

4.3.3 HypoSEA-v1 Transmission

The transmission consists of two parts: a compound planetary reduction system and a differential. Both deserve brief explanation.

Differential Mechanism

As mentioned in section 4.2.4, the hypocycloid mechanism has a maximum elastic deflection of $\theta_e = \pm 120$ degrees. In practice, this is probably too much deflection to be useful for most applications. We would prefer reduce this angle by a factor of three to make the maximum deflection ± 40 degrees. This gear reduction is called G_{je} , referring to the gear ratio between the joint and the elastic element. Further gear reduction between the elastic element and the motor rotor is also required; we call this G_{er} . Reference Figure 4.10 as required to see the connections of the elements in the system.

A block diagram of the HypoSEA-v1 transmission can be seen in Figure 4.10. The deflection of the hypocycloid mechanism is equal to the difference of the joint angle and the rotor angle, with some gear reduction in between to change the torque-velocity characteristics.

The mathematical relationship between the torques at each point is

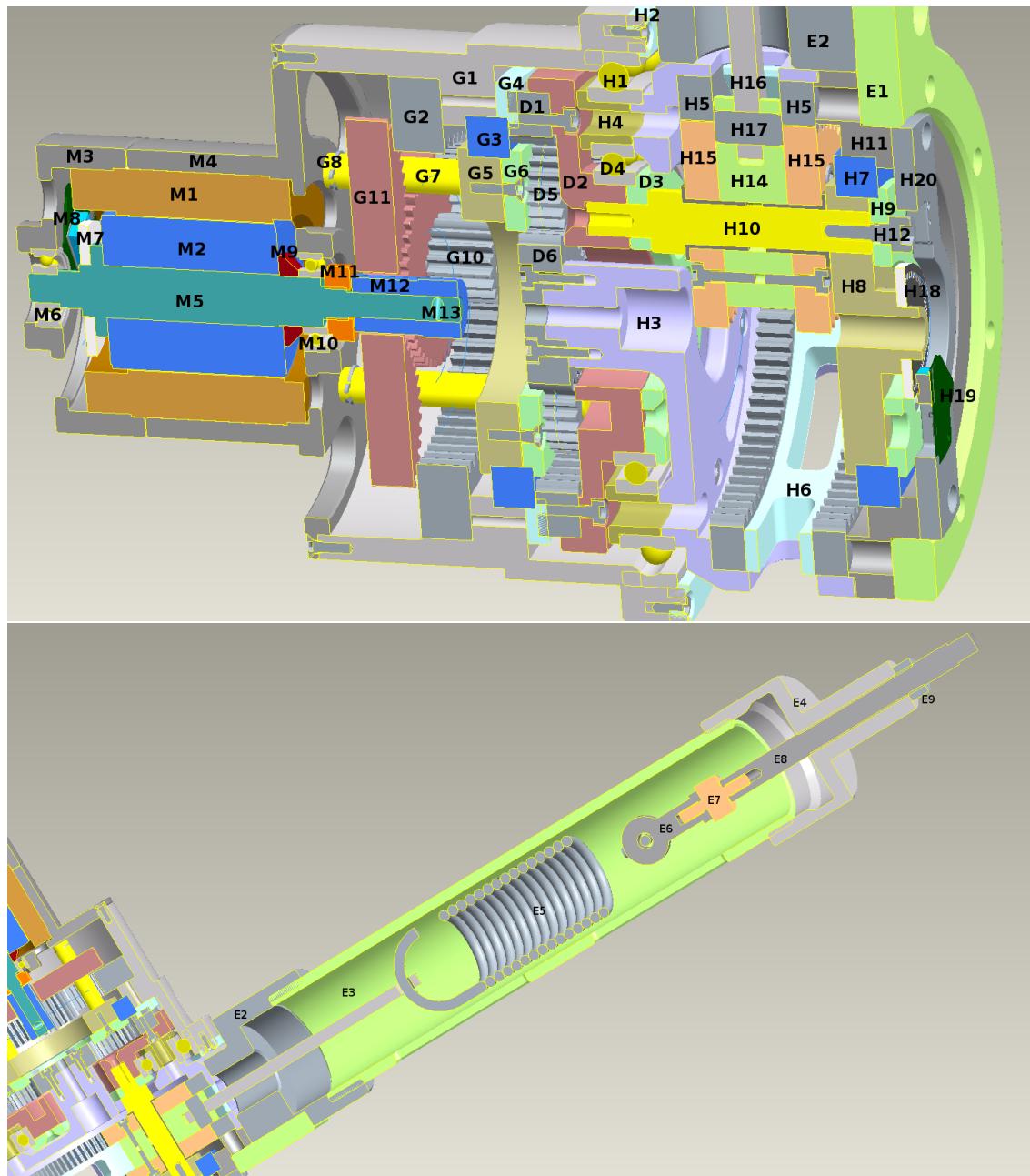


Figure 4.9: **Top:** Cross section of the invention labeling most of the components from assemblies M, G, D, and H. **Bottom:** Cross section of the HypoSEA with labeled components in the E assembly.

Figure 4.10: Block diagram of the HypoSEA-v1 actuator structure, expressing the relationship between torques and angles. The “output link” refers to E1, “hypocycloid mechanism” to H8, “joint angle” to the angular difference between G1 and E1 , and the “rotor” refers to component M2.

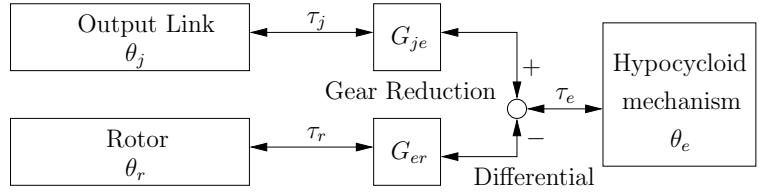
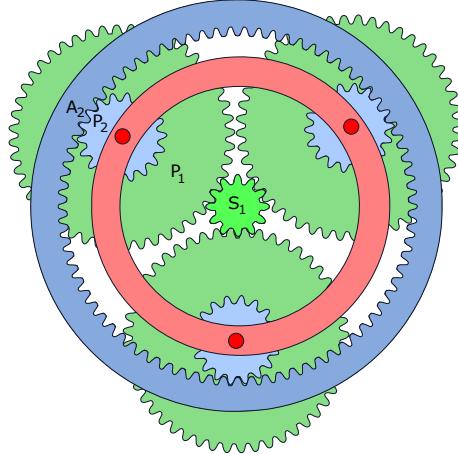


Figure 4.11: Compound planetary gear system of the HypoSEA. Identical numerical subscripts indicate that gears mesh. The two planetary gears P_1 , P_2 are rigidly connected and therefore rotate together. The red carrier ring supports the axles of all the planets.



$$G_{je} G_{er} \tau_r = G_{je} \tau_e = \tau_j \quad (4.19)$$

The relationship between the angular positions of each element is reciprocal to the torque relationship

$$\theta_r = G_{er} \theta_e = G_{je} G_{er} \theta_j \quad (4.20)$$

Compound Planetary

The HypoSEA uses a compound planetary gear system to provide a mechanical advantage from the rotor to the elasticity (G_{er}). Referring to Figures 4.9 and 4.11 as needed, let the sun gear M12 have S_1 teeth, the pair of planets rigidly connected to each other (G11 and G10) have P_1, P_2 teeth, and the annulus gear G2 have A_2 teeth. If the annulus G2 does not rotate, the resulting velocity of the carrier G5/G6 can be derived in terms of the velocity of the sun gear.

$$\omega_c = \left(\frac{P_1 A_2}{P_2 S_1} + 1 \right)^{-1} \omega_{s_1} = G_{er}^{-1} \omega_{s_1} \quad (4.21)$$

We note that this expression looks very similar to equation 4.16 but with the addition of the factor $\frac{P_1}{P_2}$. For the HypoSEA-v1 transmission, $P_1 = 48$, $P_2 = 18$, $S_1 = 12$, $A_2 = 78$. Therefore, according to equation (4.21), the total gear reduction of the first stage is $G_{er} = \frac{55}{3} = 18\bar{3}$. This is excellent for a single stage planetary system.

We now derive the equation of the mechanical differential connecting the carrier C (parts G5/G6), the hypocycloid angle H and the output link annulus A_3 . From the reference frame of the carrier C we may immediately use equation (4.17).

$$\omega_{A_3}^C = -\frac{S_3}{A_3} \omega_{S_3}^C \quad (4.22)$$

Moving this equation to the original reference frame A_2 and doing some algebra

$$\begin{aligned}\omega_{A_3} - \omega_C &= -\frac{S_3}{A_3} (\omega_{S_3} - \omega_C) \\ \omega_{A_3} - G_{er}^{-1} \omega_{S_1} &= -\frac{S_3}{A_3} \omega_{S_3} + \frac{S}{A} G_{er}^{-1} \omega_{S_1} \\ \omega_{A_3} &= -\frac{S_3}{A_3} \omega_{S_3} + (G_{er}^{-1} + \frac{S_3}{A_3} G_{er}^{-1}) \omega_{S_1} \\ \omega_{A_3} - \omega_{S_3} &= (-1 - \frac{S_3}{A_3}) \omega_{S_3} + G_{er}^{-1} \left(1 + \frac{S_3}{A_3}\right) \omega_{S_1} \\ \omega_{A_3} - \omega_{S_3} &= \left(1 + \frac{S_3}{A_3}\right) (G_{er}^{-1} \omega_{S_1} - \omega_{S_3})\end{aligned}$$

Let's rename these quantities in more familiar coordinates and call $G_{je} = \left(1 + \frac{S_3}{A_3}\right)$, $\dot{\theta}_j = \omega_{S_3}$, $\dot{\theta}_r = \omega_{S_1}$ and $\dot{\theta}_e = \omega_{A_3} - \omega_{S_3}$.

The expressions for the angular velocity of the elastic element and joint are:

$$\dot{\theta}_e = G_{je} (G_{er}^{-1} \dot{\theta}_r - \dot{\theta}_j) \quad (4.23)$$

$$\dot{\theta}_j = \dot{G}_{er}^{-1} \dot{\theta}_r - G_{je}^{-1} \dot{\theta}_e \quad (4.24)$$

In summary, if $S_3 = 36$ and $A_3 = 84$, then $G_{je} = \frac{10}{7}$.

4.4 Detailed Description of HypoSEA-v1 Components

The following section describes each of the components in the HypoSEA in more detail. Unless otherwise noted, all parts are made of high strength aluminum, except for shafts and gears, which are made of steel. Bolts, screws, washers, etc are not included in the numbering system for the sake of brevity.

4.4.1 Motor Assembly (M)

The motor assembly (Figure 4.12) is what converts an electric current into a torque on the small sun gear M12.

M1 Stator. An electric motor stator. In the prototype, a 12-pole brushless DC motor (Emotech #2303) was used, but in principle any revolute motor would work. Held tightly in place by axial pressure from M3 and M4.

M2 Rotor. An electric motor rotor. In the prototype, a 12-pole rotor for a brushless DC motor was used. Fits tightly on shaft M5 with no slipping.

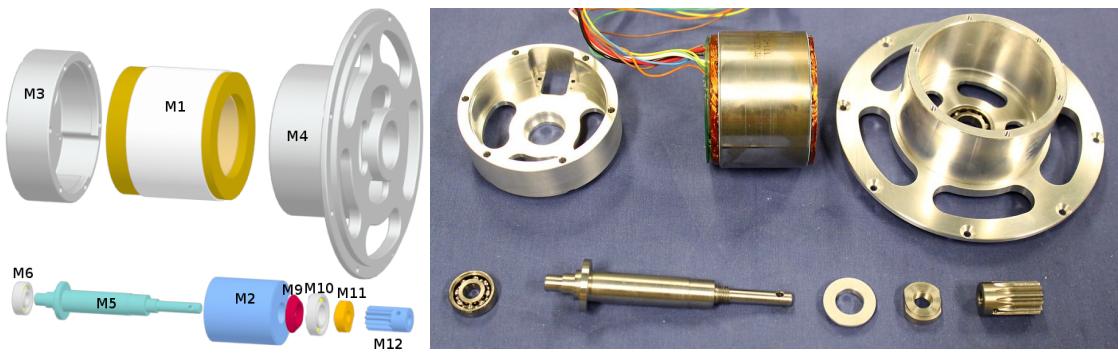


Figure 4.12: **Left:** Motor assembly (M) exploded. **Right:** Actual motor assembly components.

M3 Stator housing. Supports bearing M6 and is attached to M4 with several machine screws to provide the axial clamping force to hold stator M1 fixed. The optical encoder sensor electronics M8 also mount onto this piece with two screws. Bearings M6 are also supported by this piece. Stator and sensor wiring pass through the small holes in this piece. These holes also allow ventilation of the stator.

M4 Stator housing. Supports bearing M10 and together with M3 provides the axial clamping force to hold stator M1 fixed. Connects to G1 with several screws.

M5 Shaft. Conveys torque from rotor M2 to sun gear M12. This shaft is supported by bearings M6 and M10. Optical encoder disk M7 is glued to the shaft so that its relative angular position may be measured. The shaft is threaded so that M11 may be tightened to align all the pieces axially. Gear M12 is rigidly attached to this shaft via a set screw, keyway, or cotter pin. Rotor M2 is typically glued to this shaft to avoid slipping.

M6 Bearing. Supports one end of shaft M5. See also M10.

M7 Optical encoder disk. A reflective quadrature encoder disk affixed to shaft M5, rotating with it and providing a reflective target for M8's sensor.

M8 Optical encoder sensor board. An optical quadrature encoder circuit board that measures the rotation of the reflective disk M7.

M9 Rotor shaft spacer/clamp. Placed between bearing M10 and the rotor M2 to align the parts along the shaft and prevent rotor M2 from rubbing against M10.

M10 Bearing. Supports one end of shaft M5. See also M6.

M11 Locking collar. Used on the threaded part of M5 to tension the bearings axially against each other.

M12 Sun gear. Serves as the sun gear in the compound planetary gear system. The input to the reduction system, it typically has 12-24 teeth, depending on the desired gear reduction.

M13 Sun gear locking pin. A cotter pin, keyway, or set screw used to lock the rotation of sun gear M12 to shaft M5.

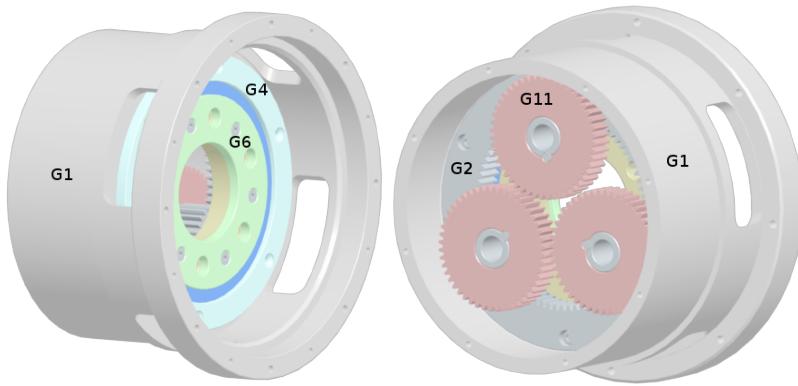


Figure 4.13: Two views of the partially-assembled gear reduction assembly G, with shafts hidden.

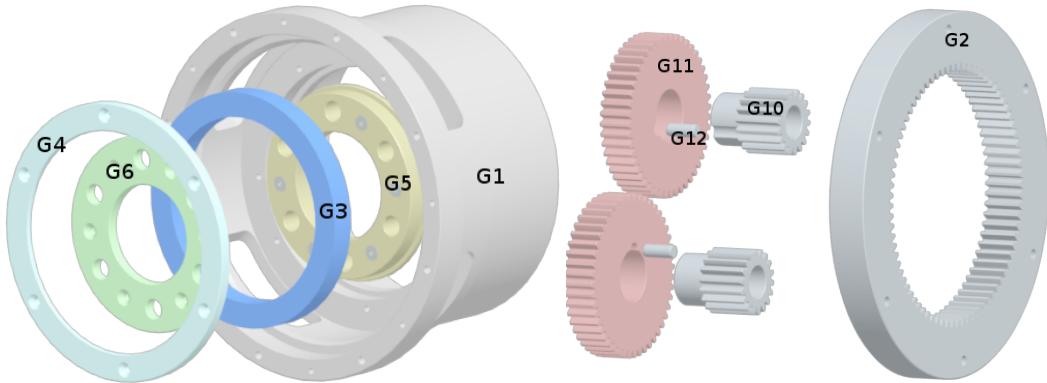


Figure 4.14: Assembly G's housing, main bearings, compound planets and annulus.

4.4.2 Compound Planetary Gear Assembly (G)

This assembly (Figures 4.13 and 4.14) significantly reduces the rotor angular velocity (by roughly 1:12 to 1:24, depending on gearing) using a single-stage compound planetary gear system. For a gear reduction of 1:20, G5/G6 will rotate roughly 20 times more slowly than M12, but with 20 times more torque.

G1 Housing. Encloses the planetary gear reduction stage to keep the gears clean of dust and dirt. Supports bearings G3 and H1, which are removable thanks to bearing clamps G4 and H2. Has holes for mounting the entire HypoSEA to another, larger assembly or test stand.

G2 Annulus gear. Interior tooth gear fixed to G1 with screws. Small planet gears G10 mesh with this.

G3 Planetary carrier bearing. Allows low-friction rotation of planetary carrier G5/G6 and all the shafts and gears attached to G5/G6.

G4 Planetary carrier bearing clamp. Holds bearing G3 in place with screws. Preferable to force-fitting if easier assembly and disassembly is desired.

G5 Planetary carrier. Three of shaft G7 are force-fit into this piece, as are three of shaft D8.

G6 Planetary carrier clamp. Holds G5 onto G3 with several screws. Preferable to force-fitting the bearings G3 in place if easier assembly and disassembly is desired.

G7 Long shaft. Supports planetary gears G10/G11 on one side of G5/G6. On the other side, it supports D5. Pieces on this shaft are held in place axially using a C-shaped clevis G8.

G8 Locking clevis. Keeps pieces rotating on shaft G7 from sliding axially.

G9 Bushing. Forced into gear G10 and G11 to reduce the friction caused by their rotation on G7. Can only be seen in Figure 4.16.

G10 Small planet gear. Typically 12-24 teeth, three of this component mesh with G2. Rigidly connected with G11 using locking pin G12. A bushing G9 is forced inside this piece to reduce its rotational friction with shaft G7.

G11 Large planet gear. Typically 24-48 teeth, three of this component mesh with M12. Rigidly connected with G10 using locking pin G12. A bushing G9 is forced inside this piece to reduce its rotational friction with shaft G7.

G12 Locking Pin. Force-fit into an axial keyway on G10 and G11 to rigidly mate the two pieces together.

4.4.3 Planetary Differential Gear System (D)

The differential assembly D (Figures 4.15, 4.16, and 4.17) is the only optional assembly in HypoSEA. In fact, in the HypoSEA-v2 this assembly is completely omitted. The purpose of assembly D is to act as a mechanical differential that takes the difference in angular position between planetary carrier G5/G6 and output link E1 and multiplies it by some gear ratio. This increased deflection results in H10 rotating around the center axis of the HypoSEA more than it otherwise would.

If assembly D's gear ratio is two to one, when rotor M2 is fixed, a 60 degree deflection of E1 results in a 120 degree deflection of H10. This is often desirable, as ± 120 degrees of compliant deflection is usually more than is needed for practical applications.

If the additional complexity of assembly D is not needed it may be omitted. In such a case, shaft H10 would be extended and connected directly to G5/G6, and all parts in assembly D would be removed.

Assembly D and its connections to assembly H may be seen in

D1 Annulus gear. This interior-tooth gear is attached to D2 with screws. It meshes with six of the D5 gears, and will rotate at twice the velocity of G5/G6 when D6 is motionless.

D2 Annulus carrier. Rotates on bearings D4, and supports D1. Also supports one end of shaft H10.

D3 Annulus carrier bearing clamp. Affixes D2 to bearing D4 without force-fitting, for easier assembly and disassembly.

D4 Annulus carrier bearing. Allows D2/D3 to rotate with little friction.

D5 Planetary gears. These six planetary gears mesh with D6 and D1; three are mounted on shafts G7, and three are mounted on shafts D8. Bushings D9 are pushed inside these to reduce friction when rotating on shafts G7 and D8.

D6 Fixed sun gear. Rigidly attached to H3 with screws, it meshes with the six D5 gears.

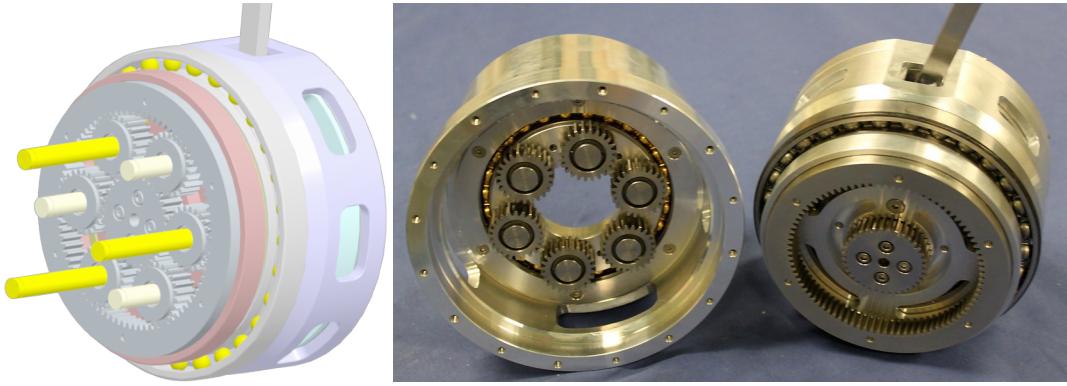


Figure 4.15: Differential assembly D split in half so that the fixed sun, freely rotating planets, and annulus may be seen.

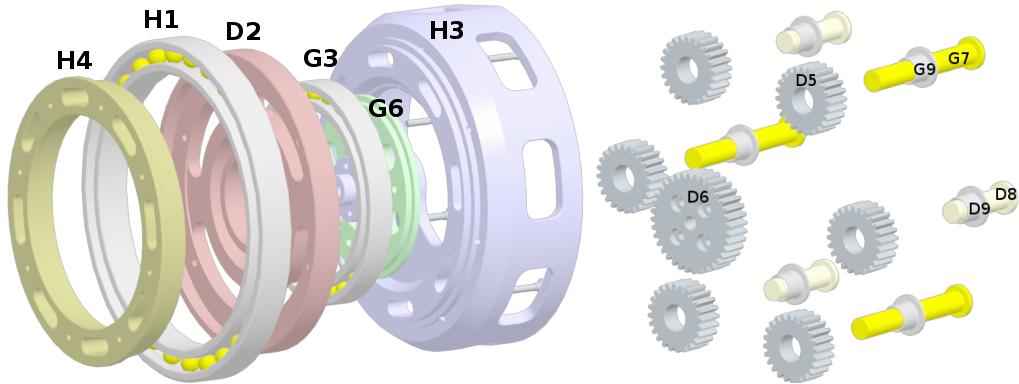


Figure 4.16: Left: Exploded view of differential assembly (D), showing gears, shafts, and bushings.



Figure 4.17: The planetary gears and planetary carrier from assemblies G and D. Note how three of the short shafts D8 pierce gears G5, while the long shafts G7 pierce G10/G11 as well as gears D5.

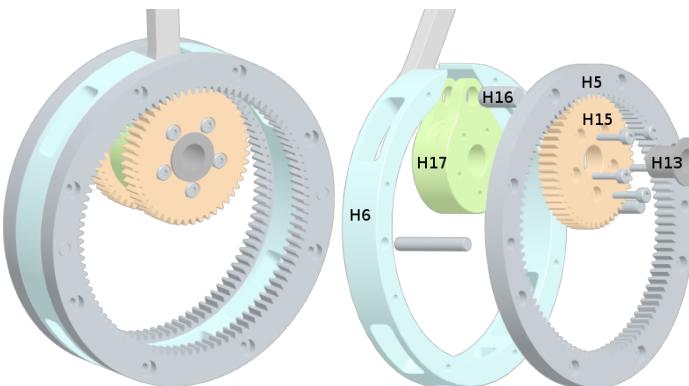


Figure 4.18: **Left:** Assembled hypocycloid gear assembly H. **Right:** Exploded view.

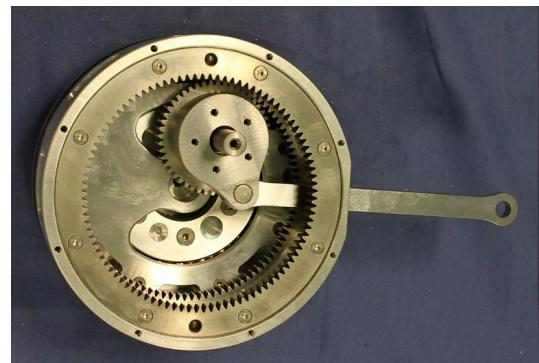


Figure 4.19: Photograph of the interior of the hypocycloid assembly showing how the rod H19 is pulled in a straight line.

D8 Short shaft. Not visible in the cross section of Figure 4.9, it can be seen instead in Figure 4.16. It is a short shaft on which three of the six D5 gears rotate. Like G7, it is force-fit into G5/G6.

D9 Bushings. Same as G9, and can also only be seen in Figure 4.16. These bushings allow gears D5 to rotate on their shafts with low friction.

4.4.4 Hypocycloid Assembly (H)

This assembly is critical to the functionality of the HypoSEA and can be seen in Figures 4.18 and 4.20. Assembly (H) converts an angular displacement into a linear displacement using the hypocycloid mathematics discussed in section 4.2.4. The relationship between the angular displacement and the deflection of the spring is nonlinear.

H1 Bearing. Connects assembly G and assembly H together, allowing the two pieces to rotate relative to each other. Must be a four-point contact bearing because it may experience moments and axial loading.

H2 Bearing clamp. Because there may be significant axial forces on bearing H1, this piece is needed to hold it in place.

H3 Complex piece. Supports gear D6 from only one side, acts as a mechanical travel stop to limit the rotation of D2/D3 to ± 120 degrees, and provides holes for alignment of pieces H5, H6, H11, and H4. Supported by H11.

H4 Double-sided bearing support. Clamps bearing D4 to H3, and also bearing H1 to H3.



Figure 4.20: Hypocycloid assembly shaft H10 and components that freely rotate on it.

H5 Hypocycloid annulus. Has twice the number of teeth as H16. Shear pins and screws connect two of these rigidly to H3 and H6.

H6 Hypocycloid annulus spacer. Because component H14 is irregularly shaped, it protrudes into the space between the two H5 pieces when pulled tight by H19. This c-shaped piece is needed to separate the two H5 pieces and provide space for H14 and H17 to pass through.

H7 Hypocycloid bearing. Lets H8 turn freely relative to H11.

H8 Hypocycloid planetary carrier. Supports one end of shaft H10, and provides a mounting place for encoder disk H18.

H9 Hypocycloid planetary carrier bearing clamp. Used to hold H8 tightly against the bearing H7 if there is any axial loading.

H10 Hypocycloid shaft. Supports gears H15 such that they remain in contact with annulus H5 as H8/H9 and D2/D3 rotate together. Held tightly against H8/H9 and D2/D3 by two machine screws H12 threaded axially.

H11 Hypocycloid housing. Supports bearings H7 and acts as connecting and aligning piece for E1, H20, and H3.

H12 Bolts. Holds shaft H10 tight against D2/D3 and H8/H9 to resist axial loads.

H13 Bushing. Same bushings as G9. Reduces the friction of the rotation of H15.

H14 Hypocycloid gear spacer. An egg-shaped piece that has an off-center hole through which shaft H17 is connected. The off center hole is placed exactly on the pitch diametral circle of hypocycloid gear H15, so that a hypocycloid mechanism is created that obeys the mathematics established in section 4.2.4.

H15 Hypocycloid gear. Rigidly connected to H14, and mesh with the H5 annuluses to create the hypocycloid mechanism. Bushings H13 are forced inside these gears to reduce friction with H10.

H16 Connecting rod. Connects spring E5 to H18. Held in alignment by a constant tension and thus never contacts H3 or H6.

H17 Short contact pin. Held in place axially by the two H15 gears. Transmits the tension load from H16 to H14.

H18 Optical encoder disk. Same type of part as M7. Used to measure the angular position of H8 relative to E1.

H19 Optical encoder disk electronics. Same type of part as M8. Used to measure the rotation of H18 relative to E1.

H20 Optical encoder mount plate. Supports H19 so that it is the proper distance from H18.

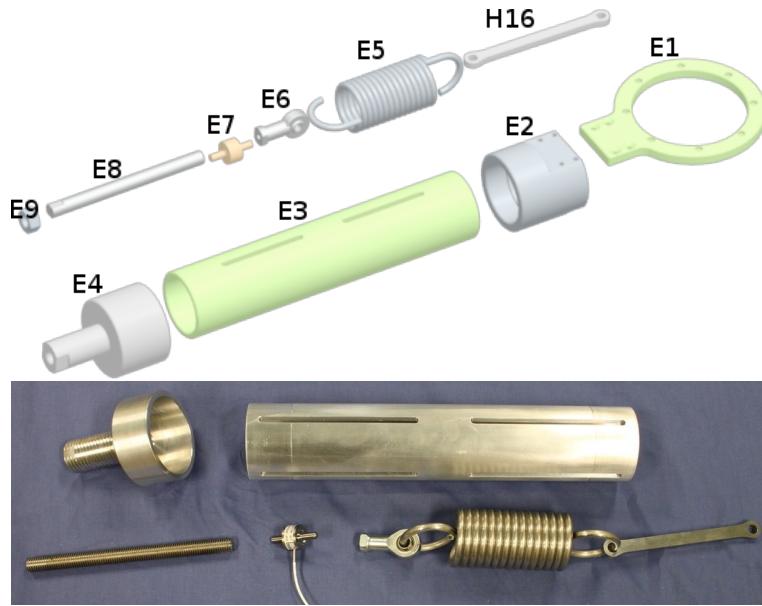


Figure 4.21: **Top:** Components of assembly E. **Bottom:** Photograph of many of the same components.

4.4.5 Elastic Element Assembly (E)

The elastic element assembly is simply an elastic element contained in a protective tube. As connecting rod H16 is pulled, it stretches a spring E5. A load cell E7 directly measures the tension in the spring. By turning E4 relative to E8, the pretension in the spring may be increased. Refer to Figure 4.21 as needed.

E1 *Mounting ring.* Used to rigidly affix E2 to H11 with several screws. When the HypoSEA is used as part of a larger system, the output link connects to this component.

E2 *Tube mount.* Connects E3 to E1.

E3 *Tube.* Used to contain the spring E5, it is held in place by the compressive load on both ends. In the prototype, grooves are cut into the tube to allow visual inspection of the spring.

E4 *Tube cap.* Rotates freely on tube E3, it is threaded such that E8 moves axially when turned, thus allowing manual pretensioning spring E5.

E5 *Spring.* A large steel tension spring. One end pierces E6, and the other end H16.

E6 *Spring eyebolt.* One end of the tension spring pierces this piece. Screws directly on to load cell E7.

E7 *Load cell.* Measures the tension in the spring. Two threads allow it to screw into E8 and E6.

E8 *Threaded tensioning rod.* Allows the spring to be pretensioned by up to 40mm deflection. Unscrewing this device allows assembly E to be disassembled.

E9 *Locking nut.* Used to tighten E8 against E4 and prevent any undesired rotation of E4 relative to E8.

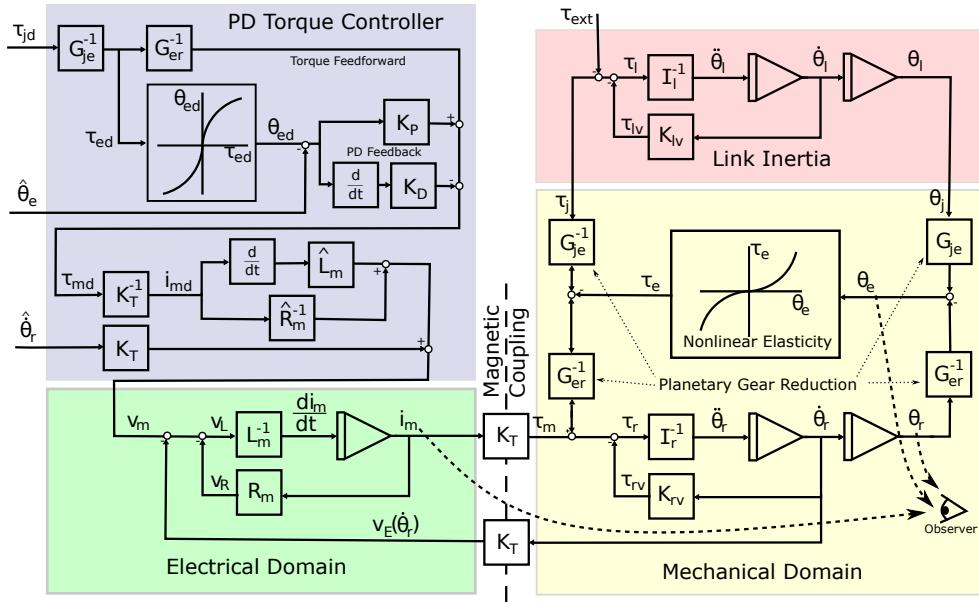


Figure 4.22: Block diagram of the HypoSEA controller and electromechanical structure. Torque constant K_T couples the electrical and mechanical domains.

4.5 HypoSEA Torque Controller

Since the HypoSEA is designed specifically for excellent torque control, the primary objective of the controller should be to produce the desired torque as precisely as possible. In this section, we will discuss how the control of the BLDC motor is performed. A block diagram of the elecromechanical system is shown in Figure 4.22.

Let's assume that we specify a desired joint torque τ_{jd} . To produce any torque, the nonlinear spring mechanism must be deflected. The appropriate amount of deflection is found by inverting equation 4.4 to get the desired elastic deflection θ_{ed} . This was done numerically and a look-up table was created for the purposes of speed.

In the following we will distinguish measured or estimated values from real values by using hats $\hat{\cdot}$ denote our best estimate of a measured, noisy quantity. How these estimates were computed is the subject of Chapter 6.

$$\theta_{rd} = -G_{er} \left(\hat{\theta}_j - (G_{je}^{-1} \theta_{ed}) \right) \quad (4.25)$$

A desired motor torque τ_{md} is then computed from a PD controller with an offset to move the actuator towards desired position θ_{ed} .

$$\tau_{md} = K_P(\theta_{ed} - \hat{\theta}_e) - K_D(\dot{\theta}_{ed} - \dot{\hat{\theta}}_e) + G_{je}^{-1} G_{er}^{-1} \tau_{jd} \quad (4.26)$$

where K_P and K_D are the proportional and derivative gains. Finally, the desired motor current i_{md} is found from the desired motor torque using the torque constant. The magnetic coupling is essentially perfect (Mevey, 2006), although the torque constant K_T may vary slightly for different orientations of the stator and rotor and produce ripple torque.

$$i_{md} = K_T^{-1} \tau_{md} \quad (4.27)$$

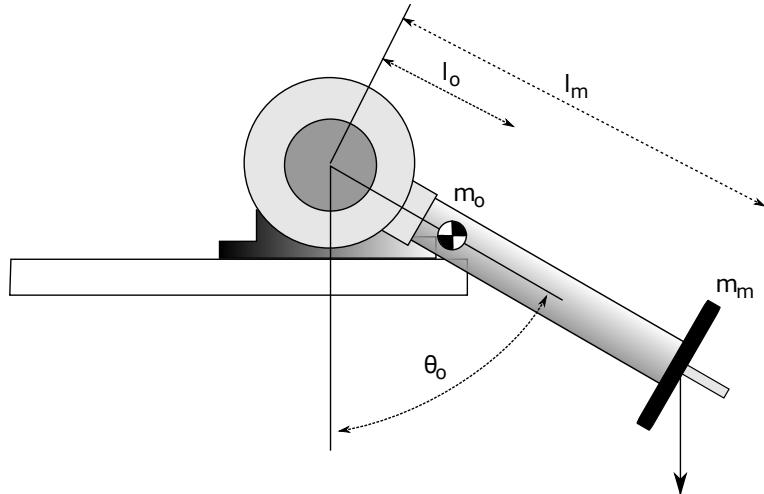


Figure 4.23: Schematic of the HypoSEA-v1 torque test setup mounted on a table, showing the positions of the CoM of the output link and the heavy load mass.

