

AN ABSTRACT OF THE DISSERTATION OF

Andrew M. Abate for the degree of Doctor of Philosophy in Robotics presented on April 11, 2018.

Title: Mechanical Design for Robot Locomotion.

Abstract approved:

Jonathan W. Hurst

Ross L. Hatton

Robotic limbs have been shown to enable mobility in unstructured, real-world terrain; they allow robots to step around cluttered environments, scramble up hills, carry heavy loads, and even perform acrobatics. However, mechanical limbs cannot operate as a means for such dynamic locomotion if they are simply treated as general articulated mechanisms. Mechanical design in the context of locomotion requires us to be aware of the control paradigm (sometimes even serving as a *part* of the controller) and how the robot relies on contact to generate motion. These two interfaces (software control, physical contact) direct the design of inherent hardware dynamics (passive dynamics), where mechanical design decisions are made specifically to serve the context. In this work, we create a practical definition of *locomotion* to lead us towards metrics for grading a design's aptitude for real-world mobility and tracking progress toward better designs. Metrics such as weight, cost, and other engineering goals are commonly understood and not considered here. Instead, we are interested in the locomotion-specific metrics of floating-base manipulability, impact reduction, power quality, and compliance tuning. These metrics have intuitive, geometric representations for use in human-driven design processes, while their numerical representations can be evaluated in simulation. Multiple competing metrics require a multi-objective design process, which here is discussed as stochastic search. This structure is analogous to a human-driven design process, but is also amenable to future computer automation.

© Copyright by Andrew M. Abate
April 11, 2018
All Rights Reserved

Mechanical Design for Robot Locomotion

by
Andrew M. Abate

A DISSERTATION

submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented April 11, 2018
Commencement June 2018

Doctor of Philosophy dissertation of Andrew M. Abate presented on April 11, 2018.

APPROVED:

Co-Major Professor, representing Robotics

Co-Major Professor, representing Robotics

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Andrew M. Abate, Author

TABLE OF CONTENTS

	<u>Page</u>
1 INTRODUCTION	1
1.1 Contribution	5
1.2 Outline	6
2 LOCOMOTION SYSTEMS	7
2.1 The first walking robots, static stability	8
2.2 Dynamic stability	9
2.3 Passive dynamics	10
2.4 Active dynamics through high-transparency actuators	12
2.5 ATRIAS and Cassie	13
3 MODELING VIRTUAL PROTOTYPES	20
3.1 Software, hardware, and the world	20
3.2 Core mechanical system	22
3.3 Control interface	27
3.4 World interaction	31
4 DEFINING THE LOCOMOTION TASK	34
4.1 Abstract locomotion	34
4.2 Spring-mass locomotion	36
4.3 Through-contact trajectory optimization	40
5 PERFORMANCE METRICS	41
5.1 Efficiency metrics	43
5.2 Manipulability metrics	52
5.3 Robustness metrics	57
5.4 Feasibility metrics	60
6 METRIC-DRIVEN DESIGN PROCESS	62
6.1 Automated design generation	62
7 EXAMPLE DESIGNS	66
7.1 Comparison	72
8 CONCLUSION	76
8.1 Disclosure	77
8.2 Acknowledgments	77
BIBLIOGRAPHY	78

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	Limbed locomotion	1
2	Candidate locomotor design	4
3	PV-II	8
4	Humanoids	10
5	Leg design silhouettes	11
6	Passive-dynamic walking	11
7	IHMC Elliptical Runner	12
8	ATRIAS	15
9	Cassie	16
10	Cassie leg design	18
11	ATRIAS motor power requirements	19
12	Three dynamic subsystems	21
13	Multibody system	23
14	Plate-spring models	30
15	Spring-loaded inverted pendulum model	37
16	Antagonistic power	45
17	Power quality	47
18	Power quality in three leg designs	48
19	Force, velocity decomposition	49
20	Compliance dyads	52
21	Acceleration limit	55
22	Impact inertia	59
23	Quadratic loss	61
24	Corner sort	63
25	Design iteration	64
26	Polar design	70
27	Serial design	71
28	Parallel design	73