The voltage to create this torque is found by modeling the motor as an RL circuit with a back-emf proportional to the angular velocity.

$$v_{md} = R_m i_{md} + L_m \frac{d\hat{i}_m}{dt} + K_T \hat{\theta}_r \quad (4.28)$$

The motor voltage v_{md} was then sent to the motor driver board, which selected a motor PWM duty cycle ψ_m based on the estimated battery voltage \hat{v}_B .

$$\psi_m = \frac{v_{md}}{\hat{v}_B} \quad (4.29)$$

The motor driver board also handled the six-step trapezoidal commutation pattern for the BLDC motors.

4.6 HypoSEA-v1 Experimental Results

The actuator works well and in laboratory tests shows promising torque control capability at force control levels four orders of magnitude apart. That is, the system is able to resolve torque increments less than 0.02Nm – exactly how much less depends on the minimum spring pretension and stiction – yet is still able produce torques in excess of 70Nm.

4.6.1 Mechanical Parameters

The methods used for system identification of the HypoSEA-v1 and HypoSEA-v2 are beyond the scope of this chapter and will be presented in Chapter 6. For now, we merely succinctly summarizes the results in Table 4.1. It is also convenient to put the mechanical performance of the HypoSEA-v2 also on this table, for more convenient comparisons between the two actuators.

There are several comments that must be made about the quantities in Table 4.1. Items marked with a star are based on extrapolation from the actuator running at higher powers than the motor controller board can actually provide. The motor control board is the limiting factor in our case, not the motor or mechanical system. That

Table 4.1: HypoSEA-v1 and -v2 Electromechanical Parameters

Description	HypoSEA-v1	HypoSEA-v2	Unit
Actuator mass	8.2	2.883	kg
Total mass, including stand	8.9	-	kg
Actuator diameter	14.0	12.4	cm
Longest exterior linear dimension	67	21	cm
τ_j	Maximum tested joint torque	71	Nm
	Maximum theoretical joint torque*	126	Nm
	Minimum resolvable torque**	<0.02	Nm
$\hat{\theta}_j$	Maximum controllable output velocity	10.2	10.6 rad/s
G_{jr}	Rotor-joint Gear Reduction	18. $\bar{3}$	17
G_{er}	Elasticity-Rotor Gear Reduction	12.8 $\bar{3}$	17
G_{je}	Joint-Elasticity Gear Reduction	$\frac{10}{7}$	1
m_o	Output link mass	5.3	n/a kg
l_o	Output link CoM distance from rotor center	11.5	n/a cm
	Encoder resolution	8192	counts/rev
	Motor poles	12	12 poles
	Motor power constant	0.182	Nm/ \sqrt{W}
K_T	Motor torque/back-emf constant	0.30	V/rad/s
R_m	Motor resistance	2.64	ohms
L_m	Motor inductance	2.76	mH
T_m	Motor electrical time constant	1.04	ms
K_e	Linear Spring Constant	10.09	N/mm
l_p	Maximum spring pretension	40	mm
l_d	Maximum spring deflection	72	mm
r_e	Hypocycloid gear radius	24	mm
V_e	Max potential energy in spring*	42.3	J

said, it is believed that the numbers presented in the table slightly *overestimate* the true values. The maximum verified torque was about 71Nm at 48V – whether or not the 126Nm of peak torque could be reached at a higher supply voltage is somewhat dubious.

The minimum resolvable torque is marked with two stars because it is somewhat difficult to precisely specify, as it depends upon the pretension of the HypoSEA's spring and the resolution of the encoder disk. The value given was typical for a small pretension – unfortunately due to friction being nonlinear and many times larger than the resolvable forces, it is difficult to verify that this calculation is correct.

Finally, the maximum controllable output velocity is the fastest the joint can spin without any spring deflection. If spring deflection is allowed, momentary joint velocities much faster than this limit may be created.

4.6.2 Maximum Torque

One of the first experiments performed on the HypoSEA-v1 was to measure the maximum torque the actuator could apply. This was achieved by having the arm lift a heavy load mass very, very slowly in the experimental setup shown in Figure 4.7, left, and Figure 4.23. The angle at which the link came to rest indicated the maximum torque that the motor could generate – although this was not the same as the maximum torque that the motor could resist, which is slightly higher due to cogging torque effects from the motor. The motor is turned very slowly so that the effects of momentum may be ignored.

The torque on the output link resulting from gravitational forces are:

$$\tau_{og} = -g(m_m l_m + m_o l_o) \sin(\theta_o) \quad (4.30)$$

where $\theta_o = 0$ corresponds to the orientation where the output link arm hangs straight down, m_o and m_m are the mass of the output link and load mass, and l_o , l_m are the distances from the center of rotation to the output link's CoM and the load mass.

The maximum torque value (with a 40V supply) was estimated by hanging a $m_m = 11\text{kg}$ mass $l_m = 50\text{cm}$ from the shaft center. The link itself weighs $m_o = 5.3\text{kg}$ and is centered $l_o = 11.5\text{cm}$ from the motor axis. In the first test, the mass swung out to $\theta_o = 1.32$ radians from the vertical. From these values it is straightforward to estimate that there must be at least 58.3Nm of torque produced by the electrical system. Let us try to confirm this electrically.

A phase-phase resistance of 2.87 ohms was measured for the EmoTec 2303 brushless DC motor, although it varies with stator temperature. When the voltage across the two motor phases is measured with an o-scope, there is typically a voltage about 3.1V lower than the supply voltage. This is believed to be the cumulative effect of the electronics voltage drops.

Given an applied voltage of 40V minus 3.1V electronics drop, this should have produced a current of 11.6 amps. This was very close to what was seen on the power supply's current meter, although it began to drop slightly as the motor heated. A current probe attached to the oscilloscope measured 12.1A. Presumably the difference is being lost in the motor driver electronics, which also dissipate some few hundred millamps when running.

Because the conversion from current to torque is essentially perfect (assuming that the torque constant $K_T = 0.30\text{Nm/A}$ is correct on average), then 11.6A must have created 3.48Nm of torque at the rotor. The HypoSEA-v1 has a gear reduction of $G_{je} = 18.3$. Ignoring friction, this would result in 63.7Nm of torque on the output link.

With an estimated friction on the device being 1-2Nm, the difference in torque estimated electronically and mechanically was just a few percent. Because the device was moving extremely slowly, we may safely ignore viscous friction.

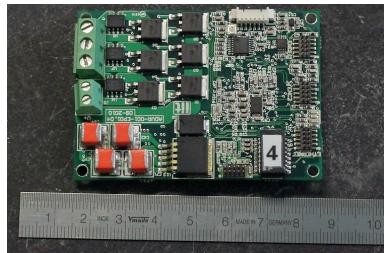


Figure 4.24: Photo of the motor board drivers used to control JIMI.

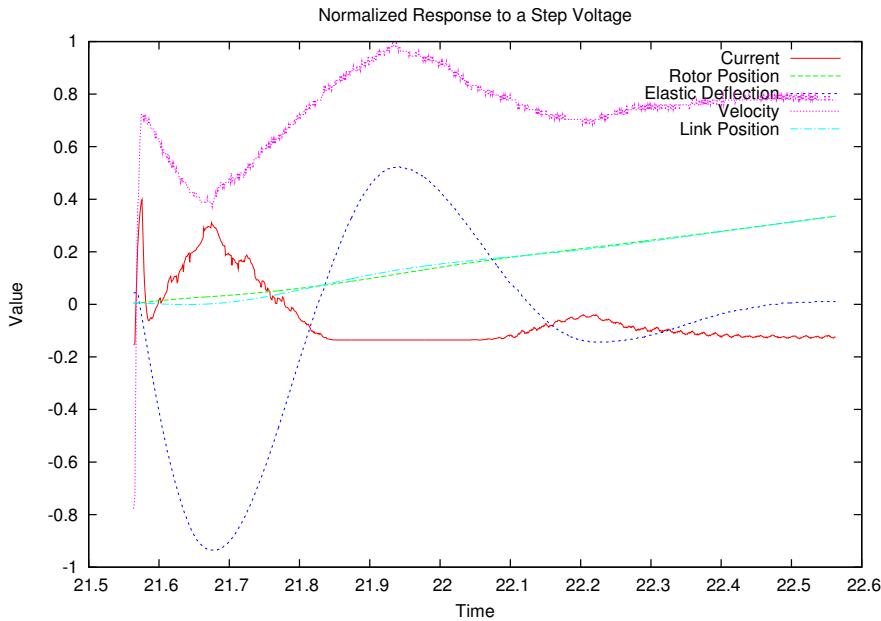


Figure 4.25: Measured open loop voltage step response with only the output link inertia. The figure is normalized for visualization purposes. Note how the effect of the elastic deflection on all other quantities can clearly be seen. The actuator was oriented horizontally so that gravitational torques are not present.

In conclusion, for a 40V supply, we estimated 63.7Nm of torque produced electrically, yet mechanically only measured 60.1Nm of torque. If we assume 1-2Nm of friction, the experimental results are reasonable.

What about using a higher supply voltage to produce a larger torque? The limitation on creating higher continuous torques was first electronic and secondarily thermal. Unfortunately, the motor driver could not support supply voltages higher than 48V, and this fundamentally limited the maximum momentary test torques that could be generated by the HypoSEA designs. When the supply voltage was 48V, higher momentary torques approximately 71Nm were tested. At this voltage, the stator windings begin to slowly heat and their resistance correspondingly increased. However momentary torques up to 126Nm should still be possible if the electronics could support much higher voltages.

4.6.3 Bandwidth

Open loop data taken from the HypoSEA-v1 is shown in Figures 4.25 and 4.26. Figures 4.27 and 4.28 show the frequency response of the simple PD torque control system. Figure captions are self-explanatory. Keep in mind that since the effective elasticity of the HypoSEA is nonlinear, these frequency spectrum plots are only valid for the particular torque offset and torque amplitude at which the signal was taken. They are intended as qualitatively interesting more than quantitatively accurate.

Figure 4.26: Measured open loop voltage frequency response for low magnitude sine waves. Magnitudes are not to scale. Note the mechanical resonant frequency is approximately 2Hz, the frequency at which most of the work in human locomotion occurs, and that rotor work is reduced at this resonant frequency. The actuator was oriented horizontally so that gravitational torques are not present.

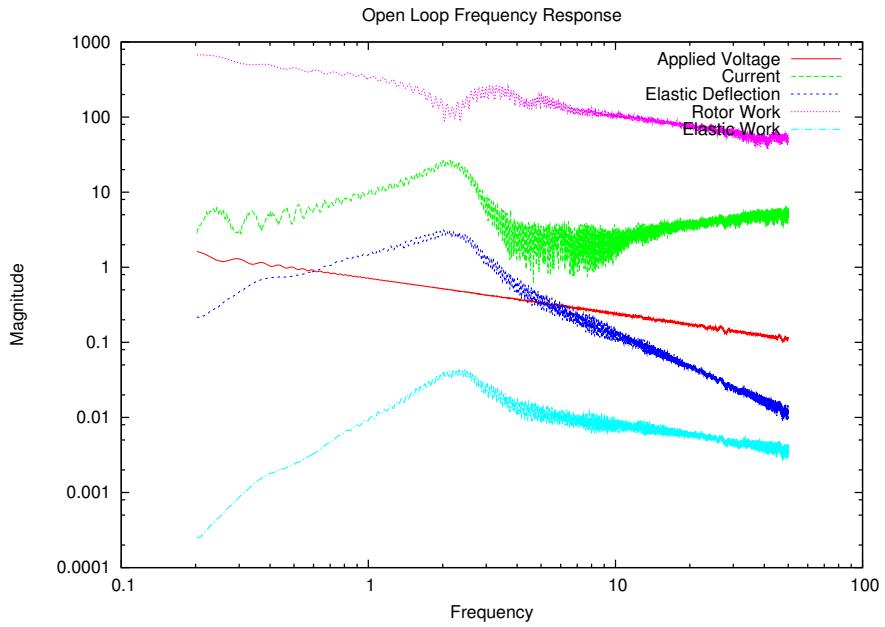
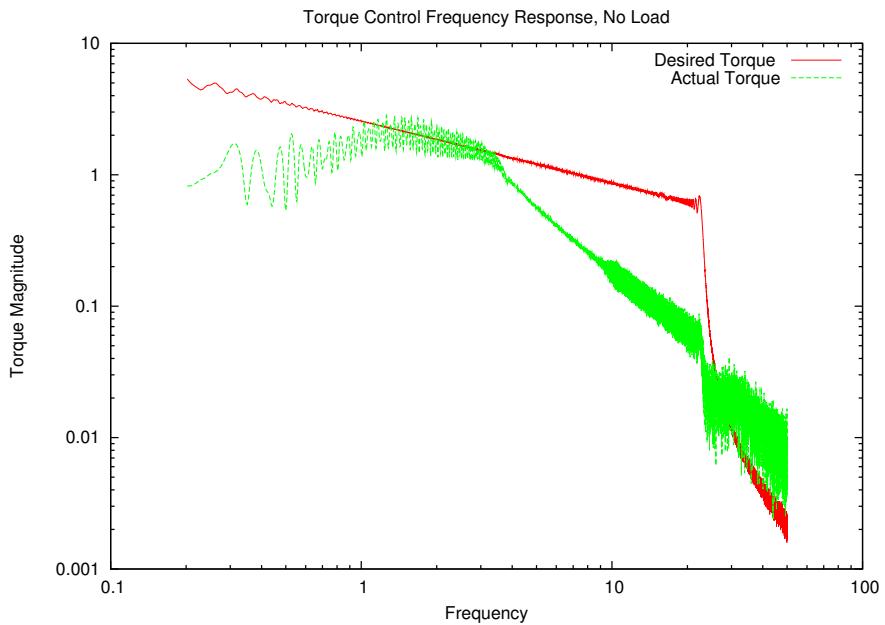


Figure 4.27: Measured torque control frequency spectrum of the HypoSEA-v1 when unloaded. Note the ability to accurately track torques is relatively limited. Below 1Hz, the maximum rotor velocity limits the torque tracking ability, and above 3Hz the low effective stiffness of the hypocycloid mechanism prevents torques from being developed quickly. The actuator was oriented horizontally so that gravitational torques are not present.



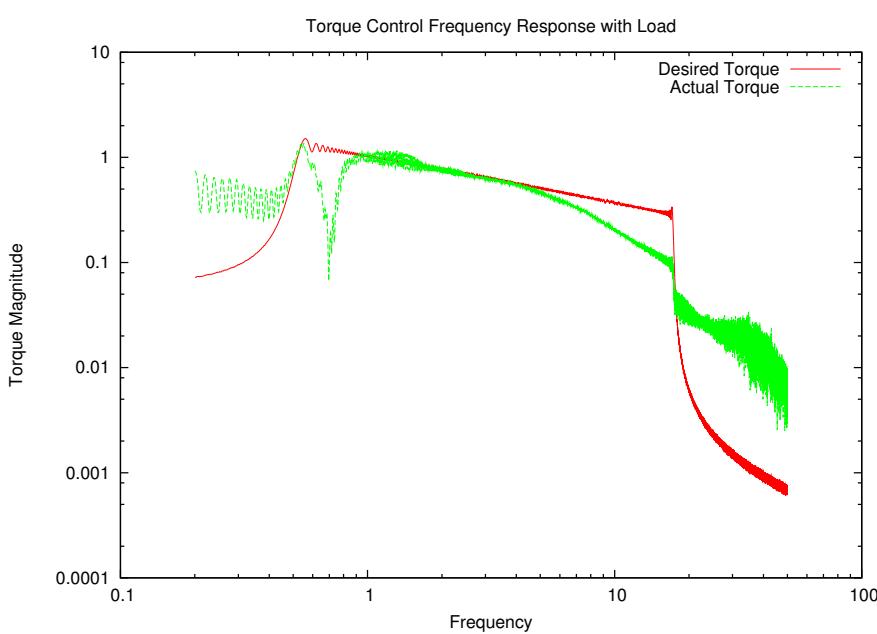


Figure 4.28: Measured torque control frequency spectrum of the HypoSEA-v1 when loaded with a $m_m = 2\text{kg}$ mass on the output link to create a constant offset gravitational torque. The orientation of the link was $\theta_0 = \pi/2$. Because of the hypocycloid mechanism's increased effective stiffness away from the zero-torque equilibrium, the small torque signal frequency response has increased to $\sim 6\text{Hz}$. For higher torques, the small-torque bandwidth exceeds 10Hz.

4.6.4 Energy Recovery with the HypoSEA-v1

We use electric motors to convert electricity into mechanical motion every day – yet we often forget that ordinary DC motors can also convert mechanical energy back into electrical energy. The energy recovery capabilities of the HypoSEA designs was discovered by accident. While doing experiments near the resonant frequency of the link inertia and spring, sparks began to arc across the 3mm gaps on the motor driver board's supply capacitors.

The cause of the sparking was soon discovered. When the motor spun quickly, the back-emf generated on the motor was higher than the supply voltage, meaning the current flowed through the transistors' protective diodes and into the supply. But the lab power supply being used would not sink current, and thanks to the highly inductive motors the voltage on the motor control board capacitors was raising to dangerous levels, creating arcs of electricity. Surprisingly, this did not harm the motor driver boards, although the faint smell of ozone proved alarming to several electrical engineers in the vicinity.

To measure how much energy was being recovered, the motor controller board was connected to a 13V battery and a $m_m = 2.0\text{kg}$ mass was attached $l_m = 0.50\text{m}$ from the center of the output link. The link itself has a center of mass weighing $m_o = 5.3\text{kg}$ at $l_o = 11.5\text{cm}$ from the center. The link was placed at an orientation of $\theta_0 = \pi$ radians, pointing straight up like an inverted pendulum. The PWM duty cycle was set to 0.01% to ensure a current path through the H-bridge, and gravity was allowed to swing the pendulum down quickly. An oscilloscope's current probe was used to measure the flow of current into the battery, as well as its voltage. As expected, as soon as the EMF on the rotor became higher than the supply voltage, the actuator began to convert mechanical energy to electrical energy (Figure 4.29).

The total gravitational energy during the swingdown was approximately 32.1J. If a dry friction of 2Nm acted on the joint as it swung over the π radians, then at least 6.3J of energy were dissipated by friction. The joint must spin at about 2.36rad/sec to produce 13V of EMF on the rotor. If the total output inertia was 1.64kgm, at least 4.6J of energy was needed to accelerate the output link to the velocity at which the rotor could begin to generate electricity. This means that the total energy available to be absorbed by the HypoSEA-v1 was

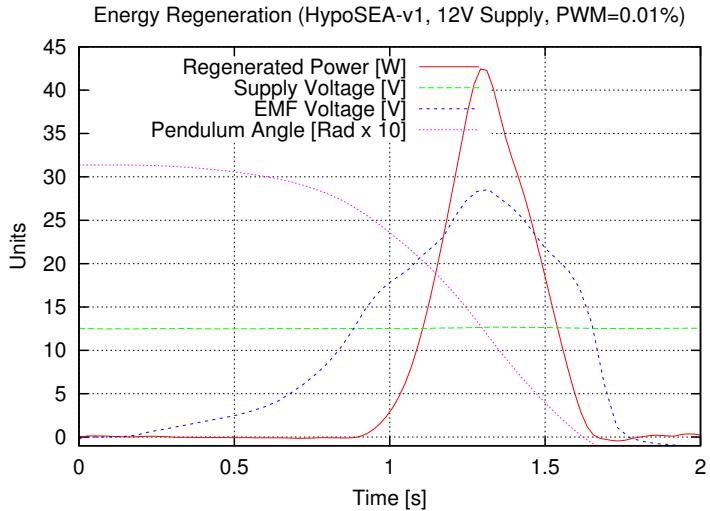


Figure 4.29: HypoSEA-v1 regeneration experiment results. As the pendulum swings down, the joint angular velocity increases and the EMF voltage produced by the rotor begins to rise. When it exceeds the supply voltage, it begins to push energy into the power supply through the H-bridge diodes. The efficiency of conversion was approximately 65%.

$32.1 - 6.3 - 4.6 = 21.0\text{J}$ of energy. By numerically integrating the power generated by the motor (4.29), we compute a total of 13.7J of energy that was actually recovered by the rotor – an efficiency of 65%.

Although further experimentation along this line of research looked interesting, it was decided to wait until the HypoSEA-v2 was constructed to study electrical generation efficiency extensively, as the -v2 device was expected to have lower coefficients of dry friction on the output than the -v1 device. Spinning the motor faster and using dedicated reversible power electronics could push also the generation efficiency higher.

4.6.5 Motivation for Redesign

Two major problems were encountered during the testing of the HypoSEA-v1: an incorrect differential unit gear ratio in assembly (D) and the limited motor driver voltage/current capacity. The actuator was also extremely difficult to assemble and required 30-40 hours of work to make everything fit properly.

There was also a significant design mistake in the HypoSEA-v1: although an output-elastic gear reduction $G_{je} = 2.0$ was intended, due to an algebraic mistake the actual relation $G_{je} = \frac{10}{7}$ was created. This strange number was discovered during calibration process, and not during the design process, when it could have been corrected.

As for the motor controllers – a custom motor control board developed at IIT for use on the iCub robots (Metta et al., 2008) – they were fraught with minor problems. They report incorrect current values above 6A, perform the H-bridge PWM switching improperly, and use extensive velocity filtering and smoothing techniques that introduce unwanted and significant phase lag. Most importantly, their maximum current and voltage limitations prevent the HypoSEA-v1 from really being tested to its maximum. Despite these limitations, it was decided not to switch to a different motor control board because the firmware of the boards is modifiable because the boards were developed in-house at IIT. The potential to completely customize their behavior was thought to be a feature crucial to the success of JIMI, regardless of bugs.

There was one other major reason to redesign the HypoSEA-v1: its ponderous weight. At a voluptuous 8.2kg in mass, it was far too heavy to be used in a hopping robot. For the second version of the HypoSEA, reducing the mass was a priority. Let us now turn to the fruits of this weight reduction.



Figure 4.30: **Left:** The assembled HypoSEA-v2 weighs only 2883g. **Right:** Components of the HypoSEA-v2.

4.7 The HypoSEA-v2

The HypoSEA-v2 (Figure 4.30) is a refinement of the HypoSEA-v1. The following changes were implemented with the goal of reducing actuator mass as much as possible.

1. Because the limiting factor for the HypoSEA-v1 was the voltage limits of the motor control boards, it made sense to revise the design peak torque to be only 70Nm. This reduce the required load safety factors of the gears and made things much lighter.
2. All aluminum parts were reduced in size by slightly decreasing the maximum outside diameter of the actuator.
3. The large steel tension spring E5 was replaced by a compact fiberglass compression spring.
4. The load cell E7 provided little useful information in practice and was removed.
5. The module of the gears in assembly G was reduced from 1.0mm to 0.8mm.
6. Assembly D was completely removed and the spring stiffness was doubled. This stresses the components in assembly (H) more, but not excessively so thanks to the reduced design peak torque.
7. All major loads now rest on a single large four-point-contact bearing H1.
8. Lightening holes were added wherever convenient.
9. Some shafts previously made in steel are now made in aluminum.

The result of all of these changes was significant; up to the 70Nm torque limit, functionally the HypoSEA-v2 is very similar to the HypoSEA-v1, yet the mass was reduced by 65%. This impressive reduction was made possible by closely monitoring the masses of all components as they were redesigned. As is shown in Table 4.2, the system masses estimated using CAD tools corresponded very closely with the real parts. Because the design of the HypoSEA-v2 is simpler and the number of parts was reduced, not all components from the HypoSEA-v1 design are present.

Referring to Table 4.2 as needed, we note that the CAD estimates were remarkably accurate. The 31g discrepancy between the two values is believed to be due to grease and bolt masses. Let us also make a few conclusions about the mass of the actuator.

Table 4.2: Predicted and actual HypoSEA-v2 component masses.

Part	Description	Material	g/Ea.	Qty.	Predicted	Real
M1	EmoTec 2303 stator	-	388	1	388	388
M2	EmoTec 2303 rotor	-	294	1	294	294
M4	Motor enclosure	Alum.	96	1	96	97
M5	Motor shaft	Alum.	23	1	23	23
M6,M10	SKF-607 bearings	-	8	2	16	16
M7/H18	Optical encoder disk	-	4	2	8	9
M8/H19	Optical encoder board	-	5	2	10	10
M9	Shaft spacer	Alum.	2	1	2	2
M11	Shaft lock nut	Alum.	3	1	3	2
M12	Sun gear, 12 teeth	Steel	7	1	7	7
G1	Enclosure	Alum.	119	1	119	119
G2	Annulus, 80 teeth	Steel	149	1	149	147
G3/H7	SKF 61805 bearings	-	20	2	40	42
G5	Planetary Carrier 1	Alum.	20	1	20	17
G6	Planetary Carrier 2	Alum.	34	1	34	33
G7	Support shafts	Alum.	12	2	24	23
G9	Bushing, LFZF-8-8	-	2	6	12	17
G10/G11	Planet gears, 20/48 teeth	Steel	53	3	159	169
G13	Carbon fiber mount	Alum.	104	1	104	104
E2	Spring mount	Alum.	35	1	35	34
E4	Spring Tensioner	Alum.	27	1	27	31
E5	Spring: SWR-46-70	-	158	1	158	146
H1	Kaydon K09008XP0 bearings	-	130	1	130	115
H2	Bearing sidewall	Alum.	21	1	21	22
H3	Hypocycloid enclosure	Alum.	138	1	138	136
H5	Annulus, 96 teeth	Steel	151	2	302	298
G7/H10	Hypocycloid shaft	Steel	36	1	36	37
H11	Enclosure	Alum.	115	1	115	115
H13	Bushing, LFZF-10-15	-	5	2	10	10
H14	Hypocycloid attachment	Alum.	26	1	26	27
H15	Hypocycloid gear, 48 teeth	Steel	60	2	120	110
H16	Tension rod	Steel	67	1	67	48
TOTAL					2852	2883

1. The Emotec 2303 motor weighs about 850-1000g in usable form, regardless of the type of speed-reducing transmission that is used on it. Although itself it only weighs 650g, a practical design needs to include bearings, encoders, a shaft, and a minimal enclosure.
2. The planetary gear reduction stage (assembly G) weighs approximately 400g-600g. A comparably strong harmonic drive has no backlash and weighs slightly less¹, but the efficiency and backdrivability of the planetary system are superior. Also, planetary systems do not produce axial loads like a harmonic drive does.
3. The hypocycloid mechanism (assembly H and E) weighs about 1050-1300g, including sensing and the spring. Much of this mass is in the heavy steel annulus gears which must withstand repeated 70Nm load torques. Storing 25J of energy, it has an energy storage density comparable to existing approaches (Wolf & Hirzinger, 2008) for creating a nonlinear elasticity, although it cannot achieve similarly high torques. The hypocycloid mechanism is easily adjustable and has an excellent coefficient of restitution.

4.7.1 Component Design Changes

Although the geometries of most components changed, the orientation and function of components from the HypoSEA-v2 are sufficiently similar to the HypoSEA-v1 to merit using the same component identification system. This allows a simpler comparison between components from each actuator. A cross section is shown in Figure 4.31.

Some components from the HypoSEA-v1 were completely omitted in the -v2 design because of the complete removal of assembly D. In other cases, two components were merged into one, reducing the part count. Only one additional new component was added:

G13 Carbon Fiber Mount. This optional aluminum piece provides a large surface area that may be embedded in carbon fiber. Without this component, any carbon fiber monocoque structure that mounts directly on M4 will experience significant stress concentrations at the bolts.

4.7.2 HypoSEA-v2 Transmission

The transmission of the HypoSEA-v2 is simplified with respect to the HypoSEA-v1. The first stage of the planetary gear reduction uses the same equation as before, which we reprint here for convenience.

$$\begin{aligned}\omega_c &= \left(\frac{P_1 A_2}{P_2 S_1} + 1 \right)^{-1} \omega_{s_1} \\ &= G_{er}^{-1} \omega_{s_1}\end{aligned}$$

However, the number of teeth on the S_1 and A_2 gears have been changed such that

$$\begin{aligned}P_1 &= 12 \\ P_2 &= 48 \\ S_1 &= 20 \\ A_2 &= 80\end{aligned}$$

¹For example, the CSD-25 produced by Harmonic Drive Inc. has 50:1 reduction, 69Nm repeated torques, and weighs 240g plus the mass of the enclosure.

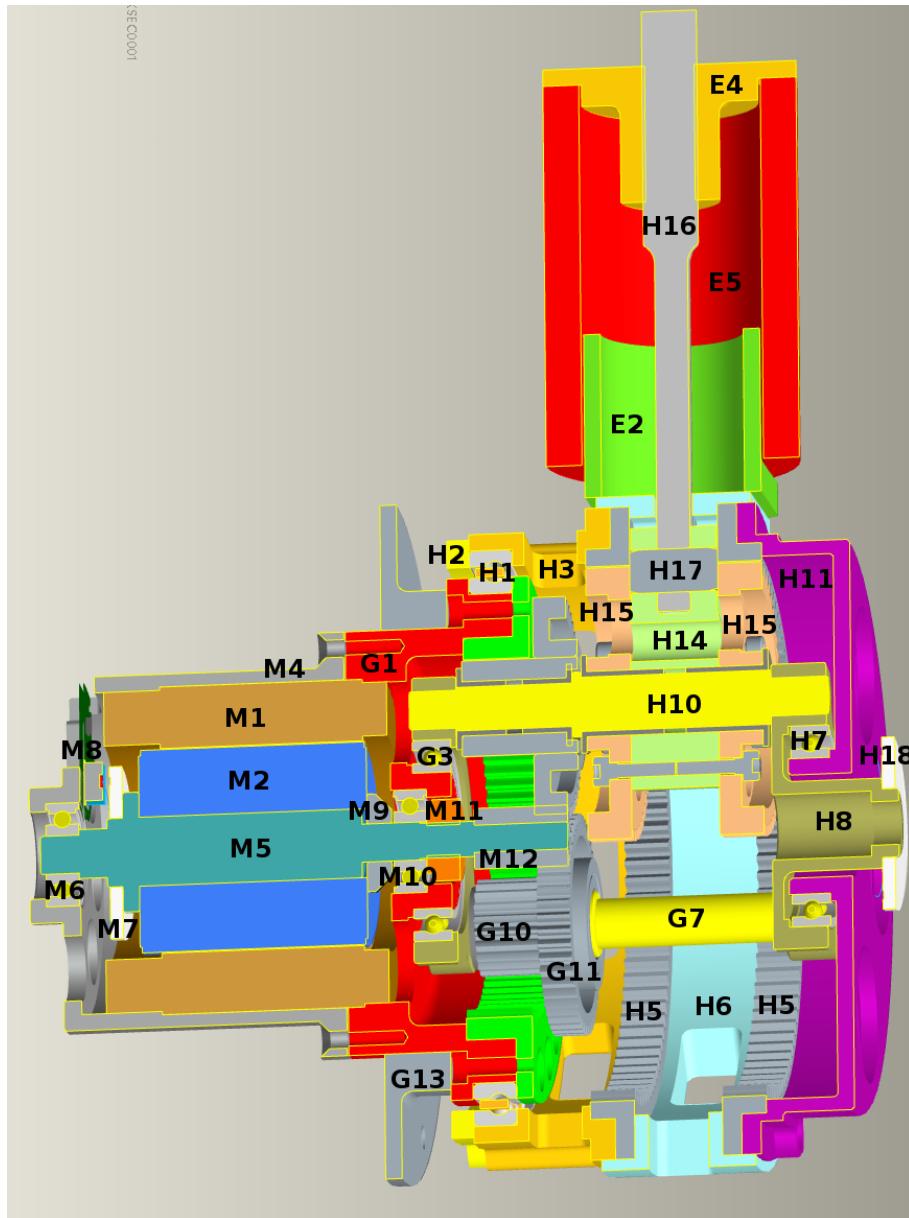


Figure 4.31: Labeled cross section of the HypoSEA-v2.

Basic arithmetic yields $G_{or} = \left(\frac{12}{48} \frac{20}{80} + 1 \right)^{-1} = 17$, a more convenient number than for the HypoSEA-v1.

Importantly, there is no second-stage planetary reduction for the second version of the HypoSEA. Therefore, $G_{je} = 1$ and

$$\theta_j = G_{er}^{-1} \theta_r - \theta_e \quad (4.31)$$

4.7.3 Stress Analysis

Because many of the components of the HypoSEA-v2 are much thinner and weaker than their corresponding parts of the HypoSEA-v1, a very rudimentary stress analysis was performed on key components using a finite element analysis tool. Figure 4.32 shows the stress concentrations of a sample component (H14).

Although only tested to $\tau_j = 65\text{Nm}$ at present, the actuator has not yet had a mechanical failure due to stress. The maximum strengths of the mechanical components are thus unknown. It seems most likely that the components that will fail first will be M5, G10, H1, or H14, as the stresses on these parts are the highest. The potential for metal fatigue is also unknown.

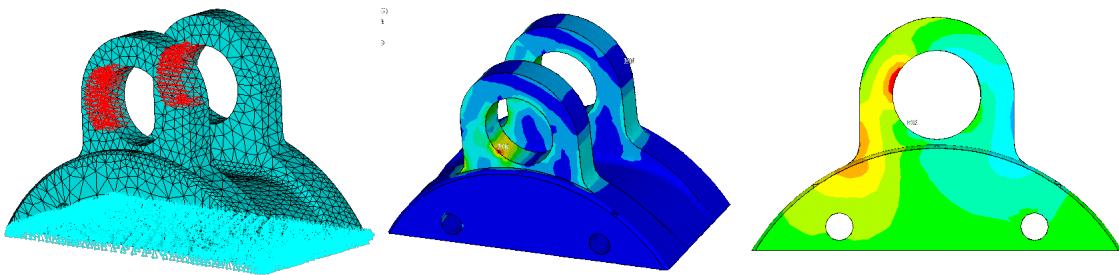


Figure 4.32: Screenshots of a finite element analysis tool showing the stress concentrations of HypoSEA-v2 component H14. **Left:** An example load direction is shown. **Center:** Stresses at 50% of peak design torque. **Right:** Stresses at 100% peak design torque (60Nm).

4.8 HypoSEA-v2 Experimental Results

Due to time constraints, the performance of the HypoSEA-v2 has not been fully characterized at the time of writing. At this preliminary stage, it appears to function well, but has not extensively been tested beyond preliminary force control tests and rudimentary hopping of JIMI (Chapter 8). The important measured parameters of the actuator are shown on Table 4.1.

4.8.1 HypoSEA-v2 Bandwidth

The resonant frequency of the HypoSEA-v2 can be analytically estimated from the rotor mass, transmission, and effective series elasticity. The incremental stiffness of the HypoSEA mechanism with significant pretension is about 75Nm/rad.

$$K_e = 75\text{Nm}$$

The rotor inertia is hard to estimate. The datasheet of the Emotec 2303 lists $5.9 \times 10^{-5} \text{ kg m}^2$, but the gears aren't light, so we will double that figure and estimate roughly at 1.2×10^{-4} . This works out to an equivalent rotor inertia of

$$I_r = 17^2(1.2 \times 10^{-4}) = 0.03468 \text{ kg m}^2$$

The natural frequency of the system is thus

$$\omega_0 = \sqrt{\frac{K_e}{I_r}} = \sqrt{\frac{75}{0.03468}} = \sqrt{2163} = 46.5 \text{ rad/s} = 7.40 \text{ Hz}$$

With higher pretension and less rotor inertia, it might be even higher. At higher deflections, the incremental stiffness will also improve.

The system will have better force control up to $\sqrt{2 \frac{K_e}{I_r}} = 65.7 \text{ rad/s} = 10.5 \text{ Hz}$. Up to this frequency, performance of the HypoSEA-v2 is improved thanks to the addition of the spring.

4.8.2 Mechanical improvements to the HypoSEA v2

At the end of any engineering project, possible improvements to the design become obvious. Some future improvements to HypoSEA-v2's design include:

1. Using looser tolerances to make certain parts of the actuator easier to assemble.
2. Using helical or double-helical gears for quieter operation.
3. Using shear pins on component H5 to reduce shear loads on the machine screws.
4. Moving the compression spring E5 internally to further reduce actuator size.
5. Cutting off the bottom half of H5 to save weight, since only the top ± 90 degrees of teeth can physically be used.
6. Using plastic materials instead of aluminum for parts which experience only low stress concentrations.

More extensive weight reduction of most components HypoSEA-v2 is also possible, but would require more significant engineering and manufacturing efforts, particularly in the field of stress analysis. As a first step, it would be informative to test one actuator until failure to get a real-world data point about the weakest components in the design.

4.9 Conclusions and Future Work

The HypoSEA-v2 represents a clear improvement of HypoSEA-v1 – not only was the mass significantly reduced, there are fewer components and it is easier to assemble. The moment of inertia of the output was greatly reduced, and the system functions well in initial motion tests, although it has not been stressed to design levels yet.

There remains considerable future work to be done. A proper system identification of the HypoSEA-v2 must be performed, and a better model developed from that. A more sophisticated gain-scheduling controller (Rugh & Shamma, 2000) which fully utilizes the nonlinearity of the mechanical system would be very useful, but it is difficult to achieve with the existing motor control boards. A report on the longevity of the actuator in actual operation and failure modes would also be important for establishing the commercial applicability of the device.

Despite all this effort, the HypoSEA still does not approach the elegant performance levels of animal muscle tissue. As a simple comparison, let us estimate the requirements for human-equivalent performance. A human skeleton typically weighs about 14% of body weight (Reynolds, 1977). Average muscle mass for women and

men is 36% and 42% of body mass, respectively (Marieb & Hoehn, 2007). Human muscle does positive work at a maximum rate of about 94W/kg, depending on muscle type, and tendon stores typical stresses of 700J/kg, although more can be stored temporarily with some risk of tissue damage (Alexander, 2003).

Neglecting scaling effects, this suggests that for a biomimetic 10kg robot with performance similar to a human, the skeleton should weigh about 1.5kg, the actuators 4kg, and the tendons should store several hundred J/kg. The actuators should also be capable of about 400W in total of continuous work. Unfortunately, when implemented in the robot JIMI (Chapter 8), two 111W HypoSEA-v2s weigh 5.8 kilograms in total – a factor of three worse than animal tissue even before batteries masses are included. Similarly, the HypoSEA-v2's hypocycloid mechanism has an energy density of approximately 25-50J/kg, roughly ten times worse than animal tendon.

CHAPTER 5

Centroidal Momentum Control of Legged Robots

This chapter presents a dynamic balancing controller for the robot JIMI. This controller relates closely with Chapters 3 and 4, as the dynamics of the actuators, rigid bodies, and controller action must all be combined to simulate the end-result dynamics of a robot. The balancing controller of this chapter was used in the simulation of dynamics in Chapter 3.

Besides the mechanical structure of the robot, the most important determinant of actuator requirements is the controller – different controllers can have dramatically different demands on the actuation system. Some types of controllers require powerful actuators for acceptable performance because they attempt to override the mechanical dynamics of the robot. Other controllers complement the mechanical dynamics of the robot and require far less actuator power. Though the most important quality of a controller is that it stabilizes the robot, commanded torques will vary from controller to controller, and these commanded torques will determine the necessary performance of the actuators.

For this reason, this chapter begins by arguing that trajectory tracking controllers are inappropriate for highly dynamic robots. Instead of independently tracking joint-level trajectories, highly dynamic robots must consider more dynamics in a more holistic manner because components exchange energy with such vigor that their controllers are no longer independent. As an alternative to trajectory tracking, a novel dynamic balancing controller for a monopod hopping robot is presented. The proposed controller uses analytic equations to specify ground reaction force profiles which in turn control the robot's centroidal momenta. Conceptually, the controller is similar to Raibert's three-part decomposed running controllers (Raibert, 1985), but without requiring the assumption that leg inertia and masses are negligible compared to the torso. Commanded torques from the controller require relatively low actuation bandwidths and are largely harmonic – thus they may be performed mechanically using properly tuned nonlinear springs at each joint. Actuators including such nonlinear springs were the subject of Chapter 4.

5.1 A Condemnation of Trajectory Tracking

The quantity of robotics research related to legged locomotion has exploded since Raibert's (Raibert, 1985) seminal research in running robots in the 1980's, yet successful examples of energetically efficient dynamically running robots remain relatively few. Perhaps the most efficient running robot is the ARL Monopod II, but it is a long way from the laboratory to the real world (Ahmadi & Buehler, 1999).

The failure to substantially improve upon the success of these early running robots may be a ironic consequence of the success of walking robots, which often successfully use single-trajectory tracking controllers with high gain joint-local feedback loops to great effect (Kaneko et al., 2008). While perhaps satisfactory for slow walking, slaving the motion of a highly dynamic system to a single trajectory is fundamentally the wrong control approach to follow. Although published mechanical details are few, even successful demonstrations of high-gain, joint-local feedback controlled "running robots" such as was produced by Toyota (Tajima et al., 2009) likely have both mechanical and control systems poorly suited to the execution of highly dynamic tasks, for the reasons discussed in section 2.2.

Considerable research has focused on trajectory generation and online high-gain position trajectory tracking. High-gain position trajectory tracking control is a well-understood and useful technique, but it is the wrong type of controller for efficient balancing legged robots for two reasons.

First, it is needlessly constraining to force the motion of a complex robot towards just a single trajectory when tasks such as running admit a plethora of acceptable trajectories. There are many trajectories that a legged robot could use to locomote which would all satisfy its most important task constraint: *to not fall down*. This task constraint is absolutely unrelated to the problem of following a trajectory – a robot might perfectly follow a joint-level trajectory and yet still fall down.

Second, if task-satisfying trajectories exist nearby the present trajectory but feedback control is used to track only a single one, the controller will constantly waste energy by applying forces in ways unrelated to actually satisfying the task. By analogy to nullspaces in redundantly actuated systems, we might say that very often feedback torques will do useless work the "task nullspace." If a robot just needs to end its motion with a particular state at a particular time, then any control effort not pushing the state of the robot directly towards this goal is in some sense wasting energy. In other words, feedback energy is often used in ways completely unrelated to satisfying the task at hand.

In the case where the trajectory is replanned each feedback cycle, these two objections might disappear, but such repeated, fast replanning techniques should probably no longer be considered a "trajectory following" controller.

Although the statement "avoid naive trajectory tracking for dynamic robots" may sound bold, there is already ample evidence of this approach in the literature. The design of controllers which avoid slaving motion to a single kinematic trajectory is evident even in the seminal work of Marc Raibert, who presented simple decomposed balancing controller that uses only three rules (Raibert, 1985). These three rules create a family of acceptable, stable trajectories based on the dynamics of the robot and its interaction with the environment – not just a single kinematic trajectory.

Trajectory-less controllers are frequently biologically inspired. The venerable SLIP model of running (Blickhan, 1989), or more recently ones based on analytic, sinusoidal GRFs (Robilliard & Wilson, 2005), use simple spring-mass models to describe the bouncing gait of running animals. More recently, springy walking models have also been developed (Geyer et al., 2006). As the centroidal angular momentum has been shown to play an important role in balance (Herr & Popovic, 2008), models such as the Reaction Mass Pendulum have recently been developed to take this into account (S. Lee & Goswami, 2007).

We will now look at the dynamics of animal hopping with the aim of developing the essential collection of

virtual components (J. Pratt et al., 2001) that stably couple a hopping robot to its environment.

5.2 Choosing Nominal Hopping Dynamics

5.2.1 Sources of Inspiration

What is hopping? The reader may have an immediate mental image a robot bouncing up and down at a regular rate, but the word “hop” is itself a little vague – we could have equivalently used the word “run”, since it has been shown that hopping is essentially just running on one virtual leg (Raibert, 1985).

It may be easiest to define hopping by what it is *not*; the only two essential constraints limiting what can be considered a “hopping” motion is that the robot *not* fall down and that on average it *not* move in the wrong horizontal direction. These are very loose constraints indeed, since they include gaits that are not stable or even well-directed when considered as single hops, yet these steps might concievably converge in a stable, directed behavior when considered over several footfalls.¹ Chaotic yet stable gaits have been observed in passive dynamic robots, showing that symmetry is not essential either(?, ?).

If the robot is allowed to fall occasionally and can get up by itself, we could expand our definition of locomotion to include robots that repeatedly fling themselves in the desired direction, fall, and get up again.

Faced with such a large canvas upon which to paint our desired dynamics, perhaps it would be wise to first reduce the size of the design space. We can do this by defining additional constraints, assumptions, or optimization criteria. Let us begin by looking at possible sources of dynamic inspiration.

Prerecorded: We might examine the output of a motion capture system that records the movement of a similarly sized human or animal subject.

Convenient: We might choose an ad-hoc class of motions that is easy to draw, program on a computer, or analyze mathematically.

Precomputed: We might select a nominal trajectory from the output of an optimization routine, planner, or heuristic algorithm such that it minimizes some quantity (say, energy) and maximizes another (say, robustness to disturbances).

Basis functions: We might use a collection of motor primitives to generate a motion, or a CPG to trigger various motor activations.

Artistic: We might pick motions sculpted by a researcher or by a character animator from an computer animation company², on the hypothesis that what humans find aesthetically pleasing is something that resembles desirable physical dynamics.

Constraints: We might use mathematical constraints to limit where we search for motions. An example would be only looking for motions in which the zero-moment point remains inside the support polygon.

Intuitive: From observations of animals, we might say that nominal running is symmetric about the sagittal plane, that the hip should stay above the knee and foot, and that the robot should not take steps that are too small or too big. These intuitive statements could then be converted into controller actions using virtual components.

¹To aid in visualization of particularly unusual gaits, the reader is encouraged to refer to (Chapman et al., 1971), in which skilled actors demonstrate asymmetric, aperiodic, irregular gaits with absurd foot trajectories that nevertheless result in stable legged locomotion.

²Pixar, for example.

Mechanical: Some purely mechanical systems have been found to walk, hop, and run stably using purely mechanical interaction with the environment. Starting from the dynamics of such structures may provide useful hints about stable running dynamics.

All of these methods and constraints have something to offer us, and in the next section we will borrow ideas from each of these sources to develop a balancing controller.

5.2.2 Essential Ideas in Hopping Dynamics

A tremendous amount of literature exists regarding legged locomotion of animals, humans, and robots. We will now mention just a few ideas relevant to the controller developed in this thesis.

Researchers in the field of animal locomotion have long ago presented the hypothesis that the dynamics of running animals is well modeled by springs and masses (Alexander, 1991; Blickhan, 1989). This model is often referred to as the SLIP (Spring-Loaded Inverted Pendulum) model, because it essentially is a point mass that interacts with the ground through a compression spring. Although the SLIP model cannot duplicate all salient features of animal running, as a first-order approximation of running dynamics it is acceptable and yields an important observation: hopping is an inherently springy activity. Not surprisingly, the first running robots developed by (Raibert, 1985) used simple decomposed control system and mechanical structures quite similar to the idealized SLIP model.

Furthermore, passive dynamic robots (McGeer, 1990) have shown us that achieving a major fraction of the dynamics via purely mechanical effects is simple, reliable, and energetically beneficial. If it is true that “running is like the bouncing of a ball” (Ringrose, 1993), this suggests that when a stance phase is viewed from a distance, the interaction between robot and ground should be essentially elastic in nature. Indeed, this is exactly what we see. Collisional studies suggest that pseudo-elastic collisions in walking is optimal (Ruina, Bertram, & Srinivasan, 2005). Animal studies further support this statement, showing that pseudo-elastic ground reaction forces are common in running animals, whether or not the forces arise from actual elasticity or via muscle work. A simple experiment showed that the effective stiffness of human legs during running varies with the stiffness of the terrain, while the stride frequency and contact time remain unchanged (Ferris, Louie, & Farley, 1998). Perhaps this suggests that humans strongly prefer a stereotyped ground reaction force pattern when running, and that we adapt our gait to maintain this pattern.

Running controllers also often emphasize the symmetry in running (Raibert, 1986). But is running really symmetric? Biomechanists such as Robilliard *et al.* have attempted attempt to explain the ground reaction forces of running animals using only sinusoids, which would correspond with a perfect symmetry (Robilliard & Wilson, 2005). While their paper is somewhat convincing for the vertical component of the ground reaction force, it is less compelling for the horizontal component. Robilliard *et al.* acknowledge this limitation, and present this interesting remark about kangaroos:

Hopping macropodids, whilst clearly bouncing, deviate from our [symmetry] assumptions as the stance phase is decidedly asymmetric: leg angle and leg length are much greater at foot-on than at foot-off, and the peak vertical GRF is significantly greater (36%) than that predicted using a sine wave.

Other research in legged locomotion focuses on central pattern generators (CPGs). Two crucial ideas relevant to CPGs are the phenomena of synchronization and entrainment. Synchronization between mechanical and control systems to create so-called “emergent behavior” is a common property of systems with CPGs, and shows a healthy connection between the two dynamic systems. The entrainment property of the controller is desirable

in that it promotes further adaptation to the current environmental-mechanical interaction. This synchronization effect will be achieved in the proposed controller via a parametric variable σ that resets at critical instants during hopping.

Finally, there is evidence that allowing whole-body angular momentum to stray from zero is useful in balancing tasks (Hofmann, Popovic, & Herr, 2009). This is supported by numerical studies of the optimal balancing strategies used in humans (Kuo, 1995), as well as recent examples in robot balancing (S. Lee & Goswami, 2007; J. Pratt & Krupp, 2008).

In conclusion, we will borrow these ideas from the literature:

- Running is inherently springy, and produces vertical GRFs that closely resemble a spring/mass system.
- A large fraction of running dynamics can be achieved through purely mechanical means.
- Symmetry is not required; the leg should be straighter when the foot lifts off the ground than when it touches down.
- A synchronizing entrainment effect between controller and environment is beneficial.
- The angular momentum of the mass centroid should be close to zero throughout a stride.

We will now develop these intuitive observations further via philosophical arguments the next section, then write them in a more mathematical way in sections 5.3 and 5.4.

5.2.3 Dynamic Balancing : The Art of Not Falling Down

What does it mean to “stabilize” a running or hopping robot? For some control problems we can define stability as a type of bounds upon the motion, as a state-space metric contraction, or as a decrease in system energy (Lyapunov function). Yet these types of stability seem inappropriate for legged robots. A robot’s energy might decrease continually as it falls over. Its state-space motion may converge to a single, perfectly tracked position trajectory, but the resulting forces on the environment might differ sufficiently from what was predicted that the robot still falls down.

The very word “stabilize” is probably inappropriate for discussing the property that we wish legged robots to express. The word “balance” is better, but we will need a little discussion about the posture of the robot, its dynamic behavior, and its response to environmental disturbances and uncertainty. So let us again pedantically consider “balancing” in terms of its opposite: falling down.

What are the fastest ways to fall down?

Without a doubt, under normal gravitational conditions, if the legs don’t push forcefully enough against the ground the robot will immediately fall. It is not the magnitude of the force – instantaneous forces might still be quite large – but rather the total vertical impulse given in an appropriately sized window of time. In general, the size of the window of time is determined by the operating conditions and parameters of the robot and may vary from step to step, depending on the extensible leg length or motion the robot’s CoM. If the robot’s legs don’t push sufficiently forcefully against the ground, the robot will fail to apply sufficient vertical impulse to the CoM during stance to counteract gravity. This is just a cumbersome definition of what it means when we say “fall in zero steps” or “fall like a bag of bricks.”

The next fastest way to fall down would be to take one step and bounce back up into the air again, but to make a mistake regulating the orientation of the robot such that it is impossible for it to take any further steps.

It is easy to imagine a robot taking one clumsy bouncing step, flipping upside down, and landing on its head. This is a kind of failure to regulate posture and angular momentum about the CoM such that the robot stays in a controllable operating range. This general statement is difficult to quantify, for the “controllable operating range” will vary from robot to robot. The regulation of “posture” involves both general concepts like keeping the torso above the feet, and specific ones like regulating the internal angular momenta about the center of mass. If at the end of the stance phase there is some non-zero centroidal angular momentum, the robot may continue to spin throughout the flight phase and make control of subsequent steps more difficult. Therefore, we should apply torques during stance to maintain a net angular momenta of approximately zero.

Finally, a third way to fall down is to create forces normal to the ground that exceed the frictional limit. It is easy to imagine what happens if the robot fails to keep ground reaction forces within the friction cone, especially if one has ever slipped on ice as a child. To satisfy the conditions of vertical impulse and orientation, we must still choose a center of pressure (CoP) that will not violate frictional constraints. The ability of an animal or robot to change its linear and angular momenta during a given step is determined only by the position of the CoP and the coefficient of friction between foot and ground. This is why the choice of CoP (*i.e.* foot location) is of critical importance, for it places limits on the direction and magnitudes of impulses that may be achieved during stance. Limitations on forces through the CoP are also why we must prioritize vertical control over horizontal control. It would be better for the robot to take a few extra steps and move horizontally in an undesired direction than to immediately fall down because the foot slips.

In summary, we have just three essential goals when balancing a legged robot: counteracting gravity, maintaining posture, and staying within the friction cone. To accomplish these goals, the choice of CoP is critical.

5.2.4 Constraints

For a robot to hop successfully, there are a variety of constraints that must be satisfied.

Time constraints: Once in the air, a robot moves ballistically. Its motion is largely uncontrolled, and it will contact the ground a short amount of time after takeoff. Extending or retracting the leg will affect the foot’s touchdown time, but only slightly.

Position constraints: The range of postures which allow the robot to push strongly against the ground is limited by maximum and minimum leg lengths, and joint range of motion. Not all postures allow sufficiently powerful leg extension motions.

Momentum constraints: The robot must be moving upward at the end of the stance phase with sufficient total momentum to lift off of the ground.

Actuator constraints: Real actuators have clear limits on the torque and angular velocities that they can achieve.

Friction constraints: There is only so much friction between the robot and ground, and if the foot slips the result in an uncontrolled robot.

To develop a controller that appropriately incorporates the constraints from all of these different domains is the goal of the next section. As we shall see, specifying a controller just the positional domain is be insufficient, as would be considering just the force domain.

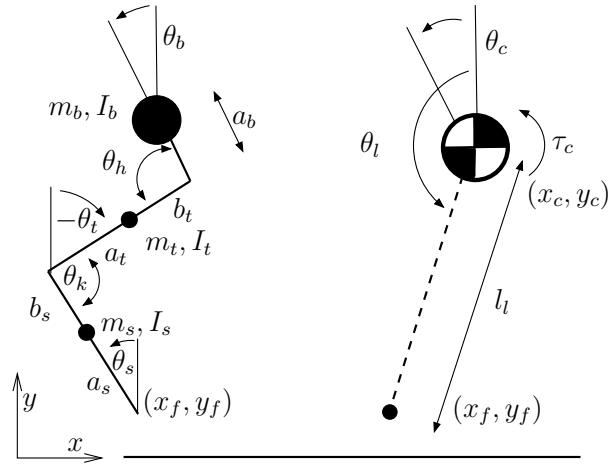


Figure 5.1: Left: The mechanical rigid body dynamics of JIMI. Right: The mass centroid task space, which possesses a “virtual leg” from JIMI’s CoM to its CoP.

5.3 Inverse Dynamics

Task space control is a popular technique for mapping torques in one coordinate system to another. In section 5.4 we will create task space controllers that operate on the mass centroid of the robot, as shown in Figure 5.1 right. In anticipation of this development, we must first create a mapping from the centroidal forces and coordinates to the system dynamics previously described in Chapter 3. This mapping will be accomplished using inverse dynamics to control ground reaction forces of the robot. The reader is encouraged to refer to section 3.4.1 as needed.

Although the mapping between CoM forces and GRFs could be created with virtual components that only use robot kinematics (J. Pratt, 2000), for a fast-moving robot a kinematic approach will not properly create the desired GRF terms. Instead, for a leg possessing mass and inertia, inverse dynamics must be used to more accurately implement the dynamical effects of the mechanical system. Let’s look at the dynamics for the stance and flight phases.

5.3.1 Stance Phase Inverse Dynamics

Rather than directly control joint torques τ_j , it would be easier if we could specify the vertical force through the centroid F_{cy} and torque about it τ_c . These two quantities are sufficient to control the centroidal momenta $P = (p_{cx}, p_{cy}, L_c)$ during stance, but because our underactuated model has only two interaction forces with the ground (F_{fx}, F_{fy}), only two of these three quantities may be regulated simultaneously. We therefore do not explicitly control p_{cx} during stance phase, as it may implicitly controlled at the step-to-step level via appropriate foot placement.

As we did with the non-slipping foot constraint, to compute the necessary joint torques that achieve some desired GRF during stance, we use Lagrange multipliers and add to equation (3.23) the constraint that real ground reaction forces be equal to the desired ones (*i.e.* that $\lambda_d - \lambda_r = 0$).

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_f^T & -\mathbf{D}_j^T \\ \mathbf{J}_f & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\lambda_r \\ \tau_{jd} \end{bmatrix} = \begin{bmatrix} \tau_v - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} \\ -\mathbf{j}_f\dot{\mathbf{q}} \\ -\lambda_d \end{bmatrix} \quad (5.1)$$

Solving this system of equations gives the desired joint torques τ_{jd} which create the desired ground reaction force λ_d , as well as the expected accelerations on the links $\ddot{\mathbf{q}}$ that will result.

It will be convenient to specify the desired torque about the center of mass τ_c , and treat F_{fx} as a function of F_{fy}, τ_c . From geometry we may write

$$F_{fx}(F_{fy}, \tau_c) = \frac{(x_c - x_f)F_{fy} - \tau_c}{(y_c - y_f)} \quad (5.2)$$

To summarize, given τ_{cd} and F_{fyd} we use (5.2) to find F_{fxd} . Then $\lambda_d = [0 \ 0 \ 0 \ F_{fxd} \ F_{fyd}]^T$ is inserted into (5.1), and solved to give joint torques τ_{jd} .

5.3.2 Flight Phase Inverse Dynamics

Since the robot cannot control any of the centroidal momenta during flight, we must resign ourselves to position control of the foot relative to the centroid. We will again use inverse dynamics to compute the joint torques which accelerate the foot in the desired way. If $\ddot{\mathbf{q}}_f = [\ddot{x}_f \ \ddot{y}_f]^T$ is the desired foot acceleration, we can add a constraint foot position $\mathbf{B}\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{fd} = 0$ to the equations of motion and solve simultaneously:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{D}^T \\ \mathbf{J}_f & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \tau_j \end{bmatrix} = \begin{bmatrix} \tau_v - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} \\ \ddot{\mathbf{q}}_{fd} \end{bmatrix} \quad (5.3)$$

Solving (5.3) gives us the joint torques τ_j needed to create a desired foot acceleration $\ddot{\mathbf{q}}_{fd}$, as well as the accelerations on the links $\ddot{\mathbf{q}}$ that will result.

5.3.3 Virtual Leg Task Space

During flight, it will be convenient to specify the swing of a virtual leg relative to the CoM than to use accelerations in world coordinates. We therefore define a virtual leg from the centroid to the point foot whose state is defined polar coordinates (θ_l, l) , as shown in Figure 5.1 right. The motion of this virtual leg motion may be computed from basic geometry

$$\theta_l = \tan^{-1} \frac{x_c - x_f}{y_c - y_f} \quad (5.4)$$

$$l_l = \sqrt{(x_c - x_f)^2 + (y_c - y_f)^2} \quad (5.5)$$

Derivatives of equations (5.4) and (5.5) are not shown here, but they and their inverses are easily derived and will be used below.

5.3.4 Centroidal Momenta

Because of the choice of generalized coordinates it is not possible to express the momenta of the CoM as a linear function of \mathbf{q} . However, linear momenta p_{cx} and p_{cy} may be computed in cartesian world coordinates from the motion of link centroids. If the x and y coordinates of the i th mass is (x_i, y_i) and $i = \{b, t, s\}$, then $p_{cx} = \sum_i m_i \dot{x}_i$ and $p_{cy} = \sum_i m_i \dot{y}_i$. Similarly, the angular momentum about the centroid is

$$L_c = \sum_i I_i \dot{\theta}_i + \sum_i m_i((d_c - d_i) \times \dot{d}_i) \quad (5.6)$$

where $d = [x \ y]^T$ are coordinate vectors. Finally, the position of the centroid x_c, y_c is found using $x_c = \frac{p_{cx}}{M}$ and $y_c = \frac{p_{cy}}{M}$, where $M = \sum m_i$ is the total system mass.

5.4 Centroidal Momentum Balancing Controller

5.4.1 Controller Overview

Marc Raibert's classical three-part decomposed controller for running robots incorporated one controller for regulating vertical hop height, one for leg swing, and one for torso angle (Raibert, 1985). A similar approach is apparent in the work of Jerry Pratt (J. Pratt & Krupp, 2008). The hopping controller for JIMI is likewise decomposed into three largely independent controllers:

1. *Vertical GRF controller*, which determines the vertical GRF and therefore hopping height. During stance, the centroidal vertical momentum p_{cy} is controlled via a vertical GRF of the form $F_{fy}(t) = F_{peak} \sin(\omega t)$. This sinusoidal profile is scaled by setting F_{peak} such that the robot has enough momentum to leap into the air at the end of the stance phase.
2. *Centroidal angular momentum controller*, which regulates the internal angular momentum via the horizontal GRF and tries to maintain body posture. A torque about the Center of Mass(CoM) is applied during stance such that the total change in centroidal angular momentum over a stride integrates to zero.
3. *Foot placement controller*, which determines the CoP range and is critical for dynamic balancing. During flight, the leg needs to be smoothly lifted, swung forward, and then lowered to the desired touchdown point.

We will now justify this choice of decomposition based on physically intuitive observations.

Ground interaction force vectors from the CoP towards the CoM provide the best way to change the direction of the robot because they do not change the internal angular momentum of the robot. For a robot with a point foot such as in our model, foot placement uniquely determines the CoP.³ Therefore foot placement and determining the CoP is arguably the most critical thing for regulating the motion of the robot and not falling over.

The next decision the controller must make is: *what patterns of force that act through this connection between robot and environment?* As established in section 5.2.2, running models such as SLIP well characterize animal running using a force vector from the CoP through the CoM that resembles a conservative, elastic motion. Choosing a sinusoid satisfies this constraint and matches the analytic hopping models developed in section 3.3.

Finally, we come to the regulation of internal angular momentum. As mentioned earlier, there is evidence that animal and humans prefer to maintain approximately zero internal angular momentum during locomotion. We follow the hypothesis that internal angular momentum is best conserved during normal operation and predominantly used to stabilize the robot in response to disturbances (J. Pratt & Krupp, 2008). Angular momentum controllers can also stabilize the rate of change of angular momentum rather than the momentum or angle itself, as other researchers have studied (Goswami & Kallem, 2004).

³For robot with bigger feet, the CoP a range of points that could be the CoP. We would then have to select the position of the CoP dynamically throughout a step using ankle torques, but such a case is outside the scope of this thesis.

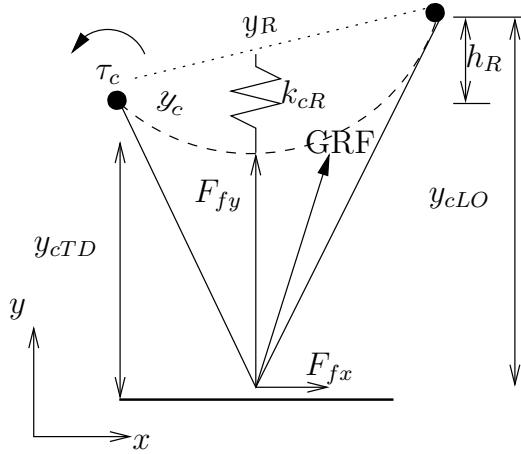


Figure 5.2: Idealized motion of JIMI's CoM during the stance phase, with relevant quantities labeled.

For the moment, we will completely ignore the regulation of the CoM's horizontal momentum. Compared to the regulation of the CoM's vertical momentum or angular momentum of the CoM, control of the horizontal motion of the robot is simply less important. Poor regulation of horizontal motion does not imply that will fall, just that it might not be going in the right direction. For most environments there is plenty of time to correct this problem. A robot that jumps in place stably is 90% of the way to a completely functioning running robot.

In the following, superscript *LO* will be used to indicate foot lift-off, *TD* touch-down, *S* stance, and *F* flight.

5.4.2 Vertical GRF Controller

Ground reaction force measurements of running animals show that the force in the vertical direction is essentially sinusoidal (Alexander, 1988). For a controller, rather than select a single sinusoidal force trajectory, our goal is to create a family of sinusoidal forces that stabilize the vertical hopping motion. Referring to Figure 5.2, this goal can be accomplished using the rule:

$$y_R(t) = \hat{y}_{cTD} + \frac{h_R t}{T_S} \quad (5.7)$$

$$F_{fy}(t) = K_{cR}(y_R - \hat{y}_c) \quad (5.8)$$

where $y_R(t)$ is the height of a rising “virtual rocket”, T_S the expected duration of stance, h_R is the height that the rocket rises every T_S seconds, t is the time elapsed so far during stance, and K_{cR} is the stiffness of the spring connecting the CoM and the virtual rocket.

In practice, the rising motion of the rocket adds energy to the system, and an equilibrium is quickly reached between the energy added to the system by the virtual rocket and the energy lost to collision and friction.

This vertical stabilization rule is similar to one presented in Robilliard's work (Robilliard & Wilson, 2005). It is also produces asymmetric leg behavior typical of animal running; the leg is more stiff and bent during landing than at takeoff (Cavagna, 2006).

5.4.3 Centroidal Angular Momentum Controller

It seems reasonable to desire that the robot's posture remain generally upright, although some small rotations of the body during stance seems acceptable, especially when moving forward. The largest disturbance to the

centroidal angular momentum occurs at the foot-ground impact.

Regulation of whole-body posture, centroidal angular momentum, and torque about the centroid are all closely related. If we define a quantity called θ_c to be the integral of the angular momentum, then we can write $\tau_c = \frac{d}{dt} L_c = \frac{d^2}{dt^2} \theta_c$. Note that if the body of the robot has a very large moment of inertia relative to the leg, it would be sufficient to regulate only θ_b (as the original Raibert controller does) to stabilize the angular momentum of the robot because body dynamics would dominate. Thus, it is tempting to use a simple PD controller in the style of Raibert's original design

$$\tau_c = K_{cp} \theta_c + K_{cD} L_c \quad (5.9)$$

This works well, with some caveats. Immediately after touchdown it creates a large torque that often destabilizes the robot or violates the friction cone. One way to avoid these problems is simply to limit the desired normal force F_{fxd} to safe levels:

$$F_{fx} = \min(K_f F_{fy}, F_{fxd}) \quad (5.10)$$

where K_f is a conservative estimate of the friction coefficient. This essentially gain-schedules F_{fx} .

We should also consider that maintaining perfectly zero angular momentum may not really be our control goal. Since the leg will have some inertia as well, it is acceptable to allow the body to rock forward and back during stance as the leg sweeps out a stride. This rocking motion doesn't interfere with our goal of balanced locomotion and seems perfectly natural from observing animals running.

An easy way to approximate θ_c and allow a slight rocking is to use the difference of the body posture and the virtual leg so that

$$\theta_c = K_{lb} \theta_b - \theta_l + \theta_{bd} \quad (5.11)$$

where θ_{bd} is a slight offset to keep the body leaning slightly forward or backward, and K_{lb} is a constant chosen experimentally to roughly express the ratio of average centroidal inertia to body-link inertia (i.e. $K_{lb} \approx \frac{I_c}{I_b}$).

5.4.4 Centroid Ballistic Motion in Flight

As we have mentioned, during the flight phase the motion of the centroid is essentially uncontrolled. Although the posture angle may be adjusted slightly mid-flight, the system's momenta will be conserved throughout the flight, and neglecting air resistance, the motion of the CoM during flight may be closely approximated with a parabolic arc. We need to find the motion of the exact CoM during flight so that we may estimate the expected touchdown time of the robot and plan a smooth trajectory to that point.

The desired virtual leg position at touchdown $l_d^{TD}, \theta_{ld}^{TD}$ can be used to compute the desired CoM height at touchdown:

$$y_{cd}^{TD} = y_f^{TD} + l_d^{TD} \cos \theta_{ld}^{TD} \quad (5.12)$$

The trajectory of the CoM is given by

$$\frac{1}{2} \ddot{y}_c t^2 + \dot{y}_c t + (y_c - y_{cd}^{TD}) = 0 \quad (5.13)$$

and we can find the estimated flight time remaining $T^F - t$ using the quadratic equation:

$$T^F - t = \frac{-\dot{y}_c \pm \sqrt{(\dot{y}_c)^2 - 4(-\frac{1}{2}g)(y_c - y_{cd}^{TD})}}{2(-\frac{1}{2}g)} \quad (5.14)$$

Although this equation provides two solutions, we only consider the positive real solution as it is one with the desired physical meaning.

Finally, let's compute a quantity $0 \leq \sigma \leq \pi$ that represents the fraction of the flight phase that has completed, so that we may create a synchronized leg swing during flight.

$$\sigma = \frac{\pi(t - t^{LO})}{T^F} + \frac{\pi}{2} \quad (5.15)$$

5.4.5 Leg Swing

During flight, we desire to swing the leg forward such that the position of the foot at the moment of touchdown is exactly as desired. As Raibert *et al.* originally did, might define the torque on the leg to be

$$\tau_l = K(\theta_{ld}^{TD} - \theta_l) \quad (5.16)$$

but this results in a huge kicking torque right after liftoff and neglects the problem of retracting the foot while in flight for better clearance. Instead, let's choose to swing the virtual leg using sinusoidal functions expressed in polar coordinates. The desired trajectory becomes

$$\theta_l(\sigma) = K_1 \sin(\sigma + \frac{\pi}{2}) + K_2 \quad (5.17)$$

$$l_l(\sigma) = K_3(\sin(\sigma) + \sin(2\sigma)) + K_4 \quad (5.18)$$

where K_1, \dots, K_4 are various constants that need to be chosen such that the boundary conditions at the start and end of the swing are met – the robot ends the motion in the desired place at the desired time.

$$K_1 = \frac{1}{2} (\theta_{lTDd} - \hat{\theta}_{lLO}) \quad (5.19)$$

$$K_2 = K_1 + \hat{\theta}_{lLO} \quad (5.20)$$

$$K_3 = \frac{1}{2} (l_{lTDd} - \hat{l}_{lLO}) \quad (5.21)$$

$$K_4 = K_3 + \hat{l}_{lLO} \quad (5.22)$$

This results in a smooth forward leg swing with a retracted leg because of the addition of the 2nd harmonic in equation (5.18). Because of the small forces during flight, the desired motion of the foot may be tracked reasonably well using the torques computed by the inverse dynamics, and only small feedback gains are typically required.