*“In design, one of the most difficult activities
is to get the specifications right.”*

— Donald A. Norman, *The Design of Everyday Things*

1 INTRODUCTION

We want to improve robot mobility. Wheels and tracks are good for traversing simple, smooth, and open environments, but *limbs* are the key to robust terrestrial locomotion (Figure 1). The seemingly simple use of limbs for self-transport is a gating issue for a stampede of applications: helper robots in homes, carrying bags, and moving furniture; courier robots on the streets delivering packages; search and rescue robots in forests and caves assisting disaster response; geological robots climbing the mountains of Earth or Mars to survey the landscape; and so on as far as our imagination takes us. Such activities seem trivial to us, but locomotion is a deceptively complex problem.

Dynamic locomotion is as much a hardware design problem as it is one of control. Studies in control theory abound, but there is a relative fuzziness in the controller's companion: a purposeful mechanical design. The algorithms and processes which make up a controller cannot directly act on the environment, so they need a *vehicle* or *transducer* to act through. Hardware must always serve that function by providing a controllable and efficient platform for locomotion. Clearly, a locomotor will be underwhelming if its hardware dynamics are too slow or heavy for walking and running. Conversely, a nimbler

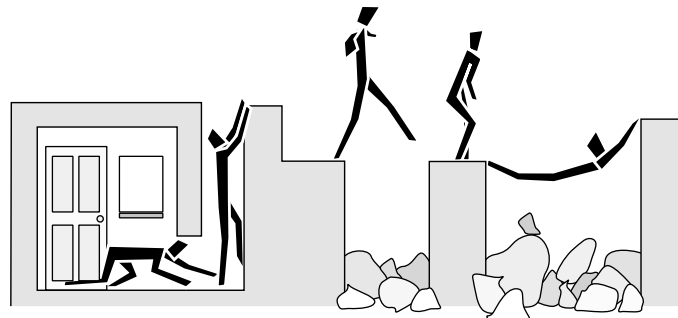


Figure 1: Limbed robots have the potential for robust terrestrial locomotion in diverse settings; they can fold to squeeze through tight spaces, extend to reach ledges, hurdle over or bridge gaps in the environment, and use arms for both locomotion and manipulation. A wheeled or tracked robot is limited by its fixed size, either being too large to fit through a space or too small to get over an obstacle [82].

hardware design could increase controllability and even manage real-world events (e.g. impacts) that are above the controller's bandwidth. It is even possible for mechanical design to affect how complex the control system must be (e.g. Cartesian kinematics make IK/FK trivial [58], gibbons effortlessly swing through natural pendular trajectories [17, 147], dead fish can "swim" upstream because of their bodies' passive dynamics [16], etc). These facts combined with the fixed nature of a hardware design makes mechanism dynamics critical to "get right"; software can be patched, but it is certainly challenging to hotfix deployed machines.

So, we have a big question to tease apart: **"how can we make better locomotion hardware?"**. As humans, we have an innate sense of animals walking and running, but what does "locomotion" mean for a robot? Does better performance depend more on control, hardware dynamics, or some balance of the two? How do we even define *better*? We sometimes have clear metrics for *better* when it comes to making the robot lighter, or selecting actuators with higher torque density, but we are quite often forced to make judgment calls[†] when confronted with the complexity of a locomotion system. Does the robot work better with forward-facing knees or backward, and why? Is it important where actuators are located in the robot, or do the limbs simply need to span the workspace? How much do we need to know about the control system before designing the hardware (e.g. [92])? Is physical compliance good or bad, or both? With enough effort, we can answer these questions and find better designs by using a formal design process.

The goal of this work is to better understand how the demands of *dynamic locomotion* influence a robot's *mechanical design*, and how to manage design goals and constraints in a methodical design process. In order to improve designs, we must have a rigorous structure for ideation, evaluation, and iteration. This repeated generation and ranking of designs can be formalized as an optimization problem. *Optimization* is perhaps the wrong word due to the levity of human thought and preference as compared to a computer, but the format of an optimization problem gives us the perfect structure for our

[†] Judgement calls in design can be troublesome; people have different "gut instincts" about design, sometimes leading a project around in circles. A project needs to decide on the appropriate metrics for success in order to make structured progress.

design problem and forces us to be specific about our design goals. It also tempts us with the possibility of automating some or all of the process of finding better designs.

If the chosen metrics are insufficient or direct the search in an undesired direction, then the metrics need to be re-evaluated. That being the case, choosing metrics is just as much a problem as choosing a design. We do not have a list of design objectives which definitively produce optimal designs, but getting closer to that ideal is a main driving force behind this work.

1.0.1 *Formulating the design problem*

Let us cast the locomotor design problem into the same three parts as an optimization problem, as these will guide us in teasing apart the question of designing better machines for locomotion:

- A **design space** of all the possible candidate robots. This is all the different configurations of links, masses, actuators, and springs. However nonsensical some designs may appear to us, *all* possible designs are necessarily in the search space.
- A set of **design objectives** allow candidate designs to be ranked against each other. Some objectives may apply strictly to the structure and composition of the robot, and others may also require a *task definition* such as a walking trajectory. All objectives discussed in this work relate specifically to locomotion. Engineering objectives such as component mass and cost must obviously be considered, but they are intentionally avoided in favor of locomotion metrics.
- **Constraints** ensure the *feasibility* of designs. These may apply at the design level where links cannot have negative mass or be shorter than a given length, or they can enforce the peak torques and speeds of actuators, or otherwise make sure that the robot is physically capable of following a task's trajectory. Engineering constraints such as component failure and fatigue are not discussed, but obviously must be included.

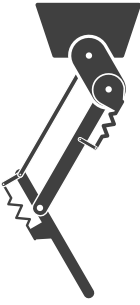
Design	Objective	Score	Order
	Mass	30 kg	<
	CoM acceleration limit	2.1 g	>
	Toe acceleration limit	25 g	>
	Nominal compliance	11 kN/m	=
	Axial compliance	+	(-,~,+)
	Impact inertia	2.5 kg	<
	Power quality	~	(-,~,+)
	Physics violation	-	(-,~,+)

Figure 2: Candidate designs are members of the search space and are evaluated according to the project's objectives. Objectives may or may not carry with them extra information about the use case or the task, such as trajectories, controllers, or environments.

Each design *candidate* is a member of the design space that has been evaluated according to the design objectives (Figure 2). Depending on the objectives, each candidate may need to include extra data to describe the task. For example, candidates may be evaluated relative to some common reference trajectory, reference controller, or an optimization which determines optimal trajectories for each new candidate.

1.0.2 Solving the design problem

Given the combinatorial structure of the search space, the locomotor design process is naturally iterative [26, 72]. We cannot deterministically arrive at the ideal design, so successive candidates must build upon earlier successes and failures. We also cannot know when a solution is truly optimal, but we can easily check that some identified solutions are better than others. So, the only way to solve the design problem is to continually find better designs. We can never find the *ultimate* design.

In fact, there is no *ultimate* design, because having multiple competing objectives means that designers must make trade-offs depending on the nature and scale of their project. The existence of trade-offs means that we are inherently searching for an optimal *set* of designs. It is up to the designer to decide what trade-offs to make and which design to pursue. Designs can either be hand-picked from the set of current best designs, or designers can

weight different metrics to give each design a single combined score with a single winner [6].

We cannot evaluate designs without first generating them, so we need a way to instantiate particular designs from the search space. The design space we consider in this work is all the different assemblages of rigid bodies, joints, actuators, and passive springs. Algorithmically searching over such a space requires a formal grammar (production rules) for strings describing valid members. Fortunately, humans are extremely good at idea generation, so we leave it up to them for the time being. The only downside of ideation is ensuring good coverage of the search space, but generating all strings from a grammar can be equally difficult (depending on the type of grammar). Regardless, this is precisely why we have industrial designers and engineers.