Table 5.1: Selected Control Parameters

	Description	Value	Unit
T_{Sd}	Desired stance time	0.2d0	-
T_{Fd}	Desired flight time	0.25d0	-
K_{cP}	Proportional gain of centroidal angle	50.0	-
K_{cD}	Damping gain of centroidal momentum	25.0	-
θ_{bLOd}	Desired body/centroid Angle at liftoff	0.35165	rad
K_R	Y-axis virtual spring stiffness	3800	N/m
h_R	Rocket height rise per step	0.1	m
K_{lP}	Leg length proportional gain	300	-
K_{lD}	Leg length derivative gain	30	-
$K_{\theta lP}$	Leg angle proportional gain	300	-
$K_{\theta lD}$	Leg angle derivative gain	30	-

5.5 Conclusion

In this chapter, we developed a very simple balancing controller for a monopod. An implementation of the inverse dynamics of the robot was used to select joint torques. In simulation the technique works well, although it has not yet been fully tested on the physical prototype of JIMI.

The work in this paper could also be rewritten equivalently under the operational space control formulation (Khatib, 1987). However, we must be careful to not create too perfect a task space control mapping, lest we lose sight of the real goal – it is more important to exploit the passive mechanical dynamics of the robot than to have perfectly sinusoidal forces on the ground. In this respect, VMC based on kinematics may prove more robust in the face of modeling uncertainty (J. Pratt, 2000).

For the model presented in this paper, control of the angular momentum cannot be done independently from the control of the posture of the torso. This may suggest why a robot with 3 links in the leg, (not just the two links + torso in this model) would be better able to regulate its posture during running; it would better decouple the angular momentum of the torso from that of the leg. It would be interesting to explore if this yielded improved stability or energy efficiency, especially on rough ground.

Another failing of the controller in this chapter is that the posture of the robot must always produce a well-conditioned matrix; being too near a singularity results in this method producing unrealistically large joint torques. Improving this simple controller to require fewer experimentally-obtained control constants is another subject of current research.

Other, future topics that should be addressed include proper regulation of the horizontal momentum. This subject was intentionally not considered in this paper because it must be left as a free parameter to accomplish tasks: horizontal motion should be planned to move in a particular direction, jump over obstacles, and interact with any foothold planning algorithms that exist at a higher level.

As an aside, it would be interesting to speculate on where animals draw the line between dynamics achieved mechanically or via neural control, and the possible reasons for the resulting balance of mechanics and control. What quantity are we minimizing when we are being lazy? Muscle work, neural control effort, or peak forces? It may be the case that many animals partition more of the dynamical task into the mechanical domain, because their ability to do complex control requires complex neural networks that are not as simply efficient as mechanical or stimulus-response neural solutions.

CHAPTER 6

Model-Based Estimation

It has been known for decades that information theory and control theory are closely related problems;¹ the more information we have about a system, the better we may control it. With an accurate system model, almost every control task becomes easier: we can understand which PID gains will be the most performant or robust; we can manipulate its dynamics into an arbitrary or more desirable form, such as in Chapter 3 when inverse dynamics were used to precisely control ground reaction forces; we can reduce feedback gains by using feedforward terms, requiring less work from our actuators; we can plan efficient, realizable trajectories in the future which do not attempt to contradict the natural dynamics of the system; for finite horizon control problems, we might even attempt to distribute our feedback effort over time by continuously replanning new trajectories that minimize energy consumption or actuator bandwidth use. All of these control problems depend upon accurate information about the system, but how do we discover that information?

This chapter is concerned with estimation problems, essentially the inference task of extracting important information from imperfect data. Although the estimation of values observed in the past, present, or future are all different sides of the same problem, we unfortunately call them by three different names: *smoothing*, *filtering*, or *prediction*. If we are attempting to estimate the quantities of a model, this closely-related inference problem is known as *System Identification* and is an epistemic problem familiar to most engineers: how do we know that the values of the components *actually are* what we think they are? We will refer collectively to all these inference problems as *estimation* and treat them under the same Bayesian, model-based framework.

Naturally, there exist many techniques used by engineers and scientists to solve estimation problems, but often the methods are ad-hoc or excessively simplified for computational simplicity. Approximations and assumptions are quite acceptable and useful especially for extracting the essential nature of a complex system, but the limits to the validity of the approximation must be remembered at all times. Discovering model parameters accurate enough for predictive control demands that we start from a firm theoretical foundation.

The approach followed in this chapter is to always consider estimation and inference problems *with reference to the system model*. For example, it is common engineering practice to filter out high frequency content as *noise*

¹ Kalman noted that model-based linear filtering is the dual problem of the noise-free regulator problem(Kalman, 1960); stochastic optimal control theory is so closely related to information theory as to be nearly inseparable.

and treat only the low frequency content as *signal*. However, in this chapter we do not seek to categorically discard high frequency information on the principle that high frequency (HF) information is noise – although HF signals are often noise, they are not necessarily so. Ignoring HF signals may lead to slower detection of impulsive collisions, faults, inductor voltage spikes, or mechanical ringing. Instead, we should look for states that are of *high probability* in light of our system model. Under such a framework, HF information can still be regarded as unlikely (as noise) when it represents an unrealistically large or fast energy flow in and out of the system. But if a HF signal has arisen from a physical process and is consistent with the entire model, the system will still have the potential to detect it.

Unfortunately, inference problems are difficult for a number of reasons:

1. Often we have neither an accurate model of the system nor accurate measurements of the system's state. Should we try to estimate the model before reconstructing the states, or estimate the states before reconstructing the model? Or both at the same time?
2. State measurements are usually noisy, often with different noise distributions for each quantity in the system.
3. Often many system states are not directly observable.
4. Some states may not even be estimable given the limited sensor inputs.
5. The system itself may transition from state to state in a nonlinear way.
6. Model parameters may change with time as the machine heats, cools, wears down, or interacts with the environment, necessitating their re-estimation.
7. It can be hard to quantify how (un-)certain we are about the estimated state and model parameters.

Before introducing the estimation system in this chapter, let us first simply dream about what an ideal solution to the above difficulties would look like. We might desire to create an observer with the following characteristics:

1. States should be estimated with reference to our white or gray box² model and not from any ad-hoc smoothing techniques.
2. Noisy measurements should affect state estimates with less weight than cleaner measurements.
3. Inter-domain comparisons should use the same units, such as energy, so that different unit scales do not affect the result.
4. The inference system should operate in real time and continuously improve itself.
5. It should use the history of states to estimate model parameters, weighting recent history more than past history to allow for very slow parameter drift.
6. In light of improved parameter estimates and acausal state information, allow reinterpretation of recent state history.
7. Estimations made by the observer should be consistent and satisfy the conservation of energy.

²Models with an unknown internal structure are often called “black box” models and their behavior is estimated with general, non-parametric learning models; if the structure is completely or partially known, we may refer to it as a “white” or “gray” box model.

8. The estimation process must be stable and not perform regression or incremental learning on so little data that numerical rounding errors affect convergence.

We will see that the state observer and model parameter estimator developed in this chapter can accomplish most of these goals.

The remainder of this chapter is organized as follows: a brief review of Bayes' Theorem and inference techniques is presented in section ??, to review why linear systems with Gaussian noise and Gaussian prior distributions (sometimes called LGG systems) are so fundamental to machine learning and inference. The section may be safely skipped by those familiar with such material. Next, a derivation of the Hamiltonian equations of motion for the HypoSEA is given in section 6.2 – readers unfamiliar with Port Hamiltonian systems may wish to consult Appendix ???. Finally, section ?? presents the results of the estimation of state and model parameters in detail, and section ?? provides some conclusions.

6.1 A Brief Review of Inference Techniques

Bayes' Theorem provides the brute-force theoretical machinery into which we feed inference problems and Gaussian distributions yield simple mathematics, so we will briefly review both topics in the context of least squares (LS), maximum likelihood (ML), and maximum *a posteriori* (MAP) techniques. This is not intended to be a complete treatment of the subject of Bayesian inference; rather it is merely a review of the essential facts that were used during the HypoSEA parameter estimation process.

6.1.1 Bayes' Theorem

Bayes' Theorem follows directly from the definition of conditional probability and is of great philosophical value when considering inference problems. It is usually written like this:

$$P(\varphi|D) = \frac{P(D|\varphi)P(\varphi)}{P(D)} \quad (6.1)$$

where φ are a set of parameters that uniquely define a model in the hypothesis space, and D is the data that you have seen so far. Some authors(MacKay, 2003) prefer to explicitly remind you that you are using a particular hypotheses space and include the \mathcal{H} everywhere:

$$P(\varphi|D, \mathcal{H}) = \frac{P(D|\varphi, \mathcal{H})P(\varphi|\mathcal{H})}{P(D|\mathcal{H})} \quad (6.2)$$

However, such notation is only really useful when one plans to compare different hypothesis spaces at some point. Since such a sophisticated approach is not needed in this work, we will leave implicit the redundant \mathcal{H} .

The four terms of Bayes' Theorems are often referred to with convenient names:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \quad (6.3)$$

Having an intuitive understanding of each term and the meaning of each is fundamental. Although they must be manipulated as probability density functions (PDFs) analytically, they can also be understood simply in terms of probabilities. The author thinks of them as follows:

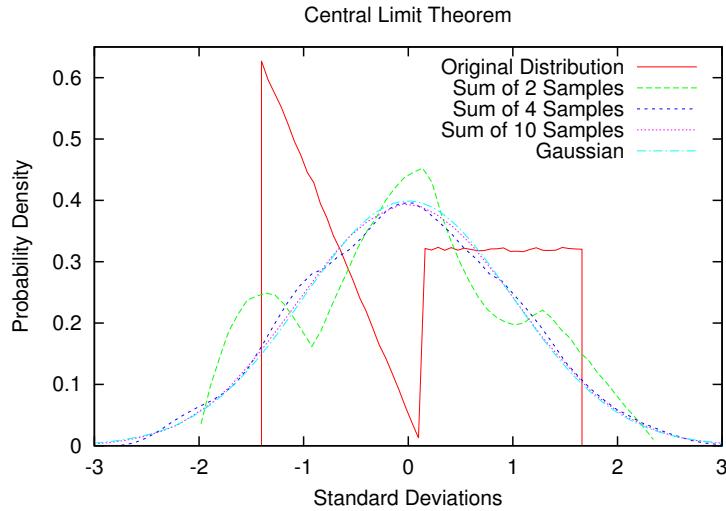


Figure 6.1: If just one sample at a time is taken from a bimodal, jagged probability density function, the original probability distribution be seen from the sample histogram. Yet if a histogram is made from *multiple* samples summed together, the new distribution will increasingly resemble a Gaussian according to the central limit theorem.

likelihood The probability you would see this data, assuming that the model with parameters φ is true.

prior The probability you think that your model parameters are true, before seeing any data.

evidence The probability that you would see this data, as predicted by the weighted average of all your models.

posterior The probability of your model parameters being true, after having seen the data.

We will use these terms extensively in later sections.

6.1.2 Gaussian Distributions

Normal distributions, or *Gaussians* as they will be called henceforth, are the reigning queen of all probability distributions due to the Central Limit Theorem(MacKay, 2003). Stated verbally, it says that if you add enough random samples from any probability distribution together, the resulting distribution will approach a Gaussian in the limit (Figure 6.1).

This is a profound and surprising result, considering the PDF of a Gaussian is

$$P(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (6.4)$$

It certainly isn't obvious to the author's intuition that such an expression would arise naturally nearly everywhere probabilities appear, but is nevertheless a mathematical truth of enormous significance. In many ways, we may go so far as to say that linearity is to vector spaces what Gaussians are to probability density functions. Both linearity and Gaussians are fundamental and can provide useful approximations to more complex nonlinear functions or probability distributions.

Gaussians have so many useful properties that we should hesitate to list some for fear of ignoring others. Yet we must mention those characteristics most relevant to this thesis:

- Gaussians maintain their shape through linear transformations (*i.e.*the conjugate prior of a Gaussian is also a Gaussian), making the post-transformation statistics easy to predict.

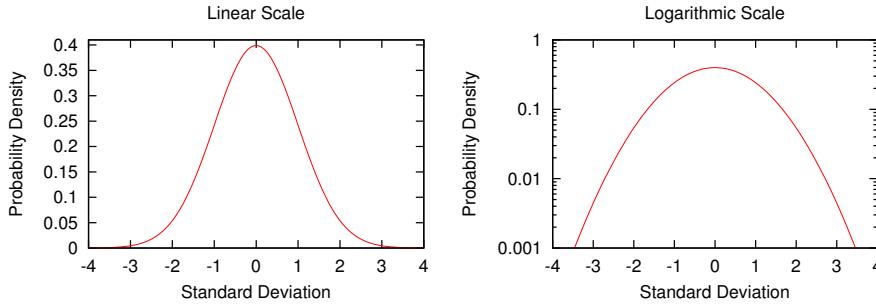


Figure 6.2: The familiar “bell curve” of the Gaussian distributions becomes a perfect quadratic curve on a logarithmic scale.

- Every Gaussian can be expressed as an exponential quadratic e^{ax^2+bx+c} . The simplest example of which is $P(x|\mu=0, \sigma^2 = \frac{1}{2\pi}) = e^{-\pi x^2}$. This fact is fundamental to their equivalence with least squares because it means Gaussians are logarithmically concave.

Let's now look at how easy it is to infer what kind of Gaussian some samples came from.

6.1.3 Maximum Likelihood Gaussian Inference

Given a single data point x and the hypothesis \mathcal{H} is that it was taken from a Gaussian with parameters $\{\mu, \sigma^2\}$, the probability that it came from that Gaussian is described by equation (6.4). If we have not just one sample x but a collection of independent samples $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$, then the likelihood of seeing that data given the model w would be the product of the probabilities of each individual data point:

$$P(\mathbf{x}|\mu, \sigma^2) = \prod_{i=1}^N P(x_i|\mu, \sigma^2) \quad (6.5)$$

Perhaps you can see how as we take more samples N the likelihood is going to get very small indeed. In fact, they will become so small that it will be easier if we switch from using likelihoods to the logarithm of the likelihood, so that the numbers can be represented with fewer digits. Another benefit of log-likelihoods is that the algebra will be simpler because likelihoods multiply but log likelihoods simply add. Furthermore, when graphed on a logarithmic scale the gaussian's log-likelihood becomes a simple parabola and can be expressed exactly with a second order equation (Figure 6.2).

The log likelihood of our observed data is thus

$$\ln P(\mathbf{x}|\mu, \sigma^2) = \sum_{i=1}^N \left[\ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{(x_i - \mu)^2}{2\sigma^2} \right] \quad (6.6)$$

$$= \ln \left(\frac{1}{2\pi\sigma^2} \right)^{\frac{N}{2}} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \quad (6.7)$$

If we define the sample mean \bar{x} and sample variance S to be

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (6.8)$$

$$S = \sum_{i=1}^N (x_i - \bar{x})^2 \quad (6.9)$$

then we may write equation (6.6) even more simply:

$$\ln P(\mathbf{x}|\mu, \sigma^2) = -\frac{N}{2} \ln(2\pi\sigma^2) - \frac{N(\bar{x} - \mu)^2 + S}{2\sigma^2} \quad (6.10)$$

This is a very useful equation, because now we can directly compute the log-likelihood of the data using only two statistics, \bar{x} and S , which are sufficient to uniquely describe a Gaussian. The forward problem is solved; given an hypothesis about $\{\mu, \sigma^2\}$ and some samples \mathbf{x} , we quickly calculate the sample statistics \bar{x} and S , and plug them into equation (6.10) to compute how likely it would be for us to see that data given that hypothesis.

Now we must turn to the inference problem: how can we find the *best* model parameters given some data? Intuitively, since we know nothing else about where the data came from, we want to find the μ, σ^2 that yield the greatest likelihood of the data, and this means maximization or minimization – which means looking for critical points where the partial derivatives of the likelihood are zero. Thankfully, since the logarithm is a strictly increasing continuous function over the range of the likelihood, the values which maximize the likelihood will also maximize its logarithm, and we do not need to worry about our earlier choice of using log likelihoods.

We therefore take the partials of equation (6.10) with respect to μ and σ .

$$\frac{\partial}{\partial \mu} \ln P(\mathbf{x}|\mu, \sigma^2) = -\frac{N(\bar{x} - \mu)}{\sigma^2} \quad (6.11)$$

$$\frac{\partial}{\partial \sigma} \ln P(\mathbf{x}|\mu, \sigma^2) = -\frac{N}{\sigma} + \frac{N(\bar{x} - \mu)^2 + S}{\sigma^3} \quad (6.12)$$

It is trivial to see that $\frac{\partial P}{\partial \mu}$ only at $\bar{x} = \mu$. We also see that $\frac{\partial P}{\partial \sigma^2} = 0$ only at $\sigma^2 = \frac{N(\bar{x} - \mu)^2 + S}{N}$. Assuming that $\mu = \bar{x}$ this would give $\sigma^2 = \frac{S}{N}$. Although this is a biased estimator (since in general, μ will not equal \bar{x}) in the limit of large N it will estimate the true value with less mean least square error than the “Bessel-corrected” unbiased $\frac{S}{N-1}$ estimator. Either way, the issue is somewhat academic from the Bayesian viewpoint – we aren’t looking for the best estimator of an unknowable quantity, just the one that most likely produced the data we have seen.

More interesting is to ask how we might quantify our certainty about our estimates $\hat{\mu}, \hat{\sigma}^2$? One way is to treat $\hat{\mu}, \hat{\sigma}^2$ as random variables and find their own standard deviations $\sigma_{\hat{\mu}}, \sigma_{\hat{\sigma}}$.

TODO: Add more to this section.

The bounds on our certainty of $\hat{\mu}$ and $\hat{\sigma}$ are therefore

$$\sigma_{\hat{\mu}}^2 = \frac{\hat{\sigma}^2}{N} \quad (6.13)$$

$$\sigma_{\hat{\sigma}}^2 = \frac{1}{2N} \quad (6.14)$$

In summary, the likelihood of seeing data \mathbf{x} is maximized by assuming it came from a Gaussian distribution of mean $\hat{\mu}_{ML} = \bar{\mathbf{x}}$ and $\hat{\sigma}_{ML}^2 = \frac{s}{N}$. These facts will be used in section 6.1.6.

6.1.4 System Model for Estimation Tasks

In later sections we will consider several estimation techniques in order of increasing complexity. To make direct comparison between techniques easier, we will consider the same system in all cases. Specifically, we will use a linear Port-Hamiltonian System (PHS) model in state-space representation:

$$\dot{\mathbf{x}}_i = \mathbf{A}(\varphi)\mathbf{x}_i + \mathbf{B}\mathbf{u}_i + \epsilon \quad (6.15)$$

$$\mathbf{y}_i = \mathbf{C}(\varphi)\mathbf{x}_i + \varepsilon \quad (6.16)$$

$$\mathbf{z}_i = \mathbf{B}^T\mathbf{x}_i \quad (6.17)$$

where the subscript \cdot_i indicates the value at time t_i , $\mathbf{A}_i(\varphi)$ is the system transition matrix (whose elements α are functions of model parameters φ), \mathbf{B} is the external port connectivity matrix, \mathbf{y}_i is the vector of observations of system state, \mathbf{u}_i is the external port effort vector, \mathbf{z}_i is the external port flow vector, ϵ is process noise and unmodeled dynamics, and ε is measurement noise.

It will be convenient when doing system estimation to group many samples of the above time-varying terms together into large matrices or vectors for the purposes of least squares estimation. When we do so, we will underline the vector or matrix to indicate that it is not just at a single instant in time, but rather a collection of matrices or vectors.

$$\underline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}, \underline{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}, \underline{\epsilon} = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_N \end{bmatrix}, \underline{\varepsilon} = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_N \end{bmatrix} \quad (6.18)$$

We may now mathematically specify the two inference problems that we face:

System Identification Given $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$, what is the best estimate of the φ which generates $\mathbf{A}(\varphi)$?

State Estimation Given observations $\underline{\mathbf{y}}$ and system parameters φ , what is the best estimate of each \mathbf{x}_i ?

These problems possess many similarities, such as both being complicated by system and measurement noise (ϵ, ε). In fact, the two problems are essentially two sides of the same coin; system identification is the inverse viewpoint, whereas state estimation is the forward viewpoint. We will consider both cases in the following techniques.

6.1.5 Least Squares (LS)

If we had no prior information or assumptions to make about an inference problem, we would probably reach first for the technique known as *least squares*. Typically we define the residual \mathbf{r} to be the difference between the predicted and measured values. Although we cannot know the true values of the system to compute the error – defined as the difference between the true and predicted values – we hope that by minimizing the residuals we reduce our error as well. For the system of equation (6.15), we could define the residual as the difference between the predicted model dynamics and the observed measurements using

$$\underline{\mathbf{r}}_i = \begin{bmatrix} \epsilon \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}}_i - \mathbf{A}(\varphi)\mathbf{x}_i - \mathbf{B}\underline{\mathbf{u}}_i \\ \mathbf{y} - \mathbf{C}(\varphi)\mathbf{x}_i \end{bmatrix} \quad (6.19)$$

A *loss function* is some function of the residuals $\underline{\mathbf{r}}$ that returns a scalar quantity indicating how “big” the residual vector is. The quadratic loss function is defined as

$$L_2(\underline{\mathbf{r}}) = \|\underline{\mathbf{r}}\|^2 = \underline{\mathbf{r}}^T \underline{\mathbf{r}} \quad (6.20)$$

The least squares method says that if we are doing system identification, our best estimates of φ should minimize the above quadratic loss function. Similarly, if doing state estimation, we should pick the \mathbf{x}_i which minimizes it too.

$$\hat{\varphi}_{LS} = \underset{\varphi}{\operatorname{argmin}} \|\underline{\mathbf{r}}\|^2 \quad (6.21)$$

$$\hat{\mathbf{x}}_{LS} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\underline{\mathbf{r}}\|^2 \quad (6.22)$$

The analytic expressions by which we may pick these least squares best estimate are well known. Geometrically speaking, the best least squares estimate is the projection of n equations onto the m dimensional subspace of solutions. That is, we assume that we have fewer parameters $\beta = [\beta_1 \ \beta_2 \ \dots \ \beta_m]^T$ than linear equations $\Gamma_i \beta = \gamma_i$ for $i = 1, \dots, n$, so that our system equations can be written as

$$\underline{\Gamma} \beta = \underline{\gamma} \quad (6.23)$$

We may then minimize the size of the residuals (as measured by the quadratic loss function) using

$$\hat{\beta}_{LS} = (\underline{\Gamma}^T \underline{\Gamma})^{-1} \underline{\Gamma}^T \underline{\gamma} \quad (6.24)$$

We now apply this technique to infer system states and model parameters.

Inferring System States

To infer system states $\hat{\mathbf{x}}$ using system matrix $\mathbf{A}(\varphi)$, \mathbf{B} , $\mathbf{C}(\varphi)$, inputs $\underline{\mathbf{u}}$ and observations $\underline{\mathbf{y}}$, we must first write the dynamics as linear function system state \mathbf{x} .

$$\Gamma_x \mathbf{x} = \gamma_x \quad (6.25)$$

Making equation (6.15) fit this form is possible using a first order discretization by which each sampled state is related to the previous one with a linear approximation $\dot{\mathbf{x}} = \frac{1}{dt} (\mathbf{x}_i - \mathbf{x}_{i-1})$. For three measurement samples $\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2$, equation (6.25) would become:

$$\begin{bmatrix} \mathbf{C}(\varphi) & & & \\ -(I + A(\varphi)dt) & I & & \\ & \mathbf{C}(\varphi) & & \\ & -(I + A(\varphi)dt) & I & \\ & & \mathbf{C}(\varphi) & \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_0 \\ \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ Bu_0 \\ y_1 \\ Bu_1 \\ y_2 \end{bmatrix} \quad (6.26)$$

Therefore, the least squares estimate of system state is:

$$\hat{\mathbf{x}}_{LS} = (\Gamma_x^T \Gamma_x)^{-1} \Gamma_x^T \gamma_x \quad (6.27)$$

Equation (6.29) is in fact the discrete first-order Kalman Filter, but without weighting coefficients or the elegant recursive formulation. We will see in the later sections how adding weighting coefficients enables us to trust good measurements more than others by weighting them more heavily, optimally balancing our trust between the system model's predictive power and noisy measured values.

Inferring Model Parameters

Inferring system parameters from known states $\underline{\mathbf{x}}$ and observations $\underline{\mathbf{y}}$ is another simple process. Once again, we must first write equations as linear functions of the parameters we wish to find, in this case φ .

$$\Gamma_\varphi \varphi = \gamma_\varphi \quad (6.28)$$

To do this, we must rewrite \mathbf{A} and \mathbf{C} to be functions of \mathbf{x}_i instead of φ . Let's call these modified matrices \mathbf{A}^* and \mathbf{C}^* , since in some sense they are duals to the originals because they are linearly related to the originals by scalars. For the moment, we ignore their detailed structure and fill in equation (6.28) as

$$\begin{bmatrix} \mathbf{C}^*(\mathbf{x}_i) \\ \mathbf{A}^*(\mathbf{x}_i) \end{bmatrix} [\varphi] = \begin{bmatrix} \mathbf{y}_i \\ \dot{\mathbf{x}}_i - \mathbf{B}\mathbf{u}_i \end{bmatrix} \quad (6.29)$$

The now-familiar expression for the least squares estimate of φ is therefore

$$\hat{\varphi}_{LS} = (\Gamma_\varphi^T \Gamma_\varphi)^{-1} \Gamma_\varphi^T \underline{\mathbf{x}} \quad (6.30)$$

Some Nuances and Caveats

In this brief treatment of least squares, we have glossed over several important issues that must now be addressed. Most obviously, up to this point we have assumed that either \mathbf{A}_i or \mathbf{x}_i was known perfectly and could be treated as independent variables when solving for the other. However, this is of course not true – both variables clearly affect the other and neither is known perfectly. We would do better to estimate both quantities at the same time. In estimation theory, Stein's Phenomenon states that when more than three parameters are estimated simultaneously, combined estimators will be more accurate than any method which handles parameters separately.

An obvious example of this effect appears in practice when trying to infer model parameters using equation (6.30), since we do not know the elements of \mathbf{x} perfectly – we may not even be able to observe all of them!

Instead, we must first estimate $\hat{\mathbf{x}}$ using equation (6.29), then use that to estimate φ . Unfortunately, this use of one estimate to find another creates quite suboptimal estimates for $\hat{\varphi}$ and $\hat{\mathbf{x}}$ that may take extensive iteration to correct. A better solution is to estimate both at the same time using a nonlinear model, as we shall see.

Another topic that we have left unconsidered is the choice of loss function. Although it is wonderfully convenient to have such an analytic expression like equation (6.24) from which we may immediately compute our answer, in some ways the decision to use an L_2 loss function seems rather arbitrary in retrospect. Why should a quadratic loss function be the best choice? Why shouldn't we minimize the sum of all the residuals using a *Manhattan norm*, defined as $L_1(\mathbf{r}) = \sum \|\mathbf{r}\|$? Alternatively, why shouldn't we minimize the *worst-case error* using $L_\infty(\mathbf{r}) = \max \|\mathbf{r}\|$?

In fact, using a quadratic loss function with unweighted least squares has a very specific meaning when considered from the Bayesian viewpoint. It is equivalent to stating that we believe ϵ to be a vector whose elements are drawn from independent distributions all possessing the same variance. According to the Gauss-Markov theorem, these distributions need not even be Gaussian for least squares to find the best solution. We'll understand the significance of this statement more clearly in the next section.

Unfortunately, assuming that each parameter is drawn from distributions with equal variances is generally a poor decision to make because different parameters and dimensions of a physical model will usually have wildly different variances or units. Comparing milliamps and meters with the same least squares residuals will generally give poor results. Comparing noisy measurements and clean measurements equally gives undue credence to the noisy signal. For this reason, we now turn to Maximum Likelihood method, which is equivalent to weighted least squares under reasonable assumptions.

6.1.6 Maximum Likelihood (ML)

Maximum likelihood estimators are based a simple concept: if you can only choose one set of model parameters φ , you should select the model parameters which have the highest probability of producing the data that you observed. This is known as maximizing the likelihood; hence the name *maximum likelihood* method.

$$\hat{\varphi}_{ML} = \underset{\varphi}{\operatorname{argmax}} P(D|\varphi) \quad (6.31)$$

ML is useful because it is invariant under reparameterizations of the problem and so represents the best possible choice of a single set of parameters. It was the same technique used to infer the sufficient statistics of a Gaussian in section 6.1.3.

We will now make a link between ML and Gaussian systems. TODO:

- The weighted sum of two gaussians
- Relation between gaussian's log parabolic and squared error
- Relation between covariances and squared error weighting
- If the errors are correlated, the resulting estimator is BLUE if the weight matrix is equal to the inverse of the variance-covariance matrix of the observations.

In the light of Bayes's Theorem, we can see that weighted least squares techniques is equivalent to maximum likelihood estimation using Gaussian variables \mathbf{x} with covariance Q_x . In this respect, maximum-likelihood is similar to LS, but allowing for more complex and useful assumptions about the variances and dependencies of each vector dimension. The fact that the log likelihood of a Gaussian is quadratic and that the quadratic loss

function used in least squares is no coincidence either – in some sense they represent some statement about the variances of the random variables.

Let \mathbf{Q} be the covariance matrix of the random variables, and

$$\varphi_{ML} = \left(\Gamma^T \mathbf{Q}^{-1} \Gamma \right)^{-1} \Gamma^T \mathbf{Q}^{-1} \gamma \quad (6.32)$$

6.1.7 Maximum a Posteriori (MAP)

MAP allows inclusion of further information not gleaned explicitly from the data; it allows the researcher to explicitly put their beliefs about the system as PDFs. These *priors* can then be manipulated using the machinery of Bayes' Theorem to predict a result. Unlike ML estimation however, MAP is not invariant under reparameterizations – depending on how the problem is formulated, you may get different MAP estimates. Furthermore, MAP estimates are often atypical of the whole distribution. It is also unfortunately easy to create bimodal distributions from overly specific priors.

For example... TODO

6.1.8 Marginalization

ML and MAP estimation methods are extremely useful in practice if we must predict things using only one set of parameters. For basing conclusions purely on the data, using ML is the best choice. If we have some prior information that we wish to include in the model, MAP can help as well (if you are more careful).

But they are hardly the end to the Bayesian story. For the truly most optimal prediction given our data and priors, we should also consider all the other other models that COULD have generated the data – albeit with lesser likelihood than our most likely model – and revise our estimates by combining the weighted probabilities of all the other models as well. This process is called *marginalization* and typically requires numerical integration techniques for nonlinear systems without well-behaved PDFs. Swarm and ensemble prediction methods are another sampling techniques based upon this principle.(?, ?)

TODO: Example of marginalization.

However, due to the complexity and computational cost of such methods, as well as the plentiful and abundant data that mechanical systems in this thesis provide when sampled at 1000Hz, the advantages of marginalization outweigh the costs and will not be considered further.

6.1.9 Nonlinear Estimation

TODO: Show that gaussians are no longer conjugate to themselves through a nonlinear transformation, show the unscented transform.

6.2 HypoSEA Observer Design

This section documents the derivation of the HypoSEA model-based observer. Readers unfamiliar with bond graph notation may wish to spend a moment briefly reviewing the concepts of duality, causality, and systems theory as presented in Appendix A.

Table 6.1: Hypothesized physical effects affecting HypoSEA system dynamics.

Domain	Magnitude	Description	Modeled?
Voltage	~ 36 V	Nominal battery supply voltage	yes
	± 50 V	Back-EMF voltage developing on motor	yes
	< 3 V	Battery voltage varies with charge state	no
	< 1.5 V	Cumulative effect of H-bridge diode drops	no
Inductance	~ 3 mH	Inductive effect of stator coils	yes
	< 0.1 mH	Estimated inductive effect of motor leads, PCB traces	no
Capacitance	~ 0 F	PCB trace stray capacitance	yes
Resistance	$2.5\text{-}4 \Omega$	Stator resistance varies with temperature	yes
	$< 0.05 \Omega$	Estimated battery + wiring series resistance from load tests	no
	$< 0.01 \Omega$	Estimated MOSFET series resistance from datasheet	no
Inertia	$> 0.5 \text{ Kg}\cdot\text{m}^2$	Joint-side link Inertia	yes
	$> 5.9\text{E-}5 \text{ Kg}\cdot\text{m}^2$	Aggregate rotor + planetary gear inertia	yes
Elasticity	$\sim 10\text{N}/\text{rad}\cdot\text{m}/\text{rad}$	Hypocycloid linear stiffness	yes
	$> 1000\text{Nm}/\text{rad}$	Planetary gear + shaft elasticity	no
Friction	$0.1 \text{ N}\cdot\text{m}/\text{rad}/\text{s}$	Rotor viscous friction	yes
	$\sim 0.03 \text{ N}\cdot\text{m}/\text{rad}/\text{s}$	Hypocycloid mechanism friction + spring restitution loss	yes
	$\sim 0.1 \text{ N}\cdot\text{m}/\text{rad}/\text{s}$	Joint-side viscous friction	yes
	$< 0.1 \text{ N}\cdot\text{m}$	Rotor dry friction	no
	$\sim 1 \text{ N}\cdot\text{m}$	Joint-side dry friction	no
Coefficients	0.3	Motor torque coefficient	yes
	± 0.05	Motor torque coefficient variance with rotor position	no

6.2.1 Selecting Model Complexity

Selecting the level of detail to include in a system model is one of the most difficult and important decisions facing theorists – typically it is guided by intuition and experience more than any specific methodology. On the one hand, we want to keep the model as simple as possible for analytic reasons, computational speed, and researcher sanity. On the other, we also want the model to be detailed enough to accurately predict the behavior of the system being modeled.

The approach followed when developing the HypoSEA model was simply to estimate the magnitude of expected effects in each domain, and rank them by descending numerical order. This is shown in Table 6.1.

Four effects were not modeled although they have significant impact on the dynamics of the system: (1) dry friction at the rotor; (2) dry friction at the output joint; (3) the variance of the motor torque coefficient as it rotates due to changes in magnetic flux linkage; (4) The variation in stator resistance due to heating effects. The former two are highly nonlinear and difficult to model. The third is nonlinear but averages out to zero effect over larger motions (Figure ??). And once steady state has been reached, the stator resistance is assumed to be approximately constant.

As the PWM voltage sent to the motors is digitally created and assumed to be perfectly controllable due to the high resolution timer, it is a quantity assumed to be perfectly known. The external torque due to gravity τ_{ext} in the test configuration of Figure ?? is also assumed perfectly known because it could be calculated very precisely from simple measurements regarding the center of mass of the output link.