1.1 CONTRIBUTION

This work is the result of trying to create better hardware for locomotion systems. It investigates how a robot’s mechanical design is influenced by the demands of locomotion, and determines logical design metrics and goals.

I have led the mechanical design of two bipeds at Oregon State University: ATRIAS 2.0/2.1 followed by three revisions of the Cassie biped[†]. It took a substantial amount of time to understand the specialized mechanical requirements of locomotors and how connected the hardware is to the control system, so I aim to render that knowledge here.

I have developed the use of task-space compliance and inertia in design [2], and I have derived the Power Quality metric to maximize mechanical efficiency under uncertainty [3]. These metrics were used in the Cassie design to minimize the amount of power lost to motor braking while maintaining good passive dynamics.

Wherever possible, I translate numerical metrics into tangible, geometric forms which provide intuition about candidate designs. Visual forms are often self-explanatory, easily transferring design intuition while also being backed by concrete mathematics.

[†] The Cassie biped design is licensed to Agility Robotics for production and sale. Cassie robots are being sold to universities working to advance legged locomotion capability.

1.2 OUTLINE

LOCOMOTION SYSTEMS Previous examples of locomotion systems show the breadth of thought and approaches to locomotion.

MODELING VIRTUAL PROTOTYPES It is not feasible to build and test all candidate designs in the real world, so we must sometimes simulate *virtual prototypes*. There is substantial modeling effort required to simulate a mechanical system interacting with the world through contact.

DEFINING THE LOCOMOTION TASK We need a clear definition of what the robot is trying to accomplish before we can create a list of metrics to rank candidate designs. Locomotion has both *abstract* and *concrete* task definitions in that both "system self-transport" and "left-right-left-right" walking *are* locomotion. Both of these definitions are good for different purposes.

PERFORMANCE METRICS With the task definitions, we can arrive at logical design goals which specifically target locomotion performance. We consider *performance* in terms of energy efficiency, manipulability, robustness, and feasibility.

METRIC-DRIVEN DESIGN PROCESS Finally, the virtual modeling, task definition, and objectives are collected into an iterative design process which is most closely described as stochastic search. Such a formalism is analogous to the engineering design process, and it is also amenable to computer automation.

EXAMPLE DESIGNS We show how limb-design metrics can be represented visually to judge candidate designs. We evaluate the acceleration limit, impact inertia, and compliant response of a polar design, a serial design, and a parallel design.

2 LOCOMOTION SYSTEMS

Researchers, engineers, and artists have been trying to make walking machines for at least a century[‡] [137], and we are only beginning to see glimpses of natural agility and efficiency. The robots with the most natural movement tend to be those which have been focused on hardware designed specifically for locomotion [63, 93, 115]. Designs which do not consider their passive response to contact tend to be the most "robotic", attempting to maintain static equilibrium and move slowly to avoid exciting unwanted dynamics (e.g. humanoids like ASIMO and Valkyrie).

Since the early 1980s, marking the beginning of using computers to drive legged machines, the robotics community has created a substantial number of robots designed for locomotion [77]. Anthropomorphic, "humanoid" robots compose the majority of the catalog, while there are also a number of monopod hopping machines [65] designed principally for research in dynamic locomotion and stability as well as bi-, quad- [66, 119], and hexapods [129]. Most of the early robots, and especially humanoids, use rigid gearmotors or hydraulics for actuation. Since the invention of the Series Elastic Actuator (SEA) [114, 126] and other compliant actuators [56], we have also seen a number of compliant leg designs. More recently, we have seen a number of smaller robots eschewing gearmotors in favor of high-transparency, direct-drive electric torquer motors. There are also a number of machines designed to rely on hardware dynamics for stability, using only minimal (or zero) actuation or control.