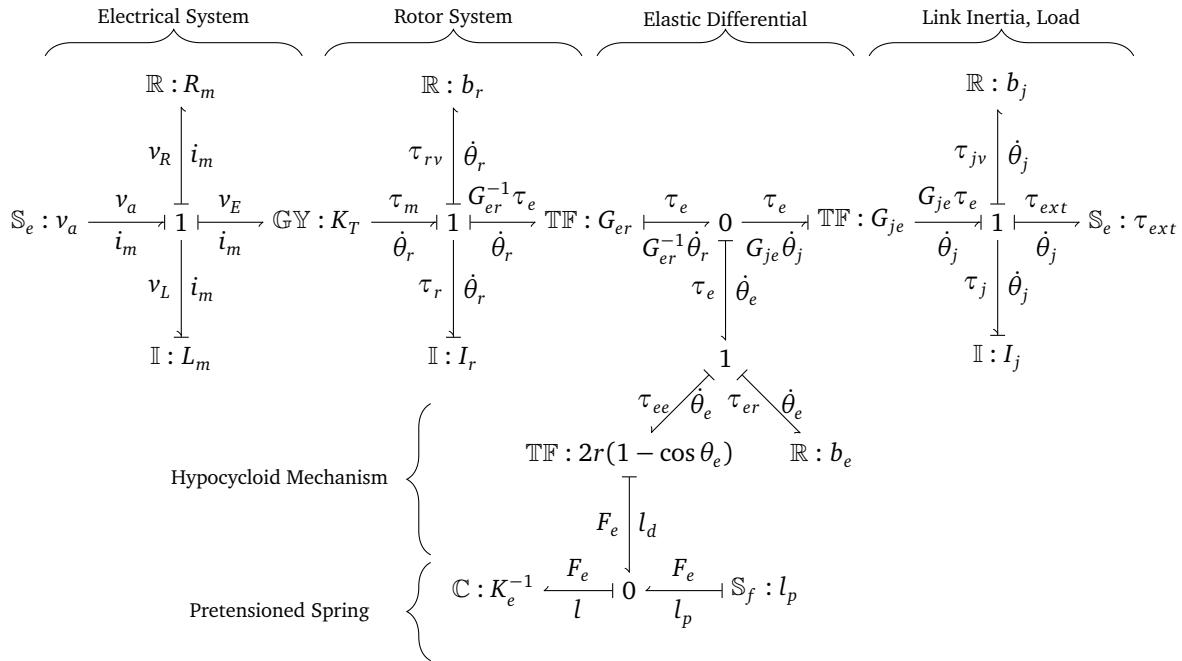


Figure 6.3: Bond Graph of the HypoSEA system model.

TODO: Plot of the rotor current vs position with no load.

After all of these modeling decisions had been made, the bond graph shown in Figure 6.3 was drawn. The reader is invited to refer to Appendix A for more information if unfamiliar with bond graphs, duality.

6.2.2 Derivation of HypoSEA System Matrices

The bond graph of Figure 6.3 gives us all the information we need to derive the Hamiltonian dynamics of the HypoSEA. As mentioned earlier, we will attempt to manipulate the system dynamics into the form of equation (6.15). We begin by grouping the effort and flow sources \mathbb{S} into input vector \mathbf{u} .

$$\mathbf{u} = \begin{bmatrix} v_a \\ \tau_{ext} \end{bmatrix} \quad (6.33)$$

The energetic state of the \mathbb{C} and \mathbb{I} elements must be packaged into state vector \mathbf{x} . Ordinarily, we would measure the state of the spring K_e by its deflection l , but because the hypocycloid mechanism has a singularity which prevents l from uniquely determining the sign of the deflection of angle θ_e , we cannot use l as a suitable state to describe the spring deflection. Instead, we use θ_e directly as the state and treat the elastic element as a nonlinear stiffness $\tilde{K}(\theta_e)$. This approximation is shown in Figure ??.

$$\mathbf{x} = \begin{bmatrix} \Phi_m \\ h_r \\ h_j \\ \theta_e \end{bmatrix} \quad (6.34)$$

Each \mathbb{C} and \mathbb{I} component on the bond graph has been written with integral causality, meaning that we may use the following useful relations to help derive the dynamical equations.

$$\Phi_m = L_m i_m \quad (6.35)$$

$$h_r = I_r \dot{\theta}_r \quad (6.36)$$

$$h_j = I_j \dot{\theta}_j \quad (6.37)$$

$$\theta_e = \tilde{K}^{-1}(\theta_e) \tau_{ee} \quad (6.38)$$

The dynamics of the components is thus given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\Phi}_m \\ \dot{h}_r \\ \dot{h}_j \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_m \\ \tau_r \\ \tau_j \\ \dot{\theta}_e \end{bmatrix} \quad (6.39)$$

We now go through each term of $\dot{\mathbf{x}}$ and expand each signal until it is expressed in terms of \mathbf{x} and \mathbf{u} . Starting with the 1-junction in the electrical domain, we solve for v_m :

$$0 = v_a - v_R - v_E - v_m \quad (6.40)$$

$$v_m = v_a - R_m i_m - K_T \dot{\theta}_r \quad (6.41)$$

$$= v_a - \frac{R_m}{L_m} \Phi_m - \frac{K_T}{I_r} h_r \quad (6.42)$$

The 0-junction in the elastic differential yields the expression for $\dot{\theta}_e$:

$$0 = G_{er}^{-1} \dot{\theta}_r - G_{je} \dot{\theta}_j - \dot{\theta}_e \quad (6.43)$$

$$\dot{\theta}_e = \frac{1}{G_{er} I_r} h_r - \frac{G_{je}}{I_j} h_j \quad (6.44)$$

The 1-junction in the elastic differential, needed to model friction across the differential, is a small but important force which directly couples the rotor and link inertias.

$$0 = \tau_e - \tau_{ee} - \tau_{er} \quad (6.45)$$

$$\tau_e = \tilde{K}(\theta_e) \theta_e + b_e \dot{\theta}_e \quad (6.46)$$

$$= \tilde{K}(\theta_e) \theta_e + \frac{b_e}{G_{er} I_r} h_r - \frac{b_e G_{je}}{I_j} h_j \quad (6.47)$$

The 1-junction connected to the rotor inertial gives us the expression for τ_r :

$$0 = \tau_m - \tau_{rv} - \tau_r - G_{er}^{-1} \tau_e \quad (6.48)$$

$$\tau_r = K_T i_m - b_r \dot{\theta}_r - G_{er}^{-1} \left(\tilde{K}(\theta_e) \theta_e + \frac{b_e}{G_{er} I_r} h_r - \frac{b_e G_{je}}{I_j} h_j \right) \quad (6.49)$$

$$= \frac{K_T}{L_m} \Phi_m - \frac{b_r}{I_r} h_r - \frac{\tilde{K}(\theta_e)}{G_{er}} \theta_e - \frac{b_e}{G_{er}^2 I_r} h_r + \frac{b_e G_{je}}{G_{er} I_j} h_j \quad (6.50)$$

$$= \frac{K_T}{L_m} \Phi_m - \left(\frac{b_r}{I_r} + \frac{b_e}{G_{er}^2 I_r} \right) h_r - \frac{b_e G_{je}}{G_{er} I_j} h_j - \frac{\tilde{K}(\theta_e)}{G_{er}} \theta_e \quad (6.51)$$

Similarly, the 1-junction of the link inertia can be manipulated to yield τ_j :

$$0 = G_{je} \tau_e - \tau_{jv} - \tau_j - \tau_{ext} \quad (6.52)$$

$$\tau_j = G_{je} \left(\tilde{K}(\theta_e) \theta_e + \frac{b_e}{G_{er} I_r} h_r - \frac{b_e G_{je}}{I_j} h_j \right) - b_j \dot{\theta}_j - \tau_{ext} \quad (6.53)$$

$$= G_{je} \tilde{K}(\theta_e) \theta_e + \frac{G_{je} b_e}{G_{er} I_r} h_r - \left(\frac{b_j}{I_j} + \frac{b_e G_{je}^2}{I_j} \right) h_j - \tau_{ext} \quad (6.54)$$

All these expressions can now be combined to form matrices $\mathbf{A}(\varphi, \mathbf{x})$ and \mathbf{B} :

$$\mathbf{A}(\varphi, \mathbf{x}) = \begin{bmatrix} -\frac{R_m}{L_m} & -\frac{K_T}{I_r} & 0 & 0 \\ \frac{K_T}{L_m} & -\left(\frac{b_r}{I_r} + \frac{b_e}{G_{er}^2 I_r} \right) & -\frac{G_{je} b_e}{G_{er} I_j} & -\frac{\tilde{K}(\theta_e)}{G_{er}} \\ 0 & \frac{G_{je} b_e}{G_{er} I_r} & -\left(\frac{b_j}{I_j} + \frac{b_e G_{je}^2}{I_j} \right) & G_{je} \tilde{K}(\theta_e) \\ 0 & \frac{1}{G_{er} I_r} & -\frac{G_{je}}{I_j} & 0 \end{bmatrix} \quad (6.55)$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \quad (6.56)$$

The dynamics of the system has now been specified sufficiently to be of practical use. However, we can improve our understanding of the system and make system identification easier (section 6.2.4) by removing factors L_m^{-1} , I_r^{-1} , I_j^{-1} and $\tilde{K}(\theta_e)$ from the columns of \mathbf{A} , and defining a new scaled state vector ψ and system matrix $(\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}})$.

$$(\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}})\psi = \begin{bmatrix} -R_m & -K_T & 0 & 0 \\ K_T & -\left(b_r + \frac{b_e}{G_{er}^2} \right) & -\frac{G_{je} b_e}{G_{er}} & -\frac{1}{G_{er}} \\ 0 & \frac{G_{je} b_e}{G_{er}} & -\left(b_j + b_e G_{je}^2 \right) & G_{je} \\ 0 & \frac{1}{G_{er}} & -G_{je} & 0 \end{bmatrix} \begin{bmatrix} \frac{\Phi_m}{L_m} \\ \frac{h_r}{h_j} \\ \frac{h_j}{I_j} \\ \tilde{K}(\theta_e) \theta_e \end{bmatrix}$$

We can now see that $\dot{\mathbf{x}}$ is the vector of *internal flow variables*, and that ψ vector of the *internal effort variables* of the system. Their inner product is power. In the same way, the *external port effort and flow variables* are \mathbf{u} and $\mathbf{z} = \mathbf{B}^T \mathbf{u}$.

$$\psi = \begin{bmatrix} \frac{\Phi_m}{L_m} \\ \frac{h_r}{I_r} \\ \frac{h_j}{I_j} \\ \tilde{K}(\theta_e)\theta_e \end{bmatrix} = \begin{bmatrix} i_m \\ \dot{\theta}_r \\ \dot{\theta}_j \\ \tau_{ee} \end{bmatrix}$$

The function of every 1-port passive element can now be clearly seen by splitting $(\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}})$ into $(\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}}) = \mathbf{A}_{\mathbb{R}} + \mathbf{A}_{01}$, where $\mathbf{A}_{\mathbb{R}}$ is the negative semidefinite dissipative matrix and \mathbf{A}_{01} is the antisymmetric matrix that describes the connectivity, gearing, and friction coupling of the system.

$$\mathbf{A}_{\mathbb{R}} = \text{diag } \mathbf{a}_{\mathbb{R}} = \text{diag} \left[-R_m, -\left(b_r + \frac{b_e}{G_{er}^2}\right), -\left(b_j + b_e G_{je}^2\right), 0 \right]$$

$$\mathbf{A}_{01} = \begin{bmatrix} 0 & -K_T & 0 & 0 \\ K_T & 0 & -\frac{G_{je}b_e}{G_{er}} & -\frac{1}{G_{er}} \\ 0 & \frac{G_{je}b_e}{G_{er}} & 0 & G_{je} \\ 0 & \frac{1}{G_{er}} & -G_{je} & 0 \end{bmatrix}$$

The PHS dynamics could therefore also be written as

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{A}_{\mathbb{R}} + \mathbf{A}_{01})\psi + \mathbf{B}\mathbf{u} \\ \mathbf{z} &= \mathbf{B}^T\psi \end{aligned}$$

By exploiting such symmetries programmatically, the number of arithmetic operations required to evaluate the system dynamics on a computer may be reduced.

6.2.3 State Measurement

The structure of the HypoSEA makes it impractical to directly measure every dimensions of \mathbf{x} directly. Instead we assume that we only observe certain quantities \mathbf{y} which are related linearly to the system state.

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(\varphi)\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned}$$

The three sensors on the HypoSEA provide three measurements: motor current i_m , rotor position θ_r , and elastic deflection θ_e . These quantities are unfortunately scattered around the system equations: θ_e is an element of \mathbf{x} , but i_m is found in ψ , and θ_r is nowhere to be seen in the dynamical equations.

The way around this difficulty is to try to relate measurements \mathbf{y} to ψ instead of \mathbf{x} . Because θ_r is the highest resolution signal in the HypoSEA, taking its derivative numerically to get $\dot{\theta}_r$ is not a terrible sacrifice. Also, if we realize that there is a simple nonlinear relationship between θ_e and τ_{ee} that is independent of other considerations, we may perform a local transformation to remove the nonlinearity from rest of the system and reduce the complexity of system identification. This is not a difficult task since we have an analytic expression for $\tilde{K}(\theta_e)$ from chapter TODO:

$$\tau_{ee}(\theta_e) = K_e(2r)(\sin \theta_e)(2r(1 - \cos \theta_e) + l_p)$$

$$\tilde{K}(\theta_e) = \frac{\partial \tau_{ee}}{\partial \theta_e} = 8K_e r^2 \sin(\theta)^2 + (4K_e r^2 + 2K_e pr) \cos \theta - 4K_e r^2$$

Indeed, $\tau_{ee}(\theta_e)$ may be accurately calibrated offline and a lookup table created so that it requires very little computation.

Now we may write our measurement matrix as by inspection, and linearly relate it to ψ .

$$\mathbf{y} = \begin{bmatrix} i_m \\ \dot{\theta}_r \\ \tau_{ee} \end{bmatrix} = \mathbf{C}'\psi$$

$$\mathbf{C}' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

All of this greatly simplifies our system since it removes the dependency of $\mathbf{A}(\varphi, \mathbf{x})$ upon system state, and allows us to use the beautiful structure matrix $(\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}})$ in the sequel. Our system equations thus far are now

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}})(\varphi)\psi + \mathbf{B}\mathbf{u} = \mathbf{A}(\varphi)\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}'\psi \end{aligned}$$

With everything else about the model more or less specified, now need to consider how to find the system parameters φ .

TODO:

We want to weight things based on their effect on POWER, so that it is a common currency instead of flow or effort...right?

6.2.4 Inference of Model Parameters

Power constraints Scaled by effect on power uncertainty Weighting recent events MORE with an update rule

6.2.5 Real Time Inference

Kalman Filter details.

6.2.6 Fourth Order Fixed Lag Smoother

For later data analysis.

6.3 Estimation Performance Comparison

6.4 Power Flow Estimation

6.4.1 Power when consumption

6.4.2 Power Generation

6.4.3 Approximating Derivatives Using Finite Differences

Numerical differentiation is the phrase used to describe algorithms that estimate the derivative of a mathematical function at a point using values of the function around that point. It is a very useful technique for using arithmetic operations to determine derivatives of a function whose analytical form is not known.

Derivatives may be estimated using only information taken in an evenly-sampled manner from the past, in which case they would be called “backward finite differences”. Since such a filter is causal, it is extremely useful for real time control but it will inevitably lag the true value of the derivative by some amount.

A better approximation can be made once we have actually seen the data that immediately follows a given point. Then, we may use nearby data points both before and after the desired point where the derivative is to be approximated. Such approximations are called “central finite differences” and approach the true value of the derivative more quickly due to their symmetric nature.

Both approximations are based on a Taylor Series expansion of the derivative, truncated at the desired order.

TODO: Show the filter's reactions to a step response.

TODO: Show each filter's reactions to a step response with additive noise.

Perfect power conservation is problematic. Iterated correction to estimated state in the direction that minimizes unmodeled energy flows (in watts) Problem: I can always perturb the estimated state to make this true, yet there may INDEED be external power flows onto the device. Iterated correction to model parameters in the direction that minimizes unmodeled energy flows (in watts) might be interesting.

6.4.4 Backward Finite Differences

The simplest way to estimate the first difference in real time is with a first order backward finite difference, which takes the difference between the present sample and the previous one:

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

This is fine for many things, but can be a little noisy. A second order approximation would use two previous samples:

$$f'(x) = \frac{3f(x) - 4f(x-1) + f(x-2)}{2h}$$

however this suffers from even more phase lag.

For online estimation in the HypoSEA, two fourth order filters were used:

$$\begin{aligned} f'(x) &= \frac{1}{h} \left(\frac{25}{12}f(x) - 4f(x-1) + 3f(x-2) - \frac{4}{3}f(x-3) + \frac{1}{4}f(x-4) \right) \\ f''(x) &= \frac{1}{h^2} \left(\frac{15}{4}f(x) - \frac{77}{6}f(x-1) + \frac{107}{6}f(x-2) - 13f(x-3) + \frac{16}{12}f(x-4) - \frac{5}{6}f(x-5) \right) \end{aligned}$$

Table 6.2: Backward finite difference coefficients.

Derivative	Order	$f(x)$	$f(x - 1)$	$f(x - 2)$	$f(x - 3)$	$f(x - 4)$	$f(x - 5)$	$f(x - 6)$
1	1	1	-1					
	2	3/2	-2	1/2				
	3	11/16	-3	3/2	-1/3			
	4	25/12	-4	3	-4/3	1/4		
2	1	1	-2	1				
	2	2	-5	4	-1			
	3	35/12	-26/3	19/2	-14/3	11/12		
	4	15/4	-77/6	107/6	-13	61/12	-5/6	
3	1	1	-3	3	-1			
	2	5/2	-9	12	-7	3/2		
	3	17/4	-71/4	59/2	-49/2	41/4	-7/4	
	4	49/8	-29	461/8	-62	307/8	-13	15/8

The coefficients for selected backward finite differences are listed in Table X.

6.4.5 Central Finite Differences

We could approximate the derivative at x better by using the secant line for the point in front and behind it. (Figure 2) This is also called the 'three point method':

$$\frac{df}{dx} = \frac{f(x+h) - f(x-h)}{2h}$$

however it is no longer a causal filter because it looks into the future when it uses $f(x+h)$.

What if instead of just using one point on each side of the derivative, we used two points on each side?

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$$

This was the filter actually used before doing least-squares analysis on the HypoSEA parameters, along with its fourth order companion for the second derivative

$$f''(x) = \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h}$$

Central difference coefficients for several other orders are shown in Table X.

6.4.6 Approximate Derivatives with Irregular Sampling

The motor control board is synchronously sending out packets at regular intervals – but we have no guarantees that they will arrive. What can we do to estimate the derivative in the presence of missing packets?

Table 6.3: Central difference coefficients.

Derivative	Order	$f(x - 4)$	$f(x - 3)$	$f(x - 2)$	$f(x - 1)$	$f(x)$	$f(x + 1)$	$f(x + 2)$	$f(x + 3)$	$f(x + 4)$
1	2				-1/2	0	1/2			
	4			1/12	-2/3	0	2/3	-1/12		
	6		-1/60	3/20	-3/4	0	3/4	-3/20	1/60	
	8	1/280	-4/105	1/5	-4/5	0	4/5	-1/5	-4/105	-1/280
2	2				1	-2	1			
	4			-1/12	4/3	-5/2	4/3	-1/12		
	6		1/90	-3/20	3/2	-49/18	3/2	-3/20	1/90	
	8	-1/560	8/315	-1/5	8/5	-205/72	8/5	-1/5	8/315	-1/560
3	2			1	-4	0	-4	1		
	4		-1/6	2	-13/2	0	-13/2	2	-1/6	
	6	7/240	-2/5	169/60	-122/15	0	-122/15	169/60	-2/5	7/240

6.5 Least Squares

Once The model parameters are actually just partials of the Hamiltonian

What I want is to project the least-squared errors down into the per-component model, then do a nonlinear regression somehow to pick the value.

Per-component energy analysis, computation, etc without matrices

Per component nonlinearities (and regression to estimate the parameters)

Junction-based energy conservation (and total energy conservation)

Least squares on the energy error

6.5.1 Model Verification

Making a bunch of 2D plots of every residual vector should help suggest new components!

6.6 Weighted Least Squares

6.7 Recursive, Weighted Least Squares

This is also called the Kalman Filter.

6.7.1 The Kalman Filter

Essentially ML or weighted least squares, in recursive form.

What would an optimal linear filter look like, intuitively speaking? “Optimal” is a tricky word because it presupposes a standard of measurement that is objective, something that may not be true. Some researchers might consider an “optimal” filter to be one that gives the best result for the least amount of computation. Others might insist that “optimal” should be measured in terms of energy, entropy, or robustness.

For linear systems we can be fairly precise about our meaning. An optimal linear filter would provide the most likely estimate of the state given our model, perform as little computation at each step as possible, store as

little of the past history as needed, provide bounds upon which we might check our certainty, and be guaranteed to converge on the proper answer given enough data.

The Kalman filter satisfies all of these considerations, as we shall see.

Imagine that we have a discrete time dynamic system with the following structure

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

where A is the system transition matrix from step to step, B describes how inputs u affect the state, and w is the random process noise. This expresses the state-to-state transitions of a system, but in most systems we must also consider the inevitable error in our measurements.

$$z_k = Hx_k + v_k$$

where v is random measurement noise.

The Kalman Filter is an optimal filter if two facts are true:

1. A, B, H are constant linear matrices
2. w and v are white gaussian noise sources with covariances Q and R , respectively. "White" in this case means that the spectral content of the signal is roughly the same at all frequencies of interest.

Why do we need to assume that the noise sources are white and gaussian? The honest answer is: because it makes the mathematics easier! Push a gaussian through a linear transformation and out comes another gaussian with predictable characteristics, a convenience that is not true for all probability distributions. More importantly, it is well known fact that when a number of random, independent probability densities are added together, their summed effect will resemble a gaussian, regardless of the shape of the individual densities. A gaussian is a good "general case" distribution.

6.7.2 Mixing Gaussians

We can say these things about combining two samples z_1 and z_2 from gaussian distributions:

$$\mu = \frac{\sigma_2^2 z_1 + \sigma_1^2 z_2}{\sigma_1^2 + \sigma_2^2}$$

$$\sigma^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right)^{-1}$$

These estimates provide the optimal maximum likelihood, weighted least squares, and linear estimate. Note that if one of the measurements has a larger σ^2 , then it can be 'trusted' less, and so affect the mean less. Compare these two equations to the kalman filter

$$\hat{x}_k = \hat{x}_{k-1} + K [z_k - \hat{x}_{k-1}]$$

$$K = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

And you may see how it weights new values compared to the old ones in a way that is also 'optimal' in some sense.

6.7.3 Operation of the Kalman Filter

The following table is a useful summary of the operation of the KF and EKF.

	Operation	Kalman Filter	Extended Kalman Filter
1	Predict next state	$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$	$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)$
2	Project error covariance ahead	$P_k^- = AP_{k-1}A^T + Q$	$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$
3	Update Kalman gain	$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$	$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$
4	Update estimate when measurement z_k is seen	$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$	$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0))$
5	Update error covariance.	$P_k = (I - K_k H) P_k^-$	$P_k = (I - K_k H) P_k^-$

6.8 Extended Kalman Filtering on Instantaneous Powers

Levenberg-Marquardt algorithm for gradient descent on pg 242 of neural network book looks interesting, but boils down to

This goes with chapter X on Bond Graphs. Basic idea: we get the instantaneous powers and constraints from the Bond Graph, then use that.

Plotting it x-y for each thing, we can see how easily it is approximated.

First, you go “backwards” and fit the model to the data using. Then you go “forwards” and fit your data to the model. Both use Kalman filters, both are optimal for linear systems, both respond to noise safely, both are stable.

6.9 Interpretation

Several interesting things:

The filter is pretty stable, because it uses a lot of data.

6.10 Real World Problems Encountered

Noisy Numerical Derivatives

Higher order filtering is one thing.

Yes. But don’t worry, least-squares fitting will do the best it can even with this noisy data.

Energy Recovery Only Occurs at High Speed

Energy recovery only occurs if voltage across the motor is higher than v_s plus the two diode drops. My motor simply doesn’t spin that fast during normal operation.

I don’t see a way around this! I wish this were more controllable, and I wish that we could get current to flow in the battery immediately, but this would require more circuitry.

Current Ripple

There is a great deal of current ripple that correlates with position. However the motor rotates too quickly to perform active compensation for the current ripple unless it was put on

Thoughts:

Ideally, we want to forsake loss functions and work directly on PDFs.

Doing gradient ascent on the probability density function of each parameter would eventually lead you to the most likely model, regardless of linearity or not, if the PDF is unimodal. A second-order method could really speed up your approach to that point. If the PDF isn't unimodal, you will reach a sub-optimal model. But that can't be helped I guess!

Our estimation (of the past, present, and future) based on the data should actually rely on the marginalized prediction of all the models.

It wouldn't be too hard for the model and data to become self-consistent and still be inaccurate.

Any assumptions must be explicit, but they can really speed up estimation by reducing exploration space. We may have n parameters, but m equality constraints at each datapoint. Energy conservation constraints are particularly useful: several will appear!

I see no reason why the process couldn't be at least somewhat recursive. New sensor data comes in This (very slightly) changes the estimated model parameter PDFs (use lots o' data) The marginalized predictions give the 'actual state' (use just last 10 pieces of data) It seems like a useful trick: the most model can be estimated over all the data (cheap, fast, recursive) the most likely state can be marginalized (expensive), but over just a few data (cheap)

Alternatively, Given a new piece of data Marginalize its interpretation using existing model PDFs Just one data point, but quadratic marginalization of predictions needs 3^n calcs Very slightly update the model PDFs based on this data. In a nutshell, we basically do incremental updates to both of these. The model PDFs should change only once with new data, if it were perfect The interpretation of the data should change only once with the PDF I don't really see a difference between model and data, actually! Was lisp the future, mixing code and data? We want to do a gradient descent on both model and data SIMULTANEOUSLY!!!! The two are related LINEARLY at an infinitesimally small point So some sort of dual thing could work.

The hand-waving idea: New data comes in Take a step in both dual spaces, estimating marginalized state and all-new PDFs Recurse until state and PDF stabilize, and you have reached information optimality.

You shouldn't estimate the state before the model, because it's stupid. You may estimate the model

GRUMBLING: Global Linearity is just a mathematical convenience, but local linearity is always true. Least squares is a mathematical convenience, but quadratics are useful. Measuring the variance is a mathematical convenience for gaussians. (See: Anscombe's Quartet)

Find these papers: "Neural dual extended Kalman Filtering" by E.A. Wan and A.T. Nelson. "The Unscented Bayes Filter" by E.A. Wan and R. Van der Merwe. (<http://cslu.cse.ogi.edu/nsei/>)

(Hyon & Emura, 2005) is not skimmable, but seems to show a hamiltonian energy preserving controller.

Several controllers have been developed for hopping robots that use Hamiltonian formulations. (Ishikawa, Neki, Imura, & Hara, 2003) presents an example controller for a Hamiltonian formulation of a linear hopping robot. They present an interconnection and damping assignment control, which uses the positive semidefinite damping matrix to keep the system energy always decreasing after a disturbance, yet still oscillating in the way the desire a hopping robot to work. In contrast, (Sakai & Stramigioli, 2007) presents a method for stabilizing a port-controlled Hamiltonian system based not on damping matrix assignment, but rather by asymptotic stabilization.

(Holoborodko, 2008) presents several so-called "Noise robust" differentiators for signal estimation. An interesting and simple approach which discards the high frequency content of a signal even more efficiently than

minimax and least squares differentiation techniques. Although this approach is simple and may be useful for information-rich signals not based on a physical model, it is fundamentally limited. High frequency information may be treated as “noise” in some applications, but for others it may signal an important event such as an impact. Throwing away the information indiscriminately is not a good policy.

Can we do better if we use a model? Yes, although non-parametric methods such as (Calinon, Guenter, & Billard, 2007) suggests that gaussian mixture models provide a good way of succinctly encoding many systems.

(De Luca et al., 2006) uses linear filters to detect collisions (as being different than gravity torques?).

NOISE DISTRIBUTIONS: Encoder noise is NOT gaussian. It's fairly directional, in fact. It lags the original signal by a certain amount. It's a flat value between the last value heard and the “next” one in the direction you are spinning. Only when velocity is very low are you uncertain in BOTH directions.

6.11 Performance Comparison of Numerical Differentiation

Most filtering methods treat the problem of numerical differentiation as one of removing the high frequency information from a signal. To that end, several techniques exist to smooth out a numerically differentiated signal: moving average, low pass filter, or polynomial approximation. All three of these techniques can be formulated in terms of polynomial approximation. The basic idea is often to approximate N points of data with a polynomial of degree M . If $M \ll N$, this polynomial must discard data, hopefully smoothing out the noise and capturing the essence of the signal. Chebyshev polynomials are easily integrated or differentiated, using the coefficients that approximated the original value(?, ?). Savitky-Golay smoothing filters fit a polynomial of degree M using N_L points to the left and N_R points to the right of each desired $x(?, ?)$.

However, from a Bayesian perspective it is almost always better to avoid pre-processing your data in this way. If we have a parametric model such as the one developed for the HypoSEA, we should just use the raw data and avoid the smoothing procedure entirely.

CHAPTER 7

Asynchronous, Concurrent, Functional Programming

This chapter describes the unusual style which the software for JIMI. This chapter is not yet fully written (see the note in the previous chapter), although the software is functional and being used. Screen shots of the software are shown in the following figures, although it is more difficult to express the controller software that is not yet represented by a GUI.

The software has several qualities that are desirable for research in robotics:

- All software runs in soft real time, including logging, analysis, visualization, and plotting.
- All code is design for asynchronous, low-latency transactions that are atomic, consistent, and independent.
- Any function may be loaded in and out of the system without recompiling or even pausing controller operation.
- All code is thread-safe and is often executed concurrently on multiple processors.
- Errors in one thread are contained and do not bring down other threads of computation.
- Data can be created that automatically updates itself based upon changes to its dependencies, creating dataflows that propagate changes from one part of the program to the next.

7.1 Motivation

7.1.1 Better Abstractions

Mathematics and programming share much in common. In mathematics we measure the elegance of an abstraction according to its generality, concision, and simplicity of its notation. For example, James Clerk Maxwell's equations for non-relativistic electromagnetic fields in a vacuum were written using X,Y, and Z coordinates () .

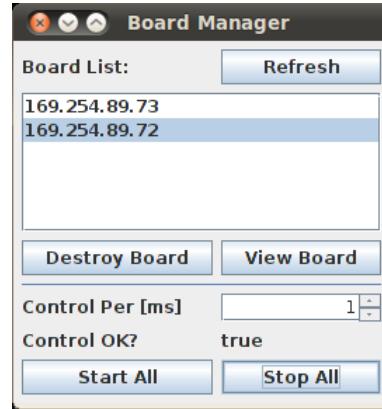


Figure 7.1: The Boardmanager discovers motor control boards via UDP, instantiates TCP connections with each, and initializes each board based upon its IP.

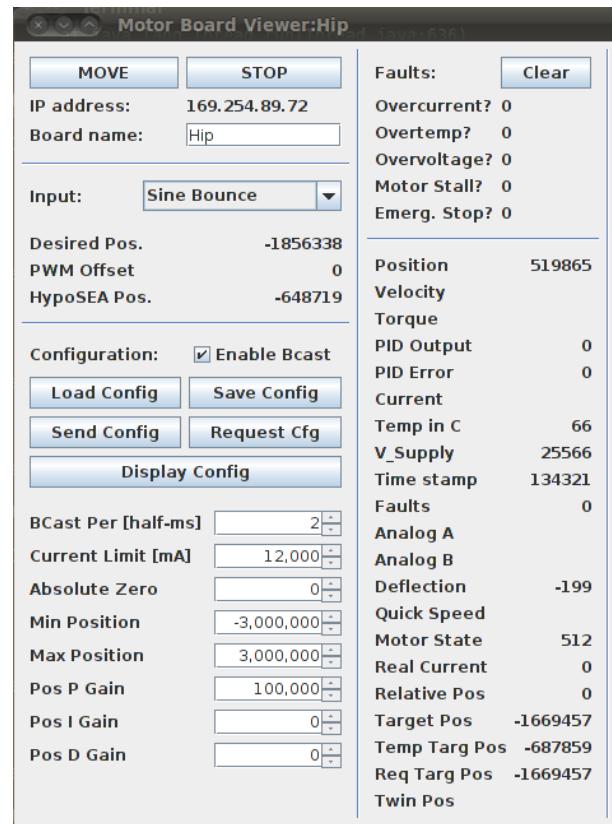


Figure 7.2: Each board has a model-view-controller (MVC) architecture, and this board-view GUI may be connected to an underlying motor board communication object. This GUI is dataflow based, so that changes to the state of the board or GUI automatically propagate to the other object.

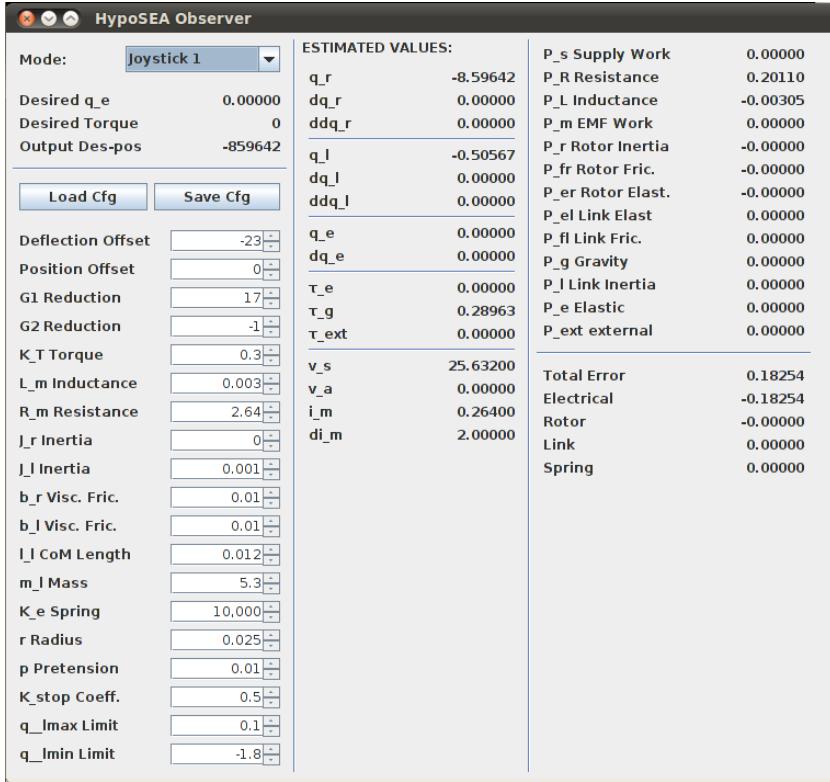


Figure 7.3: The HypoSEA observer GUI interfaces with an observer object to provide a convenient way to edit model parameters, view the predicted state of each HypoSEA and the power flows between the rotor, inductor, resistor, spring, and external load.

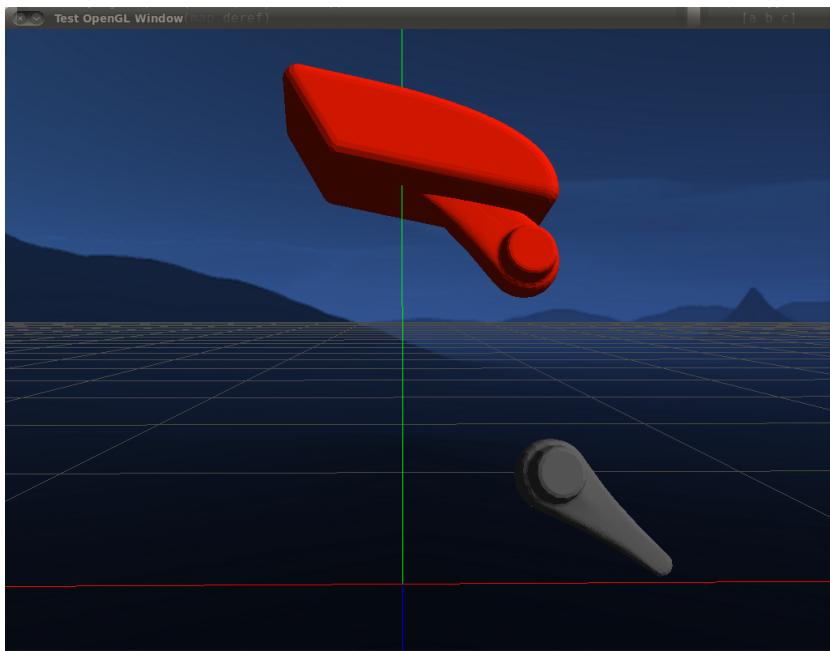


Figure 7.4: The OpenGL visualizer shows the estimated state of the robot in real time based on sensor fusion from a Vicon camera system and HypoSEA-v2 sensors. The thigh link is not shown in this screenshot because it is not being measured by the Vicon due to occlusion problems.

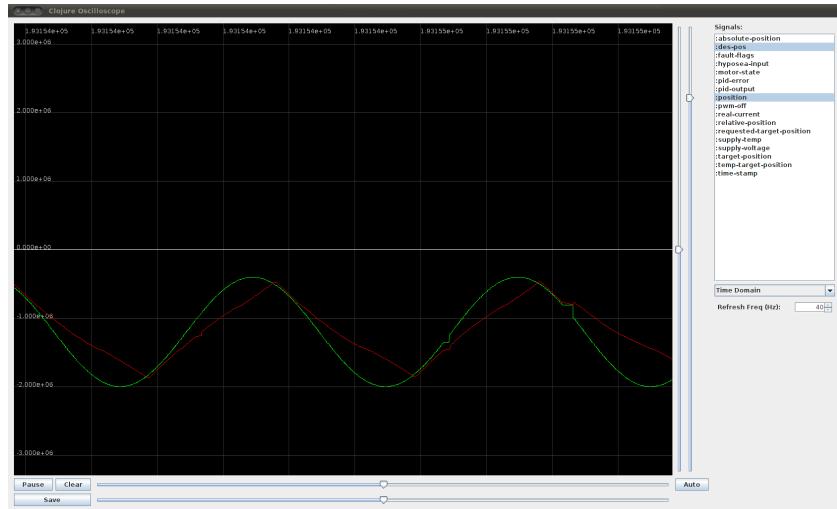


Figure 7.5: A real-time oscilloscope can be used to plot the behavior of any number of signals simultaneously. This is an invaluable when debugging a controller, especially when combined with hot-loading of control functions.

$$\begin{aligned}
 \frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} + \frac{\partial B_z}{\partial z} &= 0 \\
 \frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} + \frac{\partial E_z}{\partial z} &= \rho \\
 \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} &= -\dot{B}_x \\
 \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} &= -\dot{B}_y \\
 \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} &= -\dot{B}_z \\
 \frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z} &= j_x + \dot{E}_x \\
 \frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x} &= j_y + \dot{E}_y \\
 \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} &= j_z + \dot{E}_z
 \end{aligned}$$

With the invention of vector calculus, these cumbersome equations were rewritten in 1884 by Oliver Heaviside as a set of four vectorized equations, and is the form familiar to most engineers today.

$$\begin{aligned}
 \nabla \cdot B &= 0 \\
 \nabla \cdot E &= \rho \\
 \nabla \times E &= -\dot{B} \\
 \nabla \times B &= j + \dot{E}
 \end{aligned}$$

Physicists follow the pursuit of symmetry still further(?, ?), and can write the electromagnetic field tensor F with einstein summation and square brackets denoting anti-symmetric parts in just two equations:

$$\begin{aligned}\partial_{[\beta} F_{\beta\gamma]} &= 0 \\ \partial_\beta F^{\alpha\beta} &= \mu_0 J^\alpha\end{aligned}$$

Using the notation of geometric algebra, electromagnetic fields can be summarized as a single equation:

$$\left(\frac{1}{c} \frac{\partial}{\partial t} + \nabla \right) \mathbf{F} = \mu_0 c \mathbf{J}$$

Despite the differences in how they are written, each of these equation sets are equivalent. They all describe the same physical laws. Yet (Eqn 2laws) is simpler to write and reveals a better understanding of the underlying physical reality because it implicitly incorporates a symmetry – rotational invariance.

The analogy to computer programming should be obvious to anyone familiar with how a single line of high-level code can translate into tens or thousands of lines of low-level assembly instructions or bytecode. Although in the end the computer may perform the very same numerical operations when calculating the solution to these equations, the level of abstraction that the programmer chooses can have a great effect on how verbose and comprehensible the solution is. Studies have shown that programmer mistakes are correlated fairly well with the number of lines of code. It is therefore advantageous to use concise, elegant expressions whenever possible.

7.2 Backboard Compensation

The backboard forms an isosceles triangle 74cm at the base and 140cm on the upper sides. The angle of inclination of the backboard must be:

$$\theta_B = \frac{\pi}{2} - \cos^{-1} \left(\frac{37}{140} \right) \approx 0.267$$

Or about 15.3 degrees.

To compensate for the backboard tilt,

Markers were placed on the backboard at 1.00 m tall (1.002(3)m tall as measured by vicon) 1.70 m wide (1.706(5)m long as measured by vicon)

CHAPTER 8

JIMI: A Planar Hopper

This chapter presents the realization and performance of JIMI, a planar monopod hopping robot shown in Figure 8.1. It incorporates the centroidal momentum controller presented in Chapter 5 and uses fully asynchronous, concurrent dataflow software as developed in Chapter 7. JIMI is powered by two HypoSEA-v2 actuators (Chapter 4, has a monocoque structure comprised largely of a composite sandwich of carbon fiber and low density polyurethane foams (see Appendix ??), and stands approximately 120cm tall with the leg fully extended. The weight of the robot is 8.3kg in its minimalist test configuration, and 10.9kg when carrying lithium batteries and a controlling laptop. In the latter configuration, the robot is completely power autonomous.

The real-world performance of JIMI has not yet been fully measured. In preliminary motion experiments with some safety apparatus constraining its motion, it moves very quickly, has excellent force fidelity, appears to easily generate sufficient torques for it to leap into the air. Unfortunately, stable hopping controllers have not yet been tested in real world conditions; the robot was in fact just mechanically completed in February, 2012, and testing is still in progress. Simulations results indicate it is mechanically capable of jumping over obstacles at least 15cm in height (perhaps much more if the leg is actively retracted) and forward motion of at least 1.1m/s.

A collection of photos with captions is perhaps the fastest way to introduce the structure of the robot. Many photos of the robot are visible in Figures 8.1-8.7.

8.1 Design Details

Beyond the obvious structure of the robot, there are many details about JIMI that merit a closer look.

Saggital Plane Constraint

Although a beam of sufficient length that pierces the CoM seems to be a common way of approximately planarizing the motion of a robot (J. Pratt, 2000; Hurst, 2011; Chevallereau et al., 2003), such an approach requires a large amount of laboratory space. Another problem is that the planarization is not perfect, and typically the foot of the robot must slide radially somewhat as the CoM of the robot moves up and down. The worst problem is



Figure 8.1: Three views of the robot JIMI.

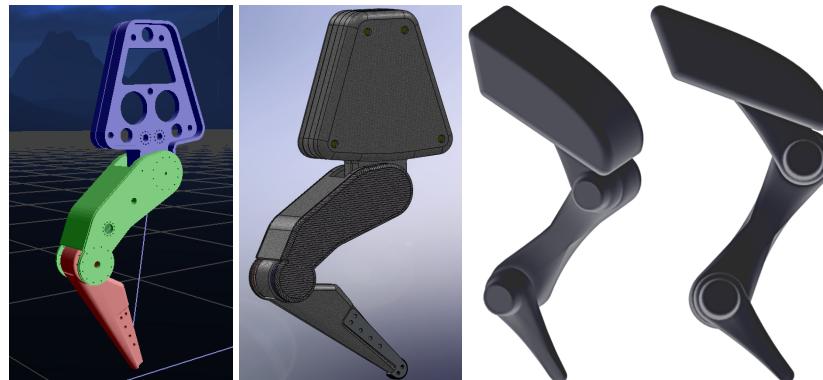


Figure 8.2: Left: An early draft of the JIMI robot during the first year of its design and simulation experiments. Right: The final composite monocoque CAD designs.



Figure 8.3: Left: Aluminum embedments provide a place for the HypoSEA actuators to mount and distribute stresses over the surface of the monocoque structure. Right: The HypoSEA-v2 before insertion into the thigh.



Figure 8.4: Left: The HypoSEA-v2s installed in the thigh. Right: Interior of the robot as viewed from the back during the construction process. It easily contains a laptop and lithium battery pack.

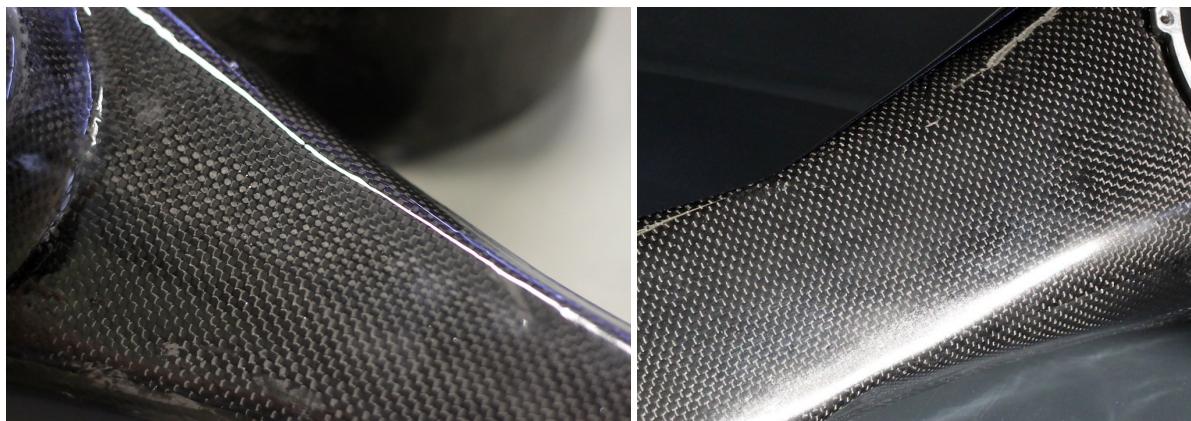


Figure 8.5: Photos of the beautiful carbon fiber weave texture visible when a molded part is placed in the sunlight.



Figure 8.6: Masses of the finished CFRP monocoque structures. Clockwise from upper left: 171g for the shank, 518g for the thigh, 84g for the thigh back cover, and 979g for the torso.



Figure 8.7: Details of JIMI. Clockwise from upper left: Cables pass through the axis of each joint and experience minimal stress. For testing, the robot may be connected to an external power source and controlled via ethernet. An emergency stop button allows it to be disabled immediately. The robot can also fold itself into a small configuration and weighs only 8.3kg.

that you add a significant amount of inertia to the robot at the connection point, disturbing the dynamics of the system.

It was originally intended that JIMI would have a follow-behind rickshaw to prevent it from falling left or right. Two long, lightweight carbon tubes will connect the robot to a pair of ultralight racing bicycle tires. If the tubes are long and lightweight enough the robot will not be significantly affected by the extra mass of the system, although some deviation from true planar dynamics is expected. The rickshaw would not need to be strong enough to support the weight of a human, and is more like a set of very lightweight training wheels for the robot. Furthermore, locking the axles of the bicycle wheels together would make the robot track perfectly straight.

In the end, due to space constraints it was simpler to use a sheet of plywood and a lightweight frame made of wood. The frame may be assembled or disassembled in a few minutes and can be placed above a 2m long treadmill for continuous running experiments. For steady state running, this closely approximates the real dynamics of planar motion.

State Measurement

Measuring the full state of a hopping robot like JIMI is a difficult task. During flight, onboard accelerometers cannot even tell what direction down is! Adding a complex state estimation system to make the robot fully self contained is an attractive idea, but much of the required engineering work for such a task is tangential to this thesis's central idea: harmonious dynamical design.

Therefore, a six-camera VICON system was used to estimate the position of reflective markers on the robot. Although the geometry of JIMI makes it difficult for all three rigid bodies to be detected by the VICON system at the same time, at least one rigid body can be tracked at all times. From the joint angles measured by the HypoSEA-v2's, the full state of the robot can be reconstructed from a single rigid body position.

The main downside of the VICON system seems to be a communication latency due to the use of TCP instead of UDP. Occasional latencies of 5-10ms are not uncommon, even when the VICON system is configured to broadcast at 500Hz. This is inconvenient, but it is not expected that it will preclude the possibility of fully autonomous hopping.

Heat Dissipation

Most of the heat in JIMI is generated in its body, where the laptop, battery, and motor controllers reside. The motor rotors also heat up under high holding torques. There has not been a mechanical problem due to heat so far, but as the HypoSEA-v2s are driven harder and harder it may be necessary to add more ventilation holes in the composite structure surrounding the motor stators. At present, the monocoque structure insulates the rotors disappointingly well and prevents them from cooling as fast as they otherwise might.

Onboard Computer

As a jumping robot will withstand repeated violent impact with the ground not just during normal operation but during failure modes such as falling, it would be obviously beneficial to have an onboard controller with no moving parts. Therefore, an Asus EEE PC with 12GB of SSD, and 8 hours of battery life was chosen. Not only is it convenient to be able to directly interact with the onboard controller of the robot, the laptop has no moving parts and when protected by the body of JIMI it should be very robust to the repeated impacts that a hopping robot experiences.

The electronics use only solid state components and are mounted on rubber standoffs.

Battery Operation

Fully power-autonomous operation has not yet been attempted. Typical energetic requirements for JIMI during nominal operation are estimated at between 100-300W continuous. If the continuous power demands are truly at this level, 1.5kg, 24V 18Ah lithium ion battery packs can be installed in JIMI for power autonomous operation. Even with such a high performance battery pack, it seems unlikely that JIMI will be able to function safely for more than about 45 minutes on a single charge.

If energy regeneration techniques (section 4.6.4) or motor braking effects were used, it may be possible to reduce the electrical energy consumption of JIMI further, but it seems unlikely that JIMI will be able to beat the ARL-II robot (Ahmadi & Buehler, 2006) and its very low specific cost of transport.

Adjustments

Even though a rigid body dynamics simulation was performed in Chapter 3, reality does not match simulation predictions precisely. To compensate for differences between model and reality, the following mechanical parameters may be adjusted with a relatively small amount of work:

1. The pretension of the HypoSEA springs (to adjust the nonlinearity).
2. The distribution of mass in the torso (by moving the battery or laptop).
3. The linear stiffness of the HypoSEA springs (by screwing a different spring in)
4. The length of the foot (by bolting a spare, longer foot on)

Each of these small adjustments affects the passive mechanical properties of JIMI. It is hoped that an efficient passive mechanical configuration can be found similar to ones studied in simulation.

Safety

Safety considerations for JIMI are fairly simple. The robot runs at low voltages and is not particularly electrically dangerous. Wiring is fused to prevent a short circuit from starting a fire. Wiring thicknesses are all above what is needed to carry the design currents.

Mechanically, the robot is somewhat more dangerous. Although the robot has no sharp corners with which to cut a human, the elasticity of the springs is a danger that must be considered – 20J of energy can be released very quickly from the series spring, and could potentially injure a human. Thankfully, the carbon fiber structure is lightweight and has little momentum, and presents less of a danger than a comparably sized aluminum structure would. It seems unlikely that the device could seriously wound someone, although a proper safety analysis would need to be done to say for certain.

Ironically, the robot is likely much less adept at protecting itself. Although very stiff along the direction of the fibers, the carbon fiber skin is less than a millimeter thick and can be crushed fairly easily during an impact with a sharp corner in the environment. For this reason, the rounded corners of the exoskeleton have been reinforced with extra layers of carbon fiber and are thicker than one millimeter. During testing, closed-cell foam rubber has also been taped to the outside corners to prevent damage if the robot falls on a hard floor.

There is also some concern that the robot could injure its own wiring at the ankle because the foot is unconstrained and may rotate continuously. If this occurs, the ankle will cut off its own power supply and stop functioning. It is presently only avoided through software checks, not through mechanical travel stops or limit switches.

Figure 8.8: Rubber skid pads on JIMI protect its carbon fiber surface from abrasion.

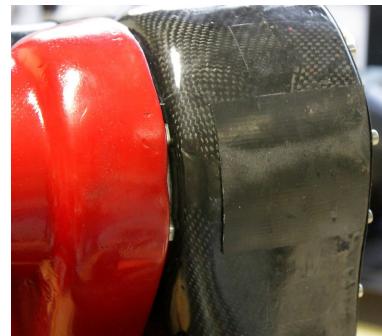


Figure 8.9: A soft rubber foot made of a squash ball was placed on the end of the foot to provide a high-friction surface.



Table 8.1: The predicted mass of JIMI according to initial estimates.

Mass	Desc
3150g	for motors (1050g x 3)
2600g	in stainless steel, bearings, and aluminum parts
1200g	for springs (396g x 3)
1100g	computer (Asus eeePC 1000H with internal battery)
970g	for a 24V 10Ah battery with low internal resistance
920g	for carbon fiber structure (Estimated surface area was 0.66m ² , and 4 layers of carbon + 1 layer of foam gives 1400g/m ²)
240g	motor drivers (80g x 3)

To catch the robot during hopping motions, a test harness, rope, and short length of shock-absorbent cord are used to suspend the robot's body and prevent it from falling in an uncontrolled manner.

8.2 Mass Budget

Original CAD Mass Budget

From CAD models and rough initial engineering estimates, the mass budget of JIMI was estimated at 10180g in total (8.1). At that point in time, the robot looked like Figure 8.2, left – substantially different its revision, which became the actual robot shown in Figure 8.1.

This information is included mostly for comparison purposes of the expected masses and the actual masses that were measured nearly two years after the original design.

Actual Masses

As mentioned in the introduction to this chapter, the total mass of the robot in its minimum configuration is 8.3kg. The hypocycloid mechanism weighs 1019g, the planetary gearing and differential weigh 828g, and the electrical motor with bearings and an encoder weighs 936g. The total mass of each HypoSEA-v2 was 2883g. The FRP structures weigh 1883g in total. Therefore, the remaining mass that is unaccounted for (650g) is a result of wiring, cables, motor drivers, standoffs, bolts, switches, corner padding, and other design details.

Masses of Composite Structure During Construction

To measure the masses of the internal, core, and external layers of the FRP composites, each of the parts was weighed multiple times during its construction. Table 8.2 shows the masses in grams for composite parts at various stages of development.

8.3 Performance Comparison

At this time, a performance comparison between JIMI and other robots cannot be made because JIMI has not been fully tested. Instead, we will merely list what robots could be used to provide perspective on the performance of JIMI; we will focus on a select few of the large number of single leg hopping robots that exist (Sayyad, Seth, & Seshu, 2007).

Table 8.2: Composite structure masses during construction.

Part	Outside FRP	+ Core	+ Inside FRP	Finished	Notes
Shank	79	108	175	171	Excess gel coat was sanded smooth, reducing finished mass.
Thigh	253	342	595	518	Metal embedments added 72g, but motors holes reduced mass by 150g.
Thigh cover	-	-	81	84	Gloss coat added 3g.
Torso body	-	486	712	-	Incomplete part, just for reference.
Torso attach	-	-	107	-	Incomplete part, just for reference.
Full Torso	-	-	-	979	Fiber glass tape to bond torso body and attachment added 55g. Red enamel spray paint added
15g					
Torso cover	-	-	126	131	Red paint added 5g.

The robot “Kenken” (Hyon & Mita, 2002) weighed 13.26kg and had a leg length range of 0.31-0.52m. The mass of Kenken’s leg was 3.6kg (thigh, shank, foot masses of 2.42, 0.75, 0.43kg). It moved forward at up to 1.5m/s, although it does not appear to have especially good speed regulation. It was not self contained, and used hydraulics for an improved power to weight ratio. Later this robot was duplicated and modified to create a biped robot, proving it was a somewhat extensible design (Hyon, Abe, & Emura, 2003; Hyon, Emura, & Mita, 2003).

The monopod robot Thumper (Poulakakis & Grizzle, 2009) (Figure 2.6, center) weighs approximately 30kg and stored ~75J of energy internally using high performance fiberglass springs (Hurst, 2008).

The Uniroo (Zeglin, 1991) weighed 6.6kg and had 1/3rd of its mass in its articulated leg. It was not power autonomous and used hydraulic actuation to store up to 60J of energy in a large spring at the ankle joint. The vertical motion of the CoM during its hopping motions appears to have been just less than 20cm, although this could probably have been increased.

The ARL Monopod (Rad, Gregorio, & Buehler, 1993) was impressive enough, but we will focus on its superior successor, the ARL Monopod II (Ahmadi & Buehler, 2006). It weighs 18kg, stands 0.7m tall, and is completely self-contained. At 1.25m/s forward speed it consumed just 20.7J/step; with 2.3hops/sec this is only 48W of mechanical power at the rotor, although the electrical energy consumption is not indicated. Latex springs were used at the hip joint, and steel springs in the telescoping leg. The maximum leg length was 0.64m. Interestingly, they experimentally found losses of 4-10J per step from impacts and dry friction. This is remarkably close to simulation results found by for JIMI.

A monopod hopping robot named CARL based on the LADD actuators weighed 8.3kg, had link lengths of 0.3m, and a 0.12m uncontrolled foot (Mennitto, 1995). Also similar to this thesis is that during the design of CARL, it was estimated that 30J of energy storage would be necessary per compliant joint. Instead of steel springs, they chose to use elastomers because the energy density is 150 times higher.

Most recently, a single leg design intended for a cheetah robot was presented (Lewis, Bunting, Salemi, & Hoffmann, 2011). The leg weighs 17-22N that is able to generate 90N, although this uses a mechanical advantage

and is not a very meaningful number – their goal was to create a hind limb capable of supporting 1.5 times the body weight of the robot (expected to be 100N). It also uses a pneumatic actuator and is not self contained or purely electric.

8.4 Engineering Improvements to JIMI

8.4.1 Mass Reduction

Although JIMI weighed 10.9kg fully power autonomous – an accomplishment in itself – further weight reduction would enhance its performance still further. There are several obvious areas of improvement:

1. With additional mechanical engineering work, lower-density materials, and careful stress analysis in particular, perhaps another 500g could be shaved off the existing HypoSEA-v2 design, reducing the mass of the robot by approximately one kilogram in total.
2. If a small PC104 computer were installed instead of a full-fledged 1.1kg laptop, another 800g of mass could be saved.
3. Due to time constraints, the torso of the robot was quickly prototyped using moldless hot-wire construction techniques, resulting in a mass of roughly a kilogram. If the torso of the robot were redesigned smaller to fit a PC104 computer and a molded construction process was used, the mass of the torso could be reduced by perhaps 500g.

Incorporating all three of these features could reduce the mass of the robot (sans batteries) to just 7.1kg, further enhancing the performance of the robot and increasing its payload carrying capacity. In fact, calculations show that if such a weight reduction were successful, JIMI could be made into a highly dynamic planar bipedal robot simply by mirroring the existing structure about the sagittal plane.

8.4.2 Revised Composite Structure

During the composite structure manufacturing process, it became obvious that compound curves dramatically increased the stiffness of a part with little change in the difficulty of mold construction. If JIMI were to be redesigned, using compound curves everywhere would further improve the strength and stiffness of the composite structures without adding any weight.

Although padded corners were used during testing phases, further increasing the thickness of the carbon fiber exterior at corner spots would improve impact robustness. This would be particularly advisable if inexperienced personnel are expected to test new balance controllers on the robot without supervision. At present, JIMI is a little like a race car: possessing high performance but being very sensitive to collisions.

8.4.3 Assembly and Manufacturing Costs

The time required to manufacture the aluminum and steel parts for the HypoSEA-v2 are unknown, as they were performed by a professional machine shop. The assembly time of the actuators is known to much better precision: it took approximately four hours to assemble each HypoSEA-v2, and with further practice this number could likely be reduced. This compares very favorably with the HypoSEA-v1, which took nearly 30 hours to assemble because of its complexity.

Building the carbon fiber parts was very labor-intensive, particularly the steps required to make good molds. Anything that would reduce the effort required to make a good mold would be a significant improvement to the prototyping construction process.

8.4.4 Electrical Improvements

The effort required to recalibrate the robot each time it was turned off and on was unnecessarily large. Using absolute encoders could reduce the effort required to calibrate the robot and would be a very good investment.

Sensing of the overall state of the robot without the VICON system would also be a fundamental improvement to the robot that would allow it to operate in more unstructured environments. Self-contained state estimation with gyroscopes, accelerometers, vision, or similar technologies would not add extensive weight and could greatly enhance the robot's real world practicality.

Other improvements to the electronics also present themselves. In particular, a new motor controller with better commutation scheme could reduce the extensive torque ripple experienced using the six-step commutation scheme for the trapezoidal BLDC motors.

Finally, adding electronics that could store regenerated energy could provide very interesting and unique capabilities, as was detailed in section 4.6.4.

CHAPTER 9

Conclusion and Future Work

Proper conclusions for this thesis are pending and should be available shortly. For now, please refer to the note in Chapter 6.

References

- Aghili, F. (1998). *Design and control of direct-drive systems with applications to robotics*. McGill University Montreal, Canada.
- Aghili, F., Buehler, M., & Hollerbach, J. (2002). Development of a high performance joint. *Int. Journal of Advanced Robotics*, 16(3), 233–250.
- Ahmadi, M., & Buehler, M. (1999). The ARL monopod II running robot: Control and energetics. In *Robotics and automation, 1999 IEEE international conference on* (Vol. 3, pp. 1689–1694).
- Ahmadi, M., & Buehler, M. (2006). Controlled passive dynamic running experiments with the ARL-monopod II. *Robotics, IEEE Transactions on*, 22(5), 974–986.
- Alexander, R. (1988). *Elastic mechanisms in animal movement*. Cambridge University Press.
- Alexander, R. (1990). Three uses for springs in legged locomotion. *The International Journal of Robotics Research*, 9(2), 53–61.
- Alexander, R. (1991). Energy-saving mechanisms in walking and running. *Journal of experimental biology*, 160(1), 55–69.
- Alexander, R. (2003). *Principles of animal locomotion*. Princeton University Press.
- Ankarali, M., Saranli, U., & Saranli, A. (2010). Control of underactuated planar hexapedal pronking through a dynamically embedded SLIP monopod. In *Robotics and automation, 2010. IEEE international conference on* (pp. 4721–4727).
- Babic, J., Lim, B., Omrcen, D., Lenarcic, J., & Park, F. (2009). A biarticulated robotic leg for jumping movements: theory and experiments. *Journal of mechanisms and robotics*, 1(1).
- Bicchi, A., & Tonietti, G. (2004). Fast and soft-arm tactics. *Robotics & Automation Magazine, IEEE*, 11(2), 22–33.
- Blaya, J., & Herr, H. (2004). Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 12(1), 24–31.
- Blickhan, R. (1989). The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12), 1217–1227.
- Breedveld, P. (1984). *Physical systems theory in terms of bond graphs*. Twente University.
- Breedveld, P. C. (2004). Port-based modeling of mechatronic systems. *Mathematics and Computers in Simulation*, 66(2–3), 99 – 128. Retrieved from <http://www.sciencedirect.com/science/article/pii/S037847540300199X> (<ce:title>Selected papers from the 4th IMACS Symposium on Mathematical Modelling</ce:title>) doi: 10.1016/j.matcom.2003.11.002
- Calinon, S., Guenter, F., & Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2), 286–298.
- Cavagna, G. (2006). The landing-take-off asymmetry in human running. *Journal of experimental biology*, 209(20), 4051–4060.
- Chapman, G., Cleese, J., Cleese, J., Idle, E., Jones, T., Palin, M., & Gilliam, T. (1971). “Ministry of Silly Walks” skit from the Montreux Special of “Monty Python’s Flying Circus”.
- Cheng, G., Hyon, S., Morimoto, J., Ude, A., Colvin, G., Scroggin, W., & Jacobsen, S. (2006). CB: A humanoid research platform for exploring neuroscience. In *Humanoid robots, 2006 6th IEEE-ras international conference on* (pp. 182–187).
- Chevallereau, C., Abba, G., Aoustin, Y., Plestan, F., Westervelt, E., Canudas-de Wit, C., & Grizzle, J. (2003). RABBIT: A testbed for advanced control theory. *Control Systems Magazine, IEEE*, 23(5), 57–79.
- Chevallereau, C., Westervelt, E., & Grizzle, J. (2005). Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. *The International Journal of Robotics Research*, 24(6), 431.

- Cho, B., & Oh, J. (2008). Running pattern generation of humanoid biped with a fixed point and its realization. In *Humanoid robots, 2008. humanoids 2008. 8th IEEE-ras international conference on* (pp. 299–305).
- Collins, S., Ruina, A., Tedrake, R., & Wisse, M. (2005). Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712), 1082.
- Costa, N., & Caldwell, D. (2006). Control of a biomimetic soft-actuated 10DOF lower body exoskeleton. In *Biomedical robotics and biomechatronics, 2006. the first IEEE/RAS-EMBS international conference on* (pp. 495–501).
- Curran, S., & Orin, D. (2008). Evolution of a jump in an articulated leg with series-elastic actuation. In *Robotics and automation, 2008. IEEE international conference on* (pp. 352–358).
- De Luca, A., Albu-Schaffer, A., Haddadin, S., & Hirzinger, G. (2006). Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *Intelligent robots and systems, 2006. IEEE/RSJ international conference on* (pp. 1623–1630).
- Eppinger, S., & Seering, W. (1987). Understanding bandwidth limitations in robot force control. In *Robotics and automation, 1987. IEEE international conference on* (Vol. 4, pp. 904–909).
- Ferris, D., Louie, M., & Farley, C. (1998). Running in the real world: adjusting leg stiffness for different surfaces. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1400), 989–994.
- Gerelli, O., Carloni, R., & Stramigioli, S. (2009). Port-based modeling and optimal control for a new very versatile energy efficient actuator. *IFAC International Federation of Automatic Control*.
- Geyer, H., Seyfarth, A., & Blickhan, R. (2006). Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings of the Royal Society B: Biological Sciences*, 273(1603), 2861.
- Goswami, A. (2008). Kinematic and dynamic analogies between planar biped robots and the reaction mass pendulum (RMP) model. In *Humanoid robots, 2008. 8th IEEE-RAS international conference on* (pp. 182–188).
- Goswami, A., & Kallem, V. (2004). Rate of change of angular momentum and balance maintenance of biped robots. In *Robotics and automation, 2004. IEEE international conference on* (Vol. 4, pp. 3785–3790).
- Griffin, T., Main, R., & Farley, C. (2004). Biomechanics of quadrupedal walking: how do four-legged animals achieve inverted pendulum-like movements? *Journal of experimental biology*, 207(20), 3545.
- Haiya, K., Komada, S., & Hirai, J. (2010). Tension control for tendon mechanisms by compensation of nonlinear spring characteristic equation error. In *Advanced motion control, 2010 11th IEEE international workshop on* (pp. 42–47).
- Hamill, D. (1993). Lumped equivalent circuits of magnetic components: the gyrator-capacitor approach. *Power Electronics, IEEE Transactions on*, 8(2), 97–103.
- Herr, H., & Popovic, M. (2008). Angular momentum in human walking. *Journal of Experimental Biology*, 211(4), 467–481.
- Hirose, M., & Ogawa, K. (2007). Honda humanoid robots development. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850), 11–19.
- Hobbelin, D., & Wisse, M. (2007). A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *Robotics, IEEE Transactions on*, 23(6), 1213–1224.
- Hofmann, A., Popovic, M., & Herr, H. (2009). Exploiting angular momentum to enhance bipedal center-of-mass control. In *Robotics and automation, 2009. IEEE international conference on* (pp. 4423–4429).
- Hogan, N. (1985). Impedance control: An approach to manipulation. *Journal of dynamic systems, measurement, and control*, 107(2), 17.
- Hollerbach, J., Hunter, I., & Ballantyne, J. (1992). A comparative analysis of actuator technologies for robotics. In *The robotics review 2* (pp. 299–342).

- Hollerbach, J., Hunter, I., Lang, J., Umans, S., Sepe, R., Vaaler, E., & Garabieta, I. (1993). The McGill/MIT direct drive motor project. In *Robotics and automation, 1993. IEEE international conference on* (pp. 611–617).
- Holoborodko, P. (2008). *Smooth noise robust differentiators*. <http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/>.
- Hosoda, K., Sakaguchi, Y., Takayama, H., & Takuma, T. (2010). Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure. *Autonomous Robots*, 28(3), 307–316.
- Hurst, J. (2008). *The role and implementation of compliance in legged locomotion*. ProQuest.
- Hurst, J. (2011). The electric cable differential leg: A novel design approach for walking and running. *Int. J. Human. Robot.*, 8(02), 301–321.
- Hurst, J., Rizzi, A., & Hobbelin, D. (2004). Series elastic actuation: Potential and pitfalls. *Carnegie Mellon University*.
- Hutter, M., Remy, C., Hopflinger, M., & Siegwart, R. (2010). SLIP running with an articulated robotic leg. In *Intelligent robots and systems, 2010. IEEE/RSJ international conference on* (pp. 4934–4939).
- Hutter, M., Remy, C., & Siegwart, R. (2009). Design of an articulated robotic leg with nonlinear series elastic actuation. In *International conference on climbing and walking robots clawar*.
- Hyodo, K., & Kobayashi, H. (1992). Kinematic and control issues on a tendon-controlled wrist mechanism. In *Proc. of international symposium on robotics, mechatronics and manufacturing systems* (pp. 67–72).
- Hyon, S., Abe, S., & Emura, T. (2003). Development of kenken II a biped running robot with tendon. *Nippon Robotto Gakkai Gakujutsu Koenkai Yokoshu (CD-ROM)*, 21, 1G14.
- Hyon, S., & Cheng, G. (2006). Passivity-based full-body force control for humanoids and application to dynamic balancing and locomotion. In *Intelligent robots and systems, 2006 IEEE/RSJ international conference on* (pp. 4915–4922).
- Hyon, S., & Emura, T. (2005). Symmetric walking control: Invariance and global stability. In *Robotics and automation, 2005. IEEE international conference on* (pp. 1443–1450).
- Hyon, S., Emura, T., & Mita, T. (2003). Dynamics-based control of a one-legged hopping robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 217(2), 83–98.
- Hyon, S., & Mita, T. (2002). Development of a biologically inspired hopping robot-kenken. In *Robotics and automation, 2002. IEEE international conference on* (Vol. 4, pp. 3984–3991).
- Hyon, S., Morimoto, J., Matsubara, T., Noda, T., & Kawato, M. (2011). XoR: Hybrid drive exoskeleton robot that can balance. In *Intelligent robots and systems, 2011. IEEE/RSJ international conference on* (pp. 3975–3981).
- Ishikawa, M., Neki, A., Imura, J., & Hara, S. (2003). Energy preserving control of a hopping robot based on hybrid port-controlled hamiltonian modeling. In *Control applications, 2003. cca 2003. IEEE conference on* (Vol. 2, pp. 1136–1141).
- Iwata, H., & Sugano, S. (2009). Design of human symbiotic robot TWENDY-ONE. In *Robotics and automation, 2009. IEEE international conference on* (pp. 580–586).
- Jafari, A., Tsagarakis, N., Vanderborght, B., & Caldwell, D. (2010). A novel actuator with adjustable stiffness (AwAS). In *Intelligent robots and systems, 2010. IEEE/RSJ international conference on* (pp. 4201–4206).
- Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Intelligent robots and systems, 2003. IEEE/RSJ international conference on* (Vol. 2, pp. 1644–1650).
- Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(Series D), 35–45.
- Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., & Akachi, K. (2008). Humanoid robot HRP-3. In *Intelligent robots and systems, 2008. IEEE/RSJ international conference on* (pp. 2471–2478).