[‡] See Reuben Hoggett's wonderful collection of early walking machines at <http://cyberneticzoo.com/walking-machine-time-line/>

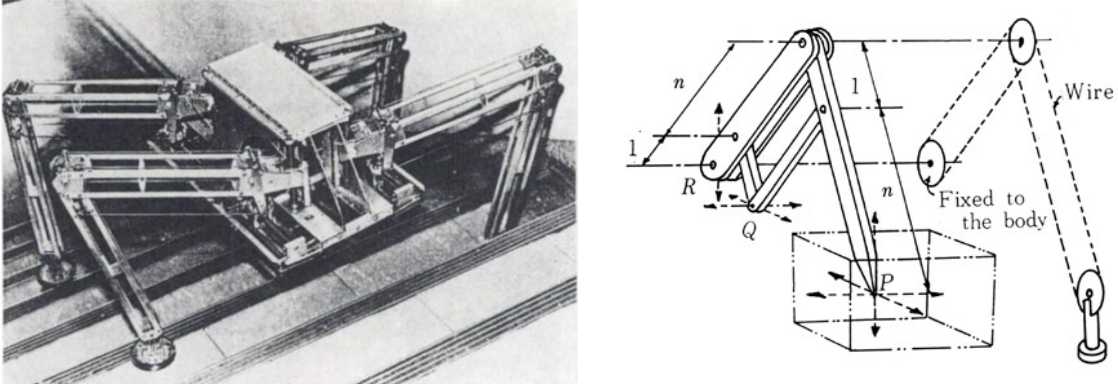


Figure 3: The 1978 PV-II "daddy long-legs" robot and its Cartesian pantograph leg mechanism [58].

2.1 THE FIRST WALKING ROBOTS, STATIC STABILITY

Early walking machines in the 1980s were designed to accommodate simple theories of balance based on static stability. Robots such as McGhee's hexapod [95, 107] used a wide base of support for better stability, and many humanoids to this day still have exaggerated foot size for better area of support. This approach is found in Zero-Moment Point gaits [150], where the robot moves slowly to reduce dynamic effects and ensures its mass-center is always over its area of support.

Early robots had to contend with the lack of available computing power, so the PV-II robot used a Cartesian leg design to mechanically reduce the cost associated with calculating kinematics [58]. The PANTOMECH legs used on the PV-II move in 3 orthogonal directions aligned with the robot's Cartesian frame, making forward and inverse kinematics trivial (Figure 3). This is a great example of how a robot's hardware can be designed specifically to aid a controller.

2.1.1 Humanoids, Static stability

Humanoid robots tend to be built as platforms for general control and manipulation. They have predominantly rigid actuation and follow precise motion trajectories, much like their fixed-base manipulator ancestors. Rigid humanoids have high acceleration potential when

they are in contact with rigid ground, and they can follow undisturbed trajectories, but they tend to be delicate and have small basins of attraction (Figure 4).

Rigid humanoids are typically designed with enough arm, leg, and body actuators to remain statically stable and interact with the environment. This allows them to be fully actuated and fully controllable (within bounds), but the robots must have detailed maps of the environment to avoid collisions which cause large force spikes and potentially damage their rigid gearing. Large, uncontrollable impulses also make rigid humanoids sensitive to disturbances. In order to get around these deficits, humanoids typically have large foot pads to deaden impacts, and they are given a wide berth so as to not accidentally cause unmodeled contact events.

One outlier is the ATLAS robot (Figure 4), which uses high-power hydraulics for more robust locomotion. Driven by a number of competing teams including MIT and IHMC at the DARPA Robotics Challenge in 2015, the ATLAS robot had some very real success (and also a lot of failures, resulting in a gag reel watched more times than any other video of the event). Using an ATLAS torso, Boston Dynamics' Handle robot even has wheels for feet (because, why not?).

SRI's DURUS humanoid (also at the DRC as a dynamic demo) uses series-elastic ankles for efficient locomotion [120], and IIT's Coman [146] uses series-elastic actuators in most of its joints.