- Karssen, J., & Wisse, M. (2011). Running with improved disturbance rejection by using non-linear leg springs. *The International Journal of Robotics Research*, 30(13), 1585–1595.
- Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of*, 3(1), 43–53.
- Khatib, O., Sentis, L., Park, J., & Warren, J. (2004). Whole body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(1), 29–43.
- Krupp, B., & Pratt, J. (2006). A power autonomous monopodal robot. In *Proceedings of SPIE-the international society for optical engineering* (Vol. 6201).
- Kubow, T., Schmitt, J., Holmes, P., Koditschek, D., et al. (2002). Quantifying dynamic stability and maneuverability in legged locomotion. *Integrative and comparative biology*, 42(1), 149–157.
- Kuo, A. (1995). An optimal control model for analyzing human postural balance. *Biomedical Engineering, IEEE Transactions on*, 42(1), 87–101.
- Laffranchi, M., Tsagarakis, N., & Caldwell, D. (2010). A variable physical damping actuator (VPDA) for compliant robotic joints. In *Robotics and automation, 2010. IEEE international conference on* (pp. 1668–1674).
- Laurin-Kovitz, K., Colgate, J., & Carnes, S. (1991). Design of components for programmable passive impedance. In *Robotics and automation, 1991. IEEE international conference on* (pp. 1476–1481).
- Lee, S., & Goswami, A. (2007). Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots. In *Robotics and automation, 2007 IEEE international conference on* (pp. 4667–4672).
- Lee, W., & Raibert, M. (1991). Control of hoof rolling in an articulated leg. In *Robotics and automation, 1991. IEEE international conference on* (pp. 1386–1391).
- Levine, W. (1996). *The control handbook*. CRC.
- Lewis, M., Bunting, M., Salemi, B., & Hoffmann, H. (2011). Toward ultra high speed locomotors: Design and test of a cheetah robot hind limb. In *Robotics and automation. 2011 IEEE international conference on* (pp. 1990–1996).
- Li, Z., Vanderborght, B., Tsagarakis, N., Colasanto, L., & Caldwell, D. (2012, May). Stabilization for the compliant humanoid robot coman: Exploiting intrinsic and controlled compliance. In *Ieee international conference on robotics and automation*. Minnesota, USA.
- MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press. Retrieved from <http://www.cambridge.org/0521642981> (Available from <http://www.inference.phy.cam.ac.uk/mackay/itila/>)
- Marieb, E., & Hoehn, K. (2007). *Human anatomy & physiology*. Pearson Education.
- McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research*, 9(2), 62–82.
- McMahon, T. (1984). *Muscles, reflexes, and locomotion*. Princeton University Press.
- Mennitto, G. (1995). *Compliant articulated robot leg with antagonistic LADD actuation*. Unpublished doctoral dissertation, McGill University.
- Mennitto, G., & Buehler, M. (1997). LADD transmissions: design, manufacture, and new compliance models. *Transactions of the American Society of Mechanical Engineers, Journal of Mechanical Design*, 119, 197–203.
- Metta, G., Sandini, G., Vernon, D., Natale, L., & Nori, F. (2008). The iCub humanoid robot: an open platform for research in embodied cognition. In *Proceedings of the 8th workshop on performance metrics for intelligent systems* (pp. 50–56).
- Mevey, R. (2006). *Sensorless field oriented control of brushless permanent magnet synchronous motors*. M.s. thesis, Kansas State University, Dept. of Electrical and Computer Engineering.
- Niiyama, R., Nagakubo, A., & Kuniyoshi, Y. (2007). Mowgli: A bipedal jumping and landing robot with an artificial musculoskeletal system. In *Robotics and automation, 2007 IEEE international conference on* (pp.

- 2546–2551).
- Orin, D., & Goswami, A. (2008). Centroidal momentum matrix of a humanoid robot: Structure and properties. In *Intelligent robots and systems, 2008. IEEE/RSJ international conference on* (pp. 653–659).
- Owaki, D., Koyama, M., Yamaguchi, S., Kubo, S., & Ishiguro, A. (2010). A two-dimensional passive dynamic running biped with knees. In *Robotics and automation, 2010. IEEE international conference on* (pp. 5237–5242).
- Palli, G., Melchiorri, C., Wimböck, T., Grebenstein, M., & Hirzinger, G. (2007). Variable stiffness control of antagonistic actuated joint. *Submitted to IEEE Trans. on Robotics*.
- Paluska, D., & Herr, H. (2006). The effect of series elasticity on actuator power and work output: Implications for robotic and prosthetic joint design. *Robotics and Autonomous Systems*, 54(8), 667–673.
- Park, H., Sreenath, K., Hurst, J., & Grizzle, J. (2011). Identification of a bipedal robot with a compliant drivetrain. *Control Systems, IEEE*, 31(2), 63–88.
- Paynter, H. (1961). *Analysis and design of engineering systems*. M.I.T. Press, Cambridge.
- Playter, R., & Raibert, M. (1992). Control of a biped somersault in 3D. In *Intelligent robots and systems, 1992. IEEE/RSJ international conference on* (Vol. 1, pp. 582–589).
- Poulakakis, I., & Grizzle, J. (2009). Modeling and control of the monopodal robot thumper. In *Robotics and automation, 2009. IEEE international conference on* (pp. 3327–3334).
- Pratt, G., & Williamson, M. (1995). Series elastic actuators. In *Intelligent robots and systems, 1995. IEEE/RSJ international conference on* (Vol. 1, pp. 399–406).
- Pratt, G., Williamson, M., Dillworth, P., Pratt, J., & Wright, A. (1997). Stiffness isn't everything. *Experimental Robotics IV*, 253–262.
- Pratt, J. (2000). *Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots* (Tech. Rep.). Ph.D Dissertation, MIT.
- Pratt, J., Chew, C., Torres, A., Dilworth, P., & Pratt, G. (2001). Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2), 129–143.
- Pratt, J., & Krupp, B. (2008). Design of a bipedal walking robot. *Proceedings of the 2008 SPIE*, 6962.
- Pratt, J., & Tedrake, R. (2006). Velocity-based stability margins for fast bipedal walking. *Fast Motions in Biomechanics and Robotics*, 299–324.
- Press, W., Flannery, B., Teukolsky, S., Vetterling, W., et al. (1986). *Numerical recipes* (Vol. 547). Cambridge University Press.
- Rad, H., Gregorio, P., & Buehler, M. (1993). Design, modeling and control of a hopping robot. In *Intelligent robots and systems, 1993. IEEE/RSJ international conference on* (Vol. 3, pp. 1778–1785).
- Raibert, M. (1985). *Legged robots that balance*. The MIT Press, Cambridge, MA.
- Raibert, M. (1986). Symmetry in running. *Science*, 231(4743), 1292.
- Raibert, M., Blankespoor, K., Nelson, G., Playter, R., et al. (2008). BigDog, the rough-terrain quadruped robot. In *Proceedings of the 17th world congress* (pp. 10823–10825).
- Raibert, M., & Hodgins, J. (1991). Animation of dynamic legged locomotion. In *ACM SIGGRAPH computer graphics* (Vol. 25, pp. 349–358).
- Rauch, H., Tung, F., & Striebel, C. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8), 1445–1450.
- Reynolds, W. W. (1977). Skeleton weight allometry in aquatic and terrestrial vertebrates. *Hydrobiologia*, 56, 35–37. Retrieved from <http://dx.doi.org/10.1007/BF00023283> (10.1007/BF00023283)
- Riese, S., & Seyfarth, A. (2012). Stance leg control: variation of leg parameters supports stable hopping. *Bioinspiration & Biomimetics*, 7, 016006.

- Ringrose, R. (1993). *Simulated creatures: Adapting control for variations in model or desired behavior*. Unpublished doctoral dissertation, Citeseer.
- Robilliard, J., & Wilson, A. (2005). Prediction of kinetics and kinematics of running animals using an analytical approximation to the planar spring-mass system. *Journal of experimental biology*, 208(23), 4377.
- Rodriguez, G. (1987). Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics. *Robotics and Automation, IEEE Journal of*, 3(6), 624–639.
- Rugh, W., & Shamma, J. (2000). Research on gain scheduling. *Automatica*, 36(10), 1401–1425.
- Ruina, A., Bertram, J., & Srinivasan, M. (2005). A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition. *Journal of theoretical biology*, 237(2), 170–192.
- Rummel, J., Iida, F., Smith, J., & Seyfarth, A. (2008). Enlarging regions of stable running with segmented legs. In *Robotics and automation, 2008. IEEE international conference on* (pp. 367–372).
- Rummel, J., & Seyfarth, A. (2008). Stable running with segmented legs. *The International Journal of Robotics Research*, 27(8), 919–934.
- Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. (2002). The intelligent ASIMO: System overview and integration. In *Intelligent robots and systems, 2002. IEEE/RSJ international conference on* (Vol. 3, pp. 2478–2483).
- Sakai, S., & Stramigioli, S. (2007). Port-hamiltonian approaches to motion generation for mechanical systems. In *Robotics and automation, 2007 IEEE international conference on* (pp. 1948–1953).
- Sayyad, A., Seth, B., & Seshu, P. (2007). Single-legged hopping robotics research – a review. *Robotica*, 25(05), 587–613.
- Schaft, A. (2006). Port-hamiltonian systems: an introductory survey. In *International congress of mathematicians, proceedings of*. European Mathematical Society Publishing House (EMS Ph).
- Schiavi, R., Grioli, G., Sen, S., & Bicchi, A. (2008). VSA-II: a novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. In *Robotics and automation, 2008. IEEE international conference on* (pp. 2171–2176).
- Semini, C. (2010). *HyQ—design and development of a hydraulically actuated quadruped robot*. Unpublished doctoral dissertation, Dissertation, Italian Institute of Technology and University of Genoa.
- Sentis, L. (2007). *Synthesis and control of whole-body behaviors in humanoid systems*. Unpublished doctoral dissertation, Stanford University.
- Shorten, M. (1993). The energetics of running and running shoes. *Journal of biomechanics*, 26, 41–51.
- Siciliano, B., & Khatib, O. (2008). *Springer handbook of robotics*. Springer-Verlag New York Inc.
- Stephens, B., & Atkeson, C. (2010). Dynamic balance force control for compliant humanoid robots. In *Intelligent robots and systems, 2010. IEEE/RSJ international conference on* (pp. 1248–1255).
- Stramigioli, S. (1998). *From differentiable manifolds to interactive robot control*. Unpublished doctoral dissertation, Technische Universiteit Delft.
- Sulzer, J., Roiz, R., Peshkin, M., & Patton, J. (2009). A highly backdrivable, lightweight knee actuator for investigating gait in stroke. *Robotics, IEEE Transactions on*, 25(3), 539–548.
- Tajima, R., Honda, D., & Suga, K. (2009). Fast running experiments involving a humanoid robot. In *Robotics and automation, 2009. IEEE international conference on* (pp. 1571–1576).
- Thoma, J., & Ould-Bouamama, B. (2000). *Modelling and simulation in thermal and chemical engineering: A bond graph approach*. Springer Verlag.
- Thorson, I., Svinin, M., Hosoe, S., Asano, F., & Taji, K. (2007). Quantifying gait robustness of passive dynamic robots. In *8th SICE system integration division annual conference, hiroshima, japan*. (pp. 265–266).

- Tonietti, G., & Bicchi, A. (2002). Adaptive simultaneous position and stiffness control for a soft robot arm. In *Intelligent robots and systems, 2002. IEEE/RSJ international conference on* (Vol. 2, pp. 1992–1997).
- Ugurlu, B., Tsagarakis, N., Spyros-Papastavridis, E., & Caldwell, D. (2011). Compliant joint modification and real-time dynamic walking implementation on bipedal robot cCub. In *Mechatronics (ICM), 2011 IEEE international conference on* (pp. 833–838).
- Vanderborght, B., Tsagarakis, N., Semini, C., Van Ham, R., & Caldwell, D. (2009). Maccepa 2.0: Adjustable compliant actuator with stiffening characteristic for energy efficient hopping. In *Robotics and automation, 2009. IEEE international conference on* (pp. 544–549).
- Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., & Lefeber, D. (2007). MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems*, 55(10), 761–768.
- Visser, L., Carloni, R., Unal, R., & Stramigioli, S. (2010). Modeling and design of energy efficient variable stiffness actuators. In *Robotics and automation, 2010. IEEE international conference on* (pp. 3273–3278).
- Wan, E., & Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive systems for signal processing, communications, and control symposium 2000. as-spcc. the IEEE 2000* (pp. 153–158).
- Wan, E., Van Der Merwe, R., & Nelson, A. (2000). Dual estimation and the unscented transformation. *Advances in Neural Information Processing Systems*, 12, 666–672.
- Welch, G., & Bishop, G. (2001). An introduction to the kalman filter. *Design*, 7(1), 1–16.
- Wellstead, P. (1979). *Introduction to physical system modelling*. Academic Press London.
- Wolf, S., & Hirzinger, G. (2008). A new variable stiffness design: Matching requirements of the next robot generation. In *Robotics and automation, 2008, IEEE international conference on* (pp. 1741–1746).
- Yamaguchi, J., & Takanishi, A. (1997). Development of a biped walking robot having antagonistic driven joints using nonlinear spring mechanism. In *Robotics and automation, 1997. IEEE international conference on* (Vol. 1, pp. 185–192).
- Zeglin, G. (1991). *Uniroo, a one legged dynamic hopping robot*. B.S. thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering.
- Zeglin, G. (1999). *The bow leg hopping robot*. Unpublished doctoral dissertation, Carnegie Mellon University.
- Zinn, M., Roth, B., Khatib, O., & Salisbury, J. (2004). A new actuation approach for human friendly robot design. *The international journal of robotics research*, 23(4-5), 379–398.

APPENDIX A

Bond Graphs

In this section, we will briefly review *systems theory* with respect to a convenient graphical notation developed by Henry Paynter called *Bond Graphs* (Paynter, 1961). A full treatment of bond graphs and systems theory is beyond the scope of this appendix, but the essentials may nevertheless be described in a few pages. The reader is encouraged to consult the bibliography for more detailed references (P Breedveld, 1984; Stramigioli, 1998). Although engineers may be hesitant to learn yet another specialized diagrammatic representation of a system, having already been required to learn to use block diagrams or circuit schematics, the benefits of bond graphs include the elegant visualization of underlying Hamiltonian mechanics and power flows of a system spanning multiple domains and may be worth the effort for the deep philosophical perspective it brings.

We begin our journey towards bond graphs with symmetry. Mathematics is most succinctly described as the search for equivalences – it is the explicit, symbolic formulation of underlying patterns, symmetries, and relations between areas of knowledge that may seem disparate to the untrained eye. The development of concise, broadly applicable mathematical abstractions which unify our ability to predict natural phenomena is perhaps the foremost goal of physicists and scientists in general. Although electrical systems, magnetic systems, physical motion, hydraulics, thermodynamics, and other branches of engineering may appear quite different at first glance, the underlying mathematics used to describe these domains has elegantly beautiful symmetries that cross domain boundaries. Particularly, the simplified physical descriptions of lumped parameter systems can be considered from a unified perspective based on the concepts of *energy* and *duality*.

Energy is a familiar quantity of enormous importance because it represents a common currency between systems in different domains and is a quantity that must be conserved in general. It's second nature for most people to grasp how a rapidly spinning tire or baseball has some (kinetic) energy, or to discuss how much (potential) energy might be stored in an electrical battery or pressurized hydraulic tank. Many scientists are intuitively familiar with how an electric motor converts electrical power into mechanical power, how “lost” energy typically manifests itself as heat, and how energy in one domain may be shifted into another domain and then back again. In a complex system, everything is connected by the flow of energy. Interaction, by definition, implies an exchange of energy.

Figure A.1 shows an example of how a RLC circuit might be redrawn as a bond graph, which is a labeled,

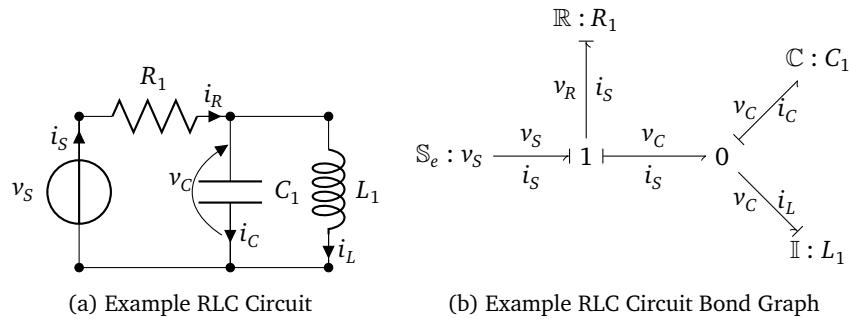


Figure A.1: Resistor-Inductor-Capacitor (RLC) circuit and its equivalent bond graph. Note how the series connectivity of the voltage source and resistor are represented using a 1-junction and the parallel connectivity of the inductor and capacitor are represented with a 0-junction.

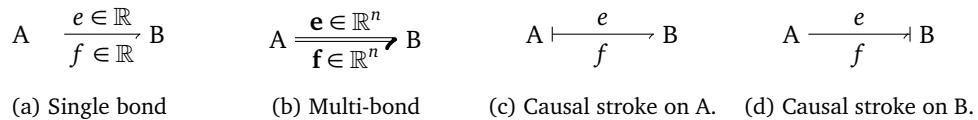


Figure A.2: Various ways to draw a bond that defines positive power flow as from component A to B.

directed graph made of vertices and edges. Each vertex represents a discrete component – or perhaps more accurately, some group of components about which it is convenient for us to draw a boundary – which exchanges energy with other components via the edges or *bonds* connected to it. Bonds look like little harpoons (Figure A.2a) which point in the direction considered by us to be a positive power flow. Yet despite their directional appearance, bonds represent *bidirectional* energy flows, which is why such a sign convention is useful to denote the direction of positive power flow. Bonds are not specific to a particular energy domain; mechanical, electrical, hydraulic, and chemical energy flows are routinely and seamlessly written on the same bond graph. The functionality of any system may be neatly encapsulated by considering only the bonds through which it connects to another system, and indeed this is precisely what we get in the the *Port Hamiltonian System* (PHS) formulation. The *ports* of a component are its interface to the rest of the world.

Each bond has two variables associated with it: *effort* (e) and *flow* (f). The harpoon-hook of each bond typically bends in the direction of the flow variable so that it need not be redundantly labeled as such. The instantaneous energy flow (power P) of each bond is equal to the product of these two variables ($P = e \cdot f$), and for this reason they are called *power conjugate variables*. Examples of power conjugate variables in different domains include voltage and current, velocity and force, or fluid pressure and flow. A complex system may benefit from having multiple effort and flow quantities bound into vectors so that multiple dimensions can be drawn with a single so-called *multi-bond* (Figure A.2b), but for now we only need consider bonds with a single effort and flow variable.

Besides representing the flow of energy, a bond graph can also be used to express the computational structure of a system for the purposes of simulation or control. In such cases they are called *causal bond graphs* because each bond has a causality stroke on the end of the bond whose component is responsible for setting the value of the flow variable (Figures A.2c and A.2d). In other words, it establishes whether the component treats the effort or flow as the independent variable. Such a convention also emphasizes the fact that each component may be dominate the control of either the effort or flow variable, but not both. This requirement is not just for computational convenience; it also expresses a physical necessity of equilibrium. It would not make much sense

if a component could set both the effort and flow variables simultaneously, since that would in principle mean than an infinite amount of power could be pulled or pushed through the bond regardless of what is attached to the other side. Pulling an infinite amount of energy out of a component obviously violates the conservation of energy. Smooth, physically plausible and incrementally smooth interaction requires that both components together must cooperate in some sense to reach an equilibrium regulating the flow of energy.¹

Five fundamental types of components need be discussed to understand bond graphs:

1. *Supply elements* (\mathbb{S}) which specify an effort or flow to determine system boundary conditions.
2. *Dissipative elements* (\mathbb{R}) from which energy cannot be recovered.
3. *Energy storage elements*, of which there are two types:
 - Generalized potential energy stores (\mathbb{C}).
 - Generalized kinetic energy stores (\mathbb{I}).
4. *Distribution elements* which describe topological constraints or intra-domain connections:
 - **0-junctions**, which constrain every connected bond to have *equal efforts*.
 - **1-junctions**, which constrain every connected bond to have *equal flows*.
5. *Conversion elements* which represent configuration constraints or inter-domain connections:
 - *Transformers* ($\mathbb{T}\mathbb{F}$) which alter the ratio of effort to flow but preserve total power.
 - *Gyrators* ($\mathbb{G}\mathbb{Y}$) which convert efforts into flows, convert flows into efforts, and can scale both quantities appropriately under the constraint that power is conserved.

We will explain each of the above in more detail in the order just presented, but we must first discuss the general model of a component and its causality before proceeding further.

A.1 Constitutive Relations and Causality

If we consider a component with only a single power bond, its interaction with the world is completely describable in terms of the effort and flow through that bond. Importantly, the mapping between the effort and flow variables at every operational point can be characterized by a single scalar variable expressing the ratio between effort and flow. This scalar is usually a familiar quantity such as resistance, mass, inductance, stiffness, etc. The equations determining the instantaneous value of this scalar quantity is the topic of this section.

For each component, the unique mapping which determines the relation between effort and flow is called the *constitutive relation* Ψ . In general Ψ is nonlinear (Figure A.3a) but linear components combine to form systems of such wonderful mathematical tractability that we often treat things as approximately linear (Figure A.3b). For linear components, the partials of Ψ are constant values, which greatly simplifies the analysis of the system.

An important question to ask when calculating the value of the scalar linking effort and flow is, “Is the effort a function of the flow, or is the flow a function of the effort?” Mathematically, should we say $f(e) = \Psi(e)$ or $e(f) = \Psi(f)$? Is e or f the independent variable? Is the flow caused by the effort, or the effort caused by the flow?

¹The author asks you to imagine prodding a ball with your fingertip on a flat surface. If you push hard, the ball accelerates quickly. If you push gently, it accelerates slowly. You cannot push hard on the ball and also insist that it accelerate slowly because you cannot control both effort and flow variables – the ball’s inertia itself helps to determine the equilibrium.

These are in fact trick questions in some sense; a more precise answer is that the laws of physics and its mathematical equalities considered here are acausal. Effort does not cause flow or vice versa. There is never an independent variable. All physics says is that effort and flow must at every instant be in an equilibrium which conserves total system energy. In this respect the equilibrium of the universe resembles a geometric shape, and the dynamical evolution a system traces out some curve on the surface of a possibly contorted but locally smooth shape. We get the same curve if we trace it in one direction or the other; physics is self-consistent regardless of whether time is moving forward or backward. This is because our mathematical descriptions of systems at the microscopic level are time-symmetric, and no experiment we can perform has a causal distinction where we can declare that one action causes another reaction.²

Our notion of causality must therefore be a purely human concept. The laws of physical reality merely say that power conjugate variables must remain in instantaneous equilibrium with each other at all times, however difficult this simple interdependency may be to handle with our human mathematical frameworks. In the end, we must never forget that the decision of causality is really for ease of mathematical calculation and has no physical basis.

We need not abandon the distinction entirely, however. Causality is still a useful concept because it is often easier for human mathematicians to perform our limited analytic techniques on systems which express state based only upon what we consider “past” information. Some components are much easier to consider as being effort-causal integrators, while others are simpler to treat as flow-causal. The addition of causal marks to our bonds (Figure A.2c and A.2d) are merely to help us create simpler computational models of the system.

To see why, let us look again at Figure A.3a and assume that Ψ is defined as a function of e . In such a case, given any e we may immediately find the value of f . But for other components the mathematics might be more elegantly written if we find f using the area α or even β . These are both possible given some knowledge of integral and differential calculus, but the mathematics can be easier to do in one form or another. We will revisit this topic in section A.4 to see why this is relevant.

A.2 Effort and Flow Sources (\mathbb{S})

The simplest components to describe do not even have a constitutive function – they simply specify the value of an effort or flow and leave the conjugate variable free to take on any value. Electrical engineers often use the concept of perfect voltage or current sources which are capable of maintaining a perfect voltage or current regardless of the connected load, but analogies exist in the mechanical domain as well in the form of applied free forces or connections to fixed objects (infinitely large masses).

Sources come in two types:

Effort Sources \mathbb{S}_e , which specify effort exactly, leave the flow variable free to take on any value, and which *never* have the causal stroke on their side.

Flow Sources \mathbb{S}_f , which specify flow exactly, leave the effort variable free to take on any value, and which *always* have the causal stroke on their side.

Sources are most useful for providing boundary conditions applied to the edges of a system or for simplifying a system in which some dynamics are negligible compared to others. Naturally, because sources may specify an effort or flow independently of their dual variable, they are necessarily active components and are often used to

²Our present understanding of the unidirectional nature of thermodynamics at the macroscopic level appear to contradict this statement, but perhaps physicists of the future will somehow treat the information content of the universe as the dual of entropy and resolve this apparent asymmetry.

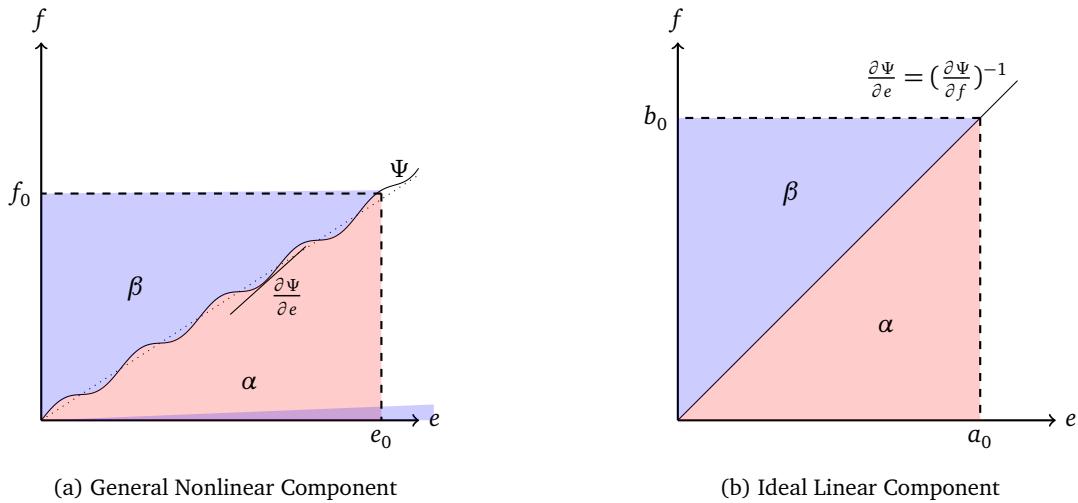


Figure A.3: General models of linear and nonlinear components in which effort and flow are related by function Ψ .

specify a constraint as a function of time or system state. An example might be a voltage applied to a motor with powerful motor driver electronics whose dynamics do not need to be considered.

A.3 Dissipative Components (\mathbb{R})

Another simple component that we now consider are *memoryless components* such as idealized electrical resistors or other elements which dissipate energy in an irrecoverable fashion.³ Examples of dissipators in other domains include viscous friction, dashpots, magnetic eddy currents, or hydraulic restrictors. However, since resistors are easy to understand, we will focus our examples on them.

If increasingly large efforts (voltages) are applied across a resistor, increasingly large currents (flow) will result; the history of voltages applied to the resistor is irrelevant to its immediate function. Of course, for real world resistors, the relation between voltage and current may not be perfectly linear, and at some high voltage the resistor may break down and its behavior will change significantly (Figure A.4a).

When solving for the voltage across or current through an ideal resistor, the issue of causality is not particularly relevant; the mathematics of dissipative element causality are related by a static constitutive function $\Psi_{\mathbb{R}}$ that we may choose with impunity to be a function of e or f . If we have an ideal current source and the resistor shown in Figure A.4a, we can quickly compute the voltage across the resistor from $\Psi_{\mathbb{R}}^{-1}$. Given an ideal voltage source, the current flow through the resistor is also easily found from $\Psi_{\mathbb{R}}$. In either case, the instantaneous power dissipated by the resistor is the same: $P = e \cdot f = f \cdot e$.

The scalar value of $\Psi_{\mathbb{R}}$ which relates e and f is often given two names – resistance R or conductance R^{-1} in the case of resistors – which are two sides of the same scalar but yield more convenient numerical values in different circumstances. For a linear resistor, we usually prefer to pretend that $R = \frac{\partial \Psi_{\mathbb{R}}}{\partial f}$ is constant so that e and f are related linearly (Figure A.4b).

³Technically, dissipators merely convert one form energy into heat and could be modeled as transformers, but for most modeling situations energy sent to the thermal domain is lost forever and it is simpler just to treat dissipators as energetically negative definite.

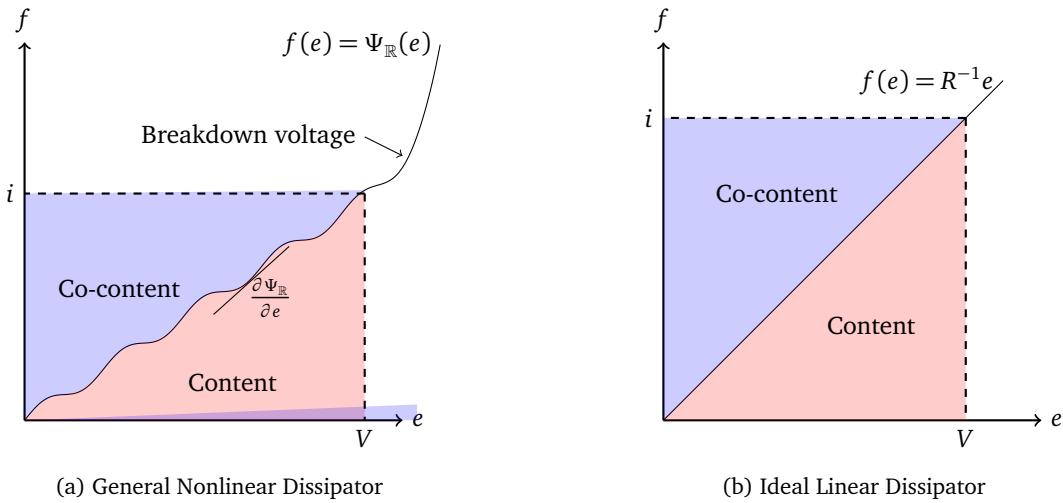


Figure A.4: Models of dissipative elements with conductive causality. Resistive causality is the same, but drawn with the axes swapped and $e = \Psi_{\mathbb{R}}(f)$.

A.4 Conservative Components (\mathbb{I} and \mathbb{C})

What does it mean for a component to have state? Components with state have some memory of their past; they have some extra variable needed to describe their configuration or dynamics. A spinning mass must have a known angular momentum to describe its gyroscopic effects; a capacitor needs to have the charge separation specified to predict the current that will flow into it; we need to know how much a spring is deflected to predict its force. The state of each of these components is closely related with the energy associated with them, as we shall see in this section.

Let us start with a thought experiment. If we measured the relation between effort and flow variables for a component three times and got three different curves as shown in Figure A.5a, what would this suggest about constitutive function Ψ ?

The author hopes that you would conclude that multiple curves signals that Ψ must really be a function of a hidden variable reflecting the accumulated *state* or *memory* of the past. To persuade you of this possibility, imagine that Figure A.5a represents a test of a non-ideal resistor which was characterized by gradually turning up a current supply and measuring the voltage across the resistor. Most electrical engineers know that a linear resistor dissipates electricity as heat with a magnitude proportional to current squared ($P_{\mathbb{R}} = R_i^2$). Therefore during the first test, some energy must have been lost as heat that would accumulate in the resistor if it were well-insulated. The resistance of most materials also increases with temperature; therefore on the second and third tests the resistor's resistance would be increased, and the effort/flow relation would be different. The hidden variable might therefore be *temperature* in this case.

If we now measured the value of the resistance R for a variety of temperatures, we might once again discover a simple, static and possibly nonlinear relation as shown in Figure A.5b. This curve is essentially the constitutive relation of the resistor's material as it heats; it expresses the ratio between effort-flow ratio for different temperatures. From this viewpoint, the resistor is essentially a variable transformer of electrical energy into heat energy.

Energy lost to heat a resistor may have the effect of changing the scalar value of resistance relating e and f , but it is generally irrecoverably lost from an energetic standpoint – the energy has flowed uni-directionally

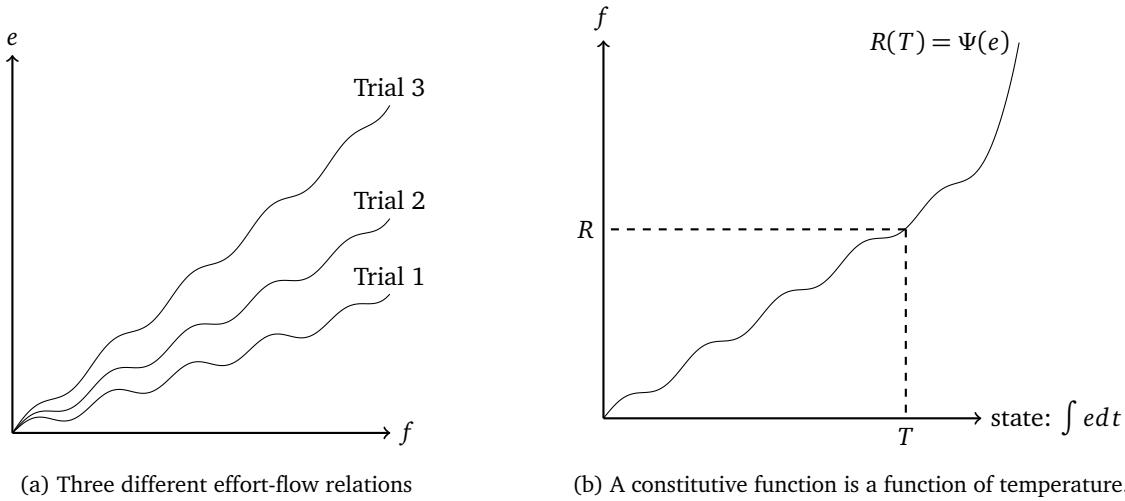


Figure A.5: A non-ideal resistor whose resistance varies with temperature.

from electrical energy into heat energy. However, most domains have components which store energy in a manner that is *immediately and necessarily recoverable*, an important and crucial fact which establishes why some components have dynamics. As a thought experiment, imagine how we might model a hypothetical resistor which actually could convert its stored heat into electrical energy. We might try treating it as a voltage (effort) store, as something that accumulated heat energy based on the voltage, and released its heat energy as a current whenever the voltage started to dip. Alternatively, we might have treated it as a current (flow) store and seen the inverse effect. In either case, power can flow both in and out of the component depending on its temperature and the power conjugate variables.

In reality, what we find is closely analogous to this thought experiment. In each domain, we may immediately distinguish two types of components based on how they store and release their energy:

I domain components which store *generalized kinetic energy*. They are also known as *effort stores* because their state p is the accumulation of the effort variable. Under integral causality, these components may be called *admittances*.

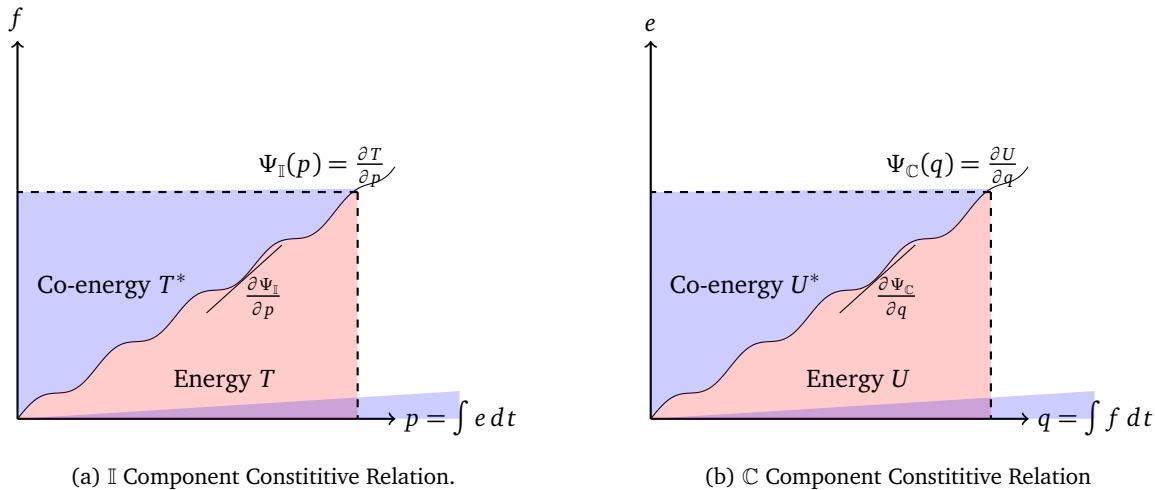
C domain components which store *generalized potential energy*. They are also known as *flow stores* because their state q is the accumulation of the flow variable. Under integral causality, these components may be called *impedances*.

Both types of components are energetically conservative and no energy is lost when being stored or released. Their constitutive functions Ψ_{I} and Ψ_{C} are not functions of e or f as was the case for resistors, but are instead functions of accumulated effort $p = \int e dt$ and accumulated flow $q = \int f dt$, as shown in Figure A.6. The expressions $e(q) = \Psi_{\text{C}}(q)$ and $f(p) = \Psi_{\text{I}}(p)$ give the so-called *integral causal form* of the dynamics because they are functions of p and q .⁴ Conversely, the equivalent *differential causal form* of their dynamics expresses the effort and flow as the rates of change of the states: $e = \frac{dp}{dt}$ and $f = \frac{dq}{dt}$. Stored energy is also expressed in terms of p and q : the kinetic energy stored in I components is $T = \int \mathbf{f} \cdot d\mathbf{p}$, and the potential energy stored in C components is $U = \int e dq$.

⁴The reader is encouraged to think about how Ψ could be expressed as function of α and β as was discussed in section A.1. Hint: α and β are related by the Legendre transform.

Table A.1: Summary of \mathbb{I}, \mathbb{C} component equations

	\mathbb{I}	\mathbb{C}
State	$\mathbf{p} = \int \mathbf{e} dt$	$\mathbf{q} = \int \mathbf{f} dt$
Rate	$\dot{\mathbf{p}} = \mathbf{e}$	$\dot{\mathbf{q}} = \mathbf{f}$
Constitutive	$\mathbf{f} = \Psi_{\mathbb{I}}(\mathbf{p})$	$\mathbf{e} = \Psi_{\mathbb{C}}(\mathbf{q})$
Energy	$T = \int \mathbf{f} \cdot d\mathbf{p}$	$U = \int \mathbf{e} \cdot d\mathbf{q}$
Co-energy	$T^* = \int \mathbf{p} \cdot d\mathbf{f}$	$U^* = \int \mathbf{q} \cdot d\mathbf{e}$
Power	$\dot{T} = \mathbf{p} \cdot \frac{\partial T}{\partial \mathbf{p}} \Big _{\mathbf{p}=\mathbf{p}_0}$	$\dot{U} = \mathbf{q} \cdot \frac{\partial U}{\partial \mathbf{q}} \Big _{\mathbf{q}=\mathbf{q}_0}$

Figure A.6: Constitutive relations for \mathbb{I} and \mathbb{C} components.

Most amazingly, because the components are energetically conservative, the implication of these equations is that the Ψ for each \mathbb{I} or \mathbb{C} component is exactly the same thing as the partial derivatives of the energy stored in p or q . That is, if the total system energy is $H = T + U$, then the constitutive functions for each component are $[\Psi_0, \Psi_1, \dots]^T = \frac{\partial H}{\partial \mathbf{x}}$ where $\mathbf{x} = [p_0, p_1, \dots, q_0, q_1, \dots]^T$ is the vector of component states.

The dynamical equations of components with state are thus summarized in table A.1.

Perhaps the most succinct and intuitive description of the relationship between $\mathbb{R}, \mathbb{I}, \mathbb{C}$, components is found graphically in Figure A.7, which expresses the equations of equilibrium or underlying geometric structure of a Hamiltonian system, as it expresses the system of equations that must be satisfied at all instants. The figure expresses the equations as a graph in which vertices represent quantities and edges represent function applications. When solving the system mathematically, it is left up to us to choose which of the arrows connecting the nodes we will use. To repeat the earlier mantra: physics has no causality, but our human mathematics does. When we solve for the effort and flow variables for some components in terms of values p and q , we are writing the math for that component in *integral causality* form because p and q are time-integrated values of e and f . The alternative is known as *differential causality* and determines e and f from the time derivatives of p or q . Although the two formulations are mathematically equivalent in some sense, integral causality is preferred for simulations and state space control because the resulting equations are ordinary differential equations (ODE) instead of differential algebraic equations (DAE), and the former is typically easier to solve. In fact, using integral causality

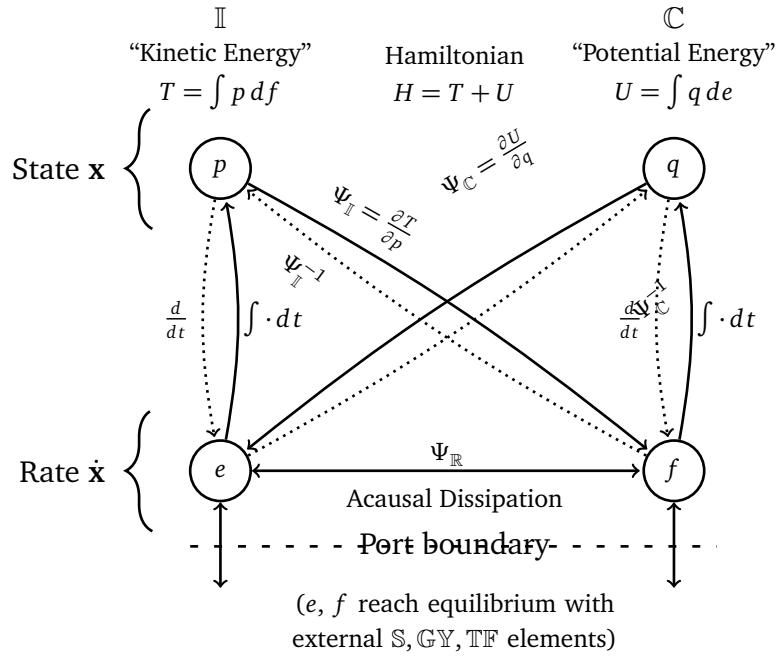


Figure A.7: The mathematical structure of \mathbb{R} , \mathbb{I} , and \mathbb{C} components. This graph describes an equilibrium which must be satisfied at all times. Vertices represent various quantities and directed edges represent the function application needed to reach equality with the destination vertex. When solving for the equilibrium using integral causality, solid lines are used. Conversely, for differential causality, the dotted lines are used. States p and q hold the system kinetic and potential energies (T, U).

and bond graphs provides a way to almost directly write the Hamiltonian equations describing the dynamics of the system, as will be shown in section A.10. For this reason, it is preferable to use integral causality whenever possible, and only use differential causality when there are algebraic loops or constraints that need to be handled.

Besides the topic of integral or differential causality, there is another terminology about causality worth mentioning. If a system supplies an effort and obtains a flow, it is said to have *impedance causality* at that port. Conversely, if we supply a flow and obtain an effort, we call this *admittance causality*. If we use integral causality exclusively, the word “admittance” becomes synonymous with “ \mathbb{I} domain components”, and “impedance” can be likewise reserved to mean “ \mathbb{C} domain components”, rather than use such valuable words to refer to the distinction of independent variable for a function that is relevant only at time of computation time.

To summarize the crucial points made for \mathbb{I} and \mathbb{C} components: (1) what we once called α, β in figure (A.3) are now called p and q and represent the accumulated effort or flow variables; (2) Ψ is now expressed as a function of p or q , which represent the *state* or *stored energy* of a component; (3) $\Psi(p)$ is the same thing as the partial of the Hamiltonian $\frac{\partial H}{\partial p}$; (4) the *rate of change* of the p state is the same quantity as the effort, and the rate of change of q state is the same as the flow; (5) we prefer to do our calculations using *integral causality* because it leads directly to state-space equations.

For reference, a complete list of power conjugate variables and common components for a variety of domains is presented in table A.2. The advantage of such a table is that the details and properties of effort e , flow f , states p and q , and the three types components ($\mathbb{R}, \mathbb{C}, \mathbb{I}$) clearly show symmetries across domains. Newton’s second law, Ohm’s law, Fourier’s law, Hooke’s law and many other classic laws appear in countless locations across the table. For all domains, common linear models of components are considered as well as the units of the domain. Notice also that the product of the power conjugate variables in every domain has units of power: W, J/s, kg·m²/s, N·m/s, or Pa·m³/s. The magnetic domain is incomplete because in reality the electric and magnetic fields are the same phenomenon and could be completely modeled from the electric perspective. Still, it is occasionally useful to consider the magnetic domain explicitly, and for this reason it can be included. Finally, although the chemical

domain might also be easily added to the table, it is not particularly relevant to most robotics and was therefore omitted so that the table would fit on a single page.

A.5 Connectivity (0- and 1-junctions)

Connections between elements in bond graphs are expressed with multi-bond *junctions*:⁵:

0-junctions which express *effort equality constraints* of every connected bond: $e_1 = e_2 = \dots = e_n$.

1-junctions which express *flow equality constraints* of every connected bond: $f_1 = f_2 = \dots = f_n$.

If the laws of power conservation is to be maintained, the above equality implies that the values of the unconstrained power conjugate variable at that junction must sum to zero. Therefore,

0-junctions also express that the flows of all connected bonds must sum to zero: $\sum_{k=i}^n f_k = 0$.

1-junctions also express that the efforts of all connected bonds must sum to zero: $\sum_{k=i}^n e_k = 0$.

These two junctions express many familiar constraints that will become familiar through examples. In the electrical domain, a 0-junction expresses both the concepts of parallel connectivity and Kirchoff's current law for a node. A 1-junction expresses series connectivity and Kirchoff's voltage law for a loop. In the mechanical domain, a 0-junction expresses Newton's third law of equal magnitude of applied and reaction forces at a kinematic constraint, or expresses connections to a deflecting elastic element or spring. Conversely, the 1-junction expresses connections to a mass with distinct velocity, and that the net forces on that mass must sum to zero by Newton's second law. In the hydraulic domain, 0-junctions express conservation of volume flow rate at a junction, and 1-junctions express how pressure drops around a loop sum to zero. In the thermal domain, 0-junctions indicate areas of the same temperature and 1-junctions are used to create temperature differences.

If the mathematical equations are to be resolved most simply, 0-junctions should have only bond specifying effort, and 1-junctions should have only one junction specifying flow. In such cases, the unique bond is called the *strong bond* and the junction will allow integral causality.

A.6 Gyrators ($\mathbb{G}\mathbb{Y}$) and Transformers ($\mathbb{T}\mathbb{F}$)

Gyrators and transformers are two-port components which express a power connection between two components in possibly different domains. Mathematically, they are very simple to explain. If the *effort* in one component is related to the *effort* in the other by a scalar K , then we call the component a *transformer*.

$$e_1 = Ke_2 \tag{A.1}$$

$$f_1 = K^{-1}f_2 \tag{A.2}$$

Alternatively, if the *effort* in the one component is related to the *flow* in the other by a scalar K , we call the component a *gyrator*:

⁵In the author's opinion, these junctions are rather poorly named because 0 and 1 suggest very little about their functionality to the novice. Perhaps it would have been preferable to name them as \mathbb{E} junctions and \mathbb{F} junctions to suggest their equality. Unfortunately, \mathbb{E} and \mathbb{F} are used by some authors to denote effort flow sources already. Room for improvement in the nomenclature clearly exists.

	Signal/Component	Symbol	Units	Constitutive	Differential	Integral	Energy
GENERAL	e Effort	e	-	$e = \Psi_{\mathbb{I}}(p)$	$e = \dot{p}$	-	-
	f Flow	f	-	$f = \Psi_{\mathbb{C}}(q)$	$f = \dot{q}$	-	-
	p State	p	-	$p = \Psi_{\mathbb{I}}^{-1}(e)$	-	$p = \int e dt$	$T = \int f dp$
	q Co-state	q	-	$q = \Psi_{\mathbb{C}}^{-1}(f)$	-	$q = \int f dt$	$U = \int e dq$
	\mathbb{I} Linear Admittance	$K_{\mathbb{I}}$	-	$K_{\mathbb{I}} = \frac{\partial T}{\partial p} =$	$e = \dot{p} = K_{\mathbb{I}} \dot{f}$	$f = K_{\mathbb{I}} p$	$T = \frac{1}{2K_{\mathbb{I}}} p^2$
	\mathbb{C} Linear Impedance	$K_{\mathbb{C}}$	-	$K_{\mathbb{C}} = \frac{\partial U}{\partial q}$	$f = \dot{q} = K_{\mathbb{C}} \dot{e}$	$e = K_{\mathbb{C}} q$	$U = \frac{1}{2K_{\mathbb{C}}} q^2$
	\mathbb{R} Linear Dissipator	$K_{\mathbb{R}}$	-	$K_{\mathbb{R}} = \frac{e}{f}$	-	-	-
ELECTRIC	e Voltage	v	V				
	f Current	i	A				
	p Flux Linkage	λ	V·s				
	q Charge	Q	C				
	\mathbb{I} Inductor	L	H		$\lambda = L i$	$v = \dot{\lambda} = L \dot{i}$	$i = \frac{\lambda}{L}$
	\mathbb{C} Capacitor	C	F		$Q = Cv$	$i = \dot{Q} = C \dot{v}$	$v = \frac{Q}{C}$
	\mathbb{R} Resistor	R	Ω		$v = iR$	-	-
MAGNETIC	e Magnetomotive Force	M	A				
	f Magnetic Flux Rate	$\dot{\Phi}$	V				
	p N/A*	-	-				
	q Magnetic Flux	Φ	Wb				
	\mathbb{I} N/A*	-	-				
	\mathbb{C} Magnetic Permeance	P	H		$\Phi = PM$	$\dot{\Phi} = P \dot{M}$	$M = \frac{\Phi}{P}$
	\mathbb{R} Induced eddy losses*	R_M	Ω^{-1}		$M = \dot{\Phi} R_M$	-	-
TRANSLATION	e Force	F	N				
	f Velocity	v	m/s				
	p Momentum	p	kg·m/s				
	q Displacement	q	m				
	\mathbb{I} Mass	m	Kg		$p = mv$	$F = \dot{p} = m \dot{v}$	$v = \frac{p}{m}$
	\mathbb{C} Spring†	K	N/m		$q = K^{-1}F$	$v = \dot{q} = K^{-1}\dot{F}$	$F = \frac{1}{K^{-1}}$
	\mathbb{R} Dashpot	b	N·s/m		$F = bv$	-	-
ROTATION	e Torque	τ	N·m				
	f Ang. Velocity	ω	rad/s				
	p Ang. Momentum	L	kg·m ² /s				
	q Angle	θ	rad				
	\mathbb{I} Inertia	I	Kg·m ²		$L = I\omega$	$\tau = \dot{L} = I\dot{\omega}$	$\omega = \frac{L}{I}$
	\mathbb{C} Torsion spring†	K	N·m/rad		$\theta = K^{-1}t$	$\omega = \dot{\theta} = K^{-1}\dot{t}$	$\tau = \frac{\dot{\theta}}{K^{-1}}$
	\mathbb{R} Ang. Dashpot	b	N·m·s/rad		$F = b\omega$	-	-
HYDRAULIC	e Pressure	P	Pa				
	f Vol. flow rate	Q	m ³ /s				
	p Flow momentum	r	kg·m/s				
	q Volume change	V	m ³				
	\mathbb{I} Flow inertia	I	Kg/m ⁴		$r = IQ$	$P = \dot{r} = I\dot{Q}$	$Q = \frac{r}{I}$
	\mathbb{C} Accumulator	K	Pa/m ³		$V = K^{-1}P$	$Q = \dot{V} = K^{-1}\dot{P}$	$P = \frac{V}{K^{-1}}$
	\mathbb{R} Restrictor	B	Pa·s/m ³		$P = BQ$	-	-
THERMAL	e Temperature	T	K				
	f Entropy flow	\dot{S}	W/K				
	p N/A*	-	-				
	q Entropy*	S	J/K				
	\mathbb{I} N/A*	-	-				
	\mathbb{C} Thermal store	?	?		?	?	?
	\mathbb{R} Thermal resistance	R	-		?	Often written as 2-port \mathbb{RS} element.	?

* Because magnetic monopoles do not exist, there cannot be magnetic admittances or true magnetic power dissipation.

† Most engineers don't measure springs in terms of compliance K^{-1} but use its inverse, stiffness K .

* Temperature → entropy is an irreversible transformation, due to the second law of thermodynamics.

Table A.2: Analogies between different system domains.

Table A.3: Examples of Integral Causal Bond Graph Symbols

Sources	Dissipative	Conservative	Transformer	Gyrator	0-Junction	1-Junction
$S_e \rightarrow$	$\rightarrow \mathbb{R}$	$\rightarrow \mathbb{I}$	$\rightarrow \text{TF} \rightarrow$	$\rightarrow \text{GY} \rightarrow$	$\begin{array}{c} \rightarrow \\ \text{(strong)} \end{array} 0 \leftarrow \begin{array}{c} \downarrow \\ \swarrow \end{array}$	$\begin{array}{c} \rightarrow \\ \text{(strong)} \end{array} 1 \leftarrow \begin{array}{c} \downarrow \\ \swarrow \end{array}$
$S_f \leftarrow$	$\leftarrow \mathbb{R}$	$\leftarrow \mathbb{C}$	$\leftarrow \text{TF} \leftarrow$	$\leftarrow \text{GY} \leftarrow$		

$$e_1 = Kf_2 \quad (\text{A.3})$$

$$f_1 = K^{-1}e_2 \quad (\text{A.4})$$

Common examples of transformers in the include levers, gearboxes, pulleys, rack and pinions, pistons, or electrical transformers. Examples of gyrators include electrical motors, spinning gyroscopes, solenoids, loops of wire, electrostatic attraction on capacitor plates, the Lorentz force, reluctance motors, and other connections which reflects a change of energy from one domain to another. Unsurprisingly, gyrators were named after the effect of gyroscopes, which turn their angular velocity (a flow) into a force (an effort) when perturbed. When modeling the magnetic domain explicitly, the gyrator modulus between the electric and magnetic domains is often the number of loops in a coil of wire, suggesting how transformers and gyrators can express geometric structure (Hamill, 1993).

If the scalar K relating effort and flow for transformers or gyrators are not constant but instead functions of a modulating variable, we may call them *modulated transformers* MTF or *modulated gyrators* MGY.

A.7 Drawing Bond Graphs

This section briefly discusses guidelines to follow when drawing integral-causal bond graph. Although modeling decisions are often based on intuition and experience and difficult to systematically describe, we may generalize the process of drawing a bond graphs with several steps: first by identifying 0- and 1-junctions in the system, then by connecting components to those junctions, and finally assigning causal strokes to help establish computability.

A selection of how components are most typically drawn with integral causality is shown in Table A.3. For a more complete treatment of how to draw bond graphs, the reader is encouraged to consult reference work on bond graphs(P. C. Breedveld, 2004) as only the basics are presented here.

To establish the connectivity of a typical electromechanical system, the following intuitive relations may be useful:

- Electrical components in parallel or experiencing the same voltage all connect to the same 0-junction.
- Electrical components in series or experiencing the same current all connect to the same 1-junction.
- Mechanical points where the velocity difference is important are represented with a 0-junction. Often the deflection rate of a spring is a result of just two other velocities, meaning mechanical 0-junctions with three connected bonds, one being a spring, are common.
- Mechanical points with a distinct velocity (such as masses) are represented with a 1-junction.

- Electric motors are gyrators connecting the rotational and electrical domains, and have a modulus equal to the motor's torque constant.

Regarding the connection of single port \mathbb{R} , \mathbb{I} , or \mathbb{C} components to the junctions or transformer/gyrators:

- A bond's half arrow points in the direction of positive energy flow, and although it may be arbitrarily chosen it is usually toward \mathbb{R} , \mathbb{I} , or \mathbb{C} elements.
- Labeling bond effort and flow signals as they connect to the junctions can help one maintain physical intuition regarding the graph.
- Lumping multiple elements of the same type together when they are connected to the same junction or transmission often simplifies the system model. (For example, the moments of inertia mass of a gear train with ratio n may be scaled by n^2 and lumped together with a mass on one side of the transformer representing the gear ratio.)

The selection of causality strokes to an existing bond graph can appear complex at first, despite the fact that it is in fact trivially automated on a computer. However, performing the process manually can be useful to verify that the modeled system is easily simulated. Causality conflicts often signal the presence of hidden components, junctions, or constraints. Some typical guidelines for assigning causality include:

- Assign source causality first: effort sources \mathbb{S}_e never have a stroke, and flow sources \mathbb{S}_f always do.
- A bond's causal stroke is placed on the side of the bond which specifies the *flow* of the bond. Under integral causality, this means that \mathbb{C} components never have a causal stroke on their side and \mathbb{L} components always do. \mathbb{R} components may or may not have a causal stroke.
- Always try to assign integral causality to \mathbb{I} and \mathbb{C} components.
- Verify that **0** junctions just have just one bond with a causal stroke, whereas **1**-junctions have all but one bond with a causal stroke on them.
- TF elements have no effect on causality, and causality propagates through them as if they did not exist.
- GY elements reverse the causality.

A.8 Multidimensional Bonds

Although only briefly mentioned and graphically shown in figure A.2b, it is relatively straightforward to expand the mathematics presented thus far so that effort and flow variables across a bond become vectors. This enables complex systems with symmetry to be described with fewer bonds and can describe extremely complex systems with a relatively compact diagram.

The road to vectorized multi-bonds is a short one. Let $\mathbf{x} = [p_1, p_2, \dots, q_1, q_2, \dots]$ be the vector of all integral causal component states, and $\mathbf{y} = [e_1, e_2, \dots, f_1, f_2, \dots]$ be the vector of efforts and flows. We then require only four conditions to hold for the bond graph to be valid:

1. The derivative of the state vector must be the effort/flow vector:

$$\mathbf{y}_i = \dot{\mathbf{x}}_i$$

2. The Hamiltonian must be the sum of all the stored state energy:

$$H(\mathbf{x}) = \sum_i \int y_i dx_i$$

3. The constitutive function $\Psi_i(\mathbf{x})$ of each component must equal a partial of the Hamiltonian:

$$\Psi_i(\mathbf{x}) = \frac{\partial H}{\partial x_i}$$

4. The duality of effort and flow must hold:

$$\frac{\partial y_i}{\partial x_j} = \frac{\partial^2 H(\mathbf{x})}{\partial x_i \partial x_j} = \frac{\partial y_j}{\partial x_i}$$

Although not strictly required, keeping all dimensions in each vector of the same causality is also highly recommended if the meaning of causal strokes is not to become confused.

A.9 Multi-Port Passive Components

Certain situations arise when it is necessary to model the multi-domain aspect of an \mathbb{R} , \mathbb{I} , or \mathbb{C} component. For example, the capacitance of a component depends upon the distance between the plates, and certain variable capacitors vary this distance and experience a varying force as they do so. If we wish to model this multi-domain effect using bond graphs, we must consider it as a two-port component connecting the translational and electrical domains (Figure A.8a).

In such cases, it is often useful to begin from the simple physical or geometric equations which determine lumped-element component values:

Capacitors The capacitance between two parallel plates is $C = \frac{\epsilon A}{l}$, where ϵ is the dielectric constant, A is the area, and l is the separation of the plates.

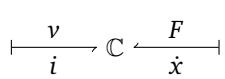
Inductors The permeance of a loop of wire is $P = \frac{\mu A}{l}$, where μ is material permeability, A is the area, and l is the length of the wire. For a group of N loops of wire, the inductance is $L = N^2 P$.

Resistors The conductivity of a wire is: $G = \frac{\sigma A}{l}$, where σ is material conductivity, A is the area, and l is the length of the wire.

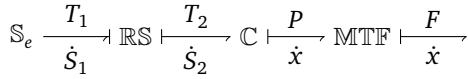
By inserting these relations into the Hamiltonian energy, the constitutive relation Ψ of the component can be found in terms of whichever variables are expected to vary. This technique can be used to model how heat source \mathbb{S}_e causes a gas to expand in a cylinder, losing heat energy but also increasing pressure to do mechanical work (Thoma & Ould-Bouamama, 2000), as shown in Figure A.8b.

A.10 From Bond Graphs to Hamiltonian Dynamics

Bond graphs and state-space representation of Hamiltonian systems are closely related. For example, the number of independent storage elements with integral causality reveals the dimension of the state of the system. Considerations of power conservation in any system naturally lead to representations that are equivalent to the Hamiltonian formulation in some way.



(a) Two-port C element.



(b) Model of a cylinder with heated gas.

Figure A.8: Examples of how inter-domain effects can be modeled with two-port passive elements.

A.10.1 Typical PHS Equations

Most control engineers are familiar with the state-space dynamics of a linear system with state \mathbf{x} , input \mathbf{u} , and observations \mathbf{y} :

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (\text{A.5})$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \quad (\text{A.6})$$

where \mathbf{A} is the system transition matrix, \mathbf{B} is the input matrix, \mathbf{C} is the output matrix, and \mathbf{D} is the feedthrough matrix. These state-space equations have rich history and are well suited for a signal-processing or control viewpoint for general systems that may or may not satisfy the conservation of energy. But when modeling physical systems with bond graphs, the extra constraints provided by the more energetically rigorous Port-Hamiltonian formulation of the system dynamics (*i.e.*satisfying the conservation of power) results in more internal structure to the equations.

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}}) \frac{\partial H}{\partial \mathbf{x}} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \\ \mathbf{z} &= \mathbf{B}^T \frac{\partial H}{\partial \mathbf{x}} \end{aligned}$$

where \mathbf{A}_{01} is a skew-symmetric connectivity matrix, $\mathbf{A}_{\mathbb{R}}$ is a diagonal matrix expressing the dissipative elements, and H is the Hamiltonian of the system. Let us examine this a little further by considering things from a power constraint perspective.

A.10.2 Power Flow Constraints

The real advantage of the Port-Hamiltonian formulation is that it becomes trivial to compute the powers flowing around a system. Recall that the Hamiltonian is a function of $\mathbf{x} = [p_1, p_2, \dots, q_1, q_2, \dots]^T$ and constitutive functions $\Psi_i(\mathbf{x})$:

$$H(\mathbf{x}) = \int \sum_i \Psi_i(\mathbf{x}) d\mathbf{x}$$

Let $\psi(\mathbf{x}) = \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \Psi_L(x_1) \\ \Psi_C(x_2) \end{bmatrix}$ be the vector of constitutive functions (*i.e.*the partials of the Hamiltonian). Then for linear constitutive functions, the Hamiltonian is a quadratic form:

$$H(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \psi$$

The rate of change of the stored system energy (*i.e.* the power $P_{\mathbb{C}}$ flowing out of the conservative components) is thus

$$\dot{H}(\mathbf{x}) = \sum_i \dot{x}_i \Psi_i(x_i) = \dot{\mathbf{x}}^T \psi = P_{\mathbb{C}}$$

Since matrix \mathbf{A}_{01} is skew symmetric and represents an internal rotation or mixing of stored power, $\psi^T \mathbf{A}_{01} \psi = 0$. This means that

$$\psi^T \mathbf{A} \psi = \psi^T (\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}}) \psi = \psi^T \mathbf{A}_{\mathbb{R}} \psi = P_{\mathbb{R}}$$

and the only way to change the energy of a system is via internal dissipation $P_{\mathbb{R}}$ or an externally supplied energy P_S :

$$\begin{aligned} P_{\mathbb{R}} &= \psi^T \mathbf{A}_{\mathbb{R}} \psi \\ P_S &= \mathbf{u}^T \mathbf{B}^T \psi \end{aligned}$$

Therefore, the constraint for power consistency for a linear system is that

$$\dot{H}(\mathbf{x}) = P_{\mathbb{R}} + P_S$$

We can see the how this instantaneous constraint on system power comes from the state space equations, which is true for nonlinear ψ as well.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & 0 \end{bmatrix} \begin{bmatrix} \psi(\mathbf{x}) \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{z} \end{bmatrix}$$

This is in fact the solution to the quadratic programming problem

$$\min_{\psi} (\psi^T \mathbf{A}_{\mathbb{R}} \psi - \dot{\mathbf{x}}^T \psi) \quad \text{s.t. } \mathbf{B}^T \psi = \mathbf{z}$$

So we see that fundamentally, the dynamics of a Hamiltonian system are minimizing power – or should I say, conserving power – for the system, subject to a linear boundary constraint expressing the power added to the system. The boundary constraint is defined by a hypersurface \mathbf{B} , and control effort \mathbf{u} can be thought of as being some sort of the lagrange multiplier of the boundary conditions which push hypersurface \mathbf{B} around.

A.10.3 Interpretation of a PD Controller

If our hamiltonian system is

$$\dot{\mathbf{x}} = (\mathbf{A}_{01} - \mathbf{A}_{\mathbb{R}}) \psi + \mathbf{B} \mathbf{u}$$

and we have a setpoint \mathbf{x}_d that we wish to track, a simple PD controller that stabilizes the system is merely

$$\mathbf{u}(\mathbf{x}) = \mathbf{k}^T (\mathbf{x}_d - \mathbf{x})$$

where \mathbf{k} is a vector of the gains of the system for each variable essentially similar to the PD gains of a PID controller.

A.11 Examples

Let us demonstrate with two simple examples how easy it is to derive each form of the dynamical equations of a system from its bond graph.

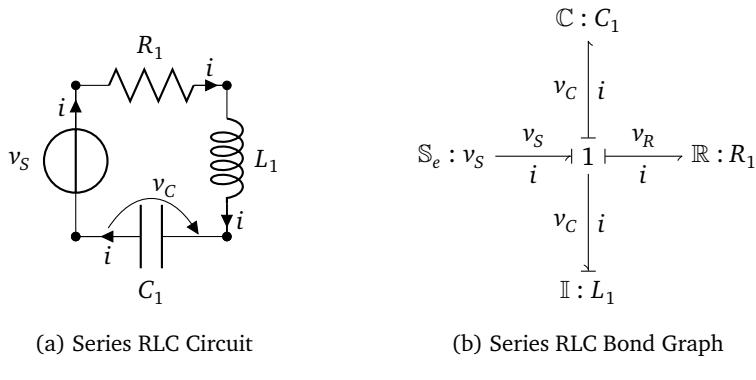


Figure A.9: Example series RLC circuit and bond graph

A.11.1 Example: Series RLC Circuit

Let's say we have a series RLC circuit excited by a voltage source, resulting in the bond graph shown in Figure A.9b. Only the inductor and capacitor store energy, so our state vector must have just two components: inductor flux linkage λ and capacitor charge Q .

$$\mathbf{x} = \begin{bmatrix} \lambda \\ Q \end{bmatrix}$$

The system input vector is also simple to write:

$$\mathbf{u} = \begin{bmatrix} v_s \end{bmatrix}$$

Linear components have simple constitutive functions

$$\begin{aligned} i_L &= \Psi_L(\lambda) = \frac{\lambda}{L} \\ v_C &= \Psi_C(Q) = \frac{Q}{C} \end{aligned}$$

So the Hamiltonian of the RLC circuit is quadratic

$$H = \frac{1}{2} (Q\Psi_C(Q) + \lambda\Psi_L(\lambda)) = \frac{1}{2C}Q^2 + \frac{1}{2L}\lambda^2$$

Now let's write the system's first order differential equation. The rate of the change of state is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\lambda} \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} v_L \\ i_C \end{bmatrix}$$

Our goal is to express v_L and i_C in terms of the state vector \mathbf{x} so it is a trivial ODE to simulate. Well, $i_C = i_L$ because there is only one current loop. Also, $v_L = v_s - v_C - v_R$. We can immediately write each of these quantities using only components from \mathbf{x} .

$$\begin{bmatrix} \dot{\lambda} \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} v_s - \frac{Q}{C} - R\frac{\lambda}{L} \\ \end{bmatrix}$$

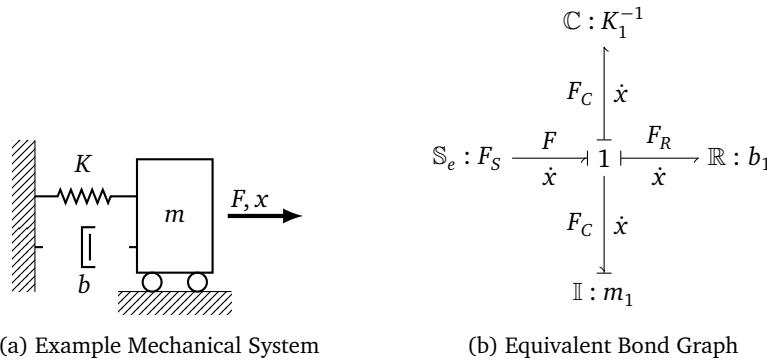


Figure A.10: Spring-mass-damper system and equivalent bond graph.

More properly written, this is

$$\begin{bmatrix} \dot{\lambda} \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{C} \\ \frac{1}{L} & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ Q \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [v_s]$$

Or even more simply, if we let $\psi(\mathbf{x}) = \frac{\partial H(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \Psi_L(x_1) \\ \Psi_C(x_2) \end{bmatrix}$ be the vector of constitutive functions (*i.e.* the partials of the Hamiltonian),

$$\begin{bmatrix} \dot{\lambda} \\ \dot{Q} \end{bmatrix} = \left(\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} -R & 0 \\ 0 & 0 \end{bmatrix} \right) \psi(\mathbf{x}) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [v_s]$$

This is a typical form of the equations for Port Hamiltonian Systems.

$$\begin{aligned} \dot{\mathbf{x}} &= [\mathbf{A}_{01} + \mathbf{A}_{\mathbb{R}}] \frac{\partial H}{\partial \mathbf{x}}(\mathbf{x}) + \mathbf{B}\mathbf{u} \\ \mathbf{z} &= \mathbf{B}^T \frac{\partial H}{\partial \mathbf{x}}(\mathbf{x}) \end{aligned}$$

where skew-symmetric matrix \mathbf{A}_{01} describes internal system connectivity, positive definite matrix $\mathbf{A}_{\mathbb{R}}$ describes the dissipative elements, \mathbf{B} describes connectivity to external systems, and the pairs of \mathbf{u} and \mathbf{z} make the 'input/output port' of the system.

A.11.2 Example: Spring-Mass-Damper System

Let's now consider an analogous mechanical system (Figure A.10a) with damper b and spring K connecting mass m to an immobile object. We apply an input force F_S to control the motion of the system.

Given that the only two state variables are the length of the spring q and momentum p of the mass, the elements of several vectors is immediately obvious.

$$\mathbf{x} = \begin{bmatrix} p \\ q \end{bmatrix}, \dot{\mathbf{x}} = \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} F_m \\ \dot{q} \end{bmatrix}, \mathbf{u} = [F_S], \psi = \begin{bmatrix} \frac{p}{m} \\ Kq \end{bmatrix}$$

The single 1-junction specifies the following balance of forces:

$$\begin{aligned} F_m &= -Kq - b\dot{q} + F_S \\ &= -Kq - \frac{b}{m}p + F_S \end{aligned}$$

Therefore, the equation of motion is

$$\begin{bmatrix} F_m \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -b & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{p}{m} \\ Kq \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [F_S]$$

Or, breaking it apart into two matrices

$$\dot{\mathbf{x}} = \left(\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} -b & 0 \\ 0 & 0 \end{bmatrix} \right) \psi + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathbf{u}$$

Note how \mathbf{A}_{01} and $\mathbf{A}_{\mathbb{R}}$ are identical to the series RLC circuit case, and how the Hamiltonian is also essentially the same.

$$H(x) = \frac{1}{2G} q^2 + \frac{1}{2m} p^2$$