2.2 DYNAMIC STABILITY

Static stability is limited to known, stable terrain, so robots needed to begin relying on *dynamic* stability to make it into the real world. The idea of dynamic stability allowed mono-, bi-, and quadruped robots to confidently move through an unknown and dynamic world, reject heavy disturbances, and even do acrobatics while remaining stable [117, 118]. Marc Raibert's machines as well as many other hoppers [5, 67, 161] and bipeds [63] are able to move robustly with the combination of passive compliance and intuitive control laws (such as placing the next step farther forward for higher speeds and injecting energy with

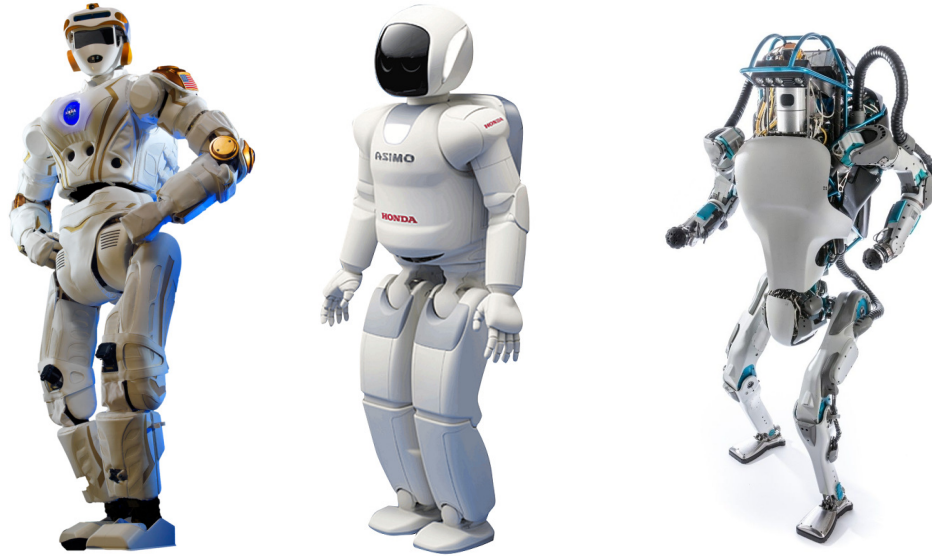


Figure 4: NASA's Valkyrie, Honda's ASIMO, and Boston Dynamics' ATLAS.

each stride). Intuitive controllers inspired by Raibert's work have enabled the compliant MABEL [111], ATRIAS, and Cassie bipeds to walk and run at Oregon State University. These and other machines try to embody reduced-order dynamics in order to simplify control [47].

Other early hoppers include Brown's Bow Leg Hopper [23], Andrews' hopper [12], and the Papantoniou straight-line linkage hopper [109] (Figure 5). Like Raibert's hoppers, these are designed with a low leg-to-torso inertia ratio and leg angle axes near the CoM for simplified dynamics. As such, they are amenable to intuitive control schemes and are used to test new theories in dynamic balance based on the spring-loaded inverted pendulum (SLIP) and other reduced-order models.

2.3 PASSIVE DYNAMICS

Taking the idea of optimized kinematics/dynamics to the extreme, passive-dynamic walkers and runners can generate gaits without any control, electronics, or motors whatsoever (Figure 6). Tad McGeer created a fully-passive mechanism — a pair of legs — which could "walk" down a slight slope, powered only by gravity [94]. Since they lack any control or actuation, these mechanisms are very sensitive to disturbances and cannot walk over level

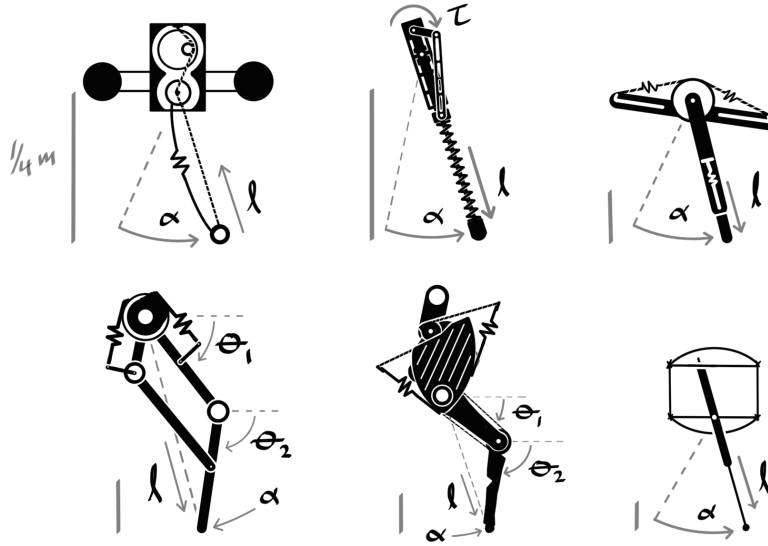


Figure 5: Silhouettes of various compliant leg designs. From top left: bow-leg hopper, Andrew's hopper, ARL monopod, ATRIAS, MABEL, Raibert's hopper.

ground [31, 32]. However, McGeer's walkers evolved into minimally-actuated robots which can walk on flat ground with astounding efficiency. For example, Cornell's Ranger robot was able to walk an Ultra-Marathon on a single battery charge [18]. So, even with access to high-powered mobile computers, designing a robot's hardware for a specific purpose can still have huge advantages.

Other examples of designing for passive dynamics include IHMC's HexRunner, FastRunner [102], and EllipticalRunner. HexRunner is a rimless wheel with spring-loaded spokes which generate a spring-mass gait when spinning at a nominal rate. EllipticalRunner has a

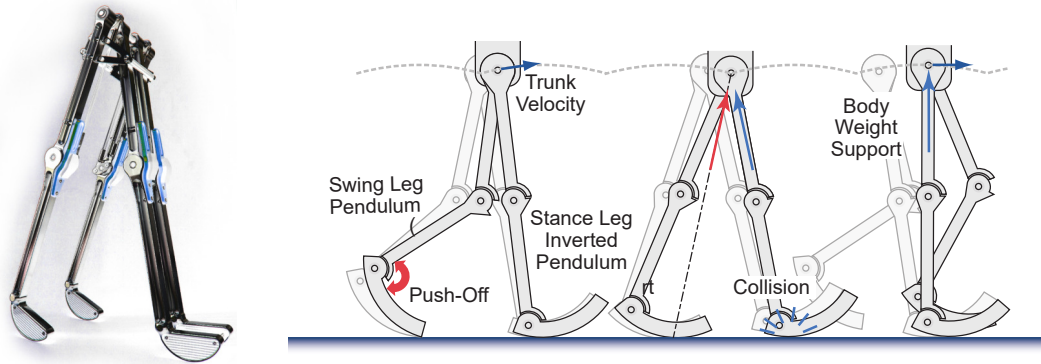


Figure 6: BlueBiped, a passive-dynamic walking mechanism from Nagoya Institute of Technology. Strobograph from [81] shows the walking sequence (toe-off torque only applies to actuated walkers, where BlueBiped is passive and requires an external force to walk continuously).



Figure 7: IHMC Elliptical Runner gait.

sophisticated mechanism which automatically generates toe trajectories which adapt to real-world state, not a measured state (Figure 7). EllipticalRunner has only a single motor which powers two out-of-phase legs. There is significant compliance onboard, both in the leg-length direction for spring-mass ground forces and in the drive cranks to allow the legs to become more in-phase under higher load.

Obviously, these heavily under-actuated designs are only effective at a handful of tasks; their fixed designs are not adaptable to tasks like stairs, stepping stones, or ladders.

2.4 ACTIVE DYNAMICS THROUGH HIGH-TRANSPARENCY ACTUATORS

Locomotion is principally concerned with controlling contact forces (Section 4), but high-ratio gearmotors make force control nearly impossible. A number of smaller robots have used direct-drive motors and ultralight legs to maximize transparency for locomotion. Since electric motors are a pure torque source, there is no reason to measure and feed back on applied force as in an SEA; the commanded force is exactly the contact force, neglecting the minimal dynamic effects of lightweight links.

Robots such as the MIT Cheetah [154], Jerboa [73], and Minitaur [73] take this approach, allowing for simplified mechanisms, better contact force transparency, and software-controlled impedance with almost zero leg inertia. Such designs dramatically reduce contact impulses and simplify control since they produce almost exactly the contact force that is commanded. The direct-drive robots can be slightly lighter since they do not require the steel and bearings of gearmotors.

Force transparency comes at a cost, however, in the form of resistive heating. Electric motors are most efficient when they are driven at high speeds and low torque, but direct-

drive motors operate at low speeds and high torque. Applying higher torque requires more current through the motor windings, producing much more heat than mechanical power. Resistive heating dominates the power requirements of low-reduction robots. For example, the MIT Cheetah dissipates 75% of its electric power output to resistive heating [133]. Even so, the MIT Cheetah has a very good total cost of transport at 0.7.

Direct-drive approaches minimize reflected inertia, allowing for easier feed-forward control, but electric motors are just not ideal for high torque. Perhaps electric motors will one day be replaced by actuators more similar to biological muscle, which is great at efficiently producing torque. Muscle is so force-dense that it is generally geared up (increasing velocity and decreasing force). Direct-drive joints are phenomenally better for locomotion than geared joints, but we will have to wait for actuator technology to improve before seeing inertialess, high-torque robotic joints.

2.5 ATRIAS AND CASSIE

I have led the mechanical design of two robots which try and find the right balance between passive dynamics and software control: the ATRIAS and Cassie bipeds at Oregon State University (Figures 8 and 9). ATRIAS was designed to have trivial hardware dynamics to allow for intuitive, heuristic controllers. Cassie was designed to be a more production-ready robot, intended to be eventually sold by Oregon State’s spinoff company Agility Robotics. Cassie makes compromises in favor of mechanical simplicity rather than dynamical simplicity, forcing us to investigate more advanced modeling and control techniques.

2.5.1 ATRIAS

Perhaps the best way to describe the philosophy of ATRIAS is to explain the name: Assume The Robot Is A Sphere. Introductory physics classes often tell us that everything from cars to cows can be modeled by simple point masses, allowing us to calculate their gross motion. Despite being aesthetically absurd, there is a lot of truth to this approach; all

mechanical systems have a center of mass which has a proportionality between its own acceleration and the forces acting upon the system it represents.

ATRIAS attempts to gather all its mass near the hip joints so that the lightweight pantograph legs rotate precisely around the robot's CoM [53, 54]. Series springs connect the highly geared leg motors to the pantograph, making the overall dynamics of ATRIAS very close to a simple spring-mass model. Each leg acts as a nonlinear, massless spring between the ground contact and the "point-mass" robot.

It took a considerable amount of effort and compromise to force all of ATRIAS's mass into the hips, but we were sure that it would pay dividends in the form of simple heuristic controllers [79]. If the robot's dynamics approximated a simple spring-mass model, then we ought to be able to control it just like a simple spring-mass model. The idea was that we could develop heuristics on simulated models with simple dynamics and have those gaits work on the real robot. However, even when going to extremes to match reduced-order dynamics, ATRIAS's dynamics were different enough to invalidate many of the reduced-order controllers. In the end, ATRIAS required quite a lot of tuning and human involvement in creating workable controllers [62, 123, 124].

2.5.2 *Cassie*

After the maintenance nightmare of ATRIAS, we were focused on making Cassie a simpler robot to operate and maintain. Where ATRIAS had encoders and wires buried underneath layers of bearings and motors, Cassie's electronics are all immediately accessible. Each joint is its own module, allowing for easy replacement of different components. Did a bad fall destroy the lower leg? Four bolts and a wire bundle is all it takes to remove and replace.

Cassie has similar series compliance to ATRIAS, with two springs in the lower leg buffering impacts and storing energy within the leg plane. Interestingly, series compliance does not have to be directly in series with the motors. On Cassie, the hip is rigidly actuated, and the knee has a series spring. The knee spring only provides one dimension



Figure 8: ATRIAS at the OSU football field. ATRIAS has only three degrees of freedom per leg. Abduction raises and lowers the planar leg, and two concentric motors operate the pantograph in the plane.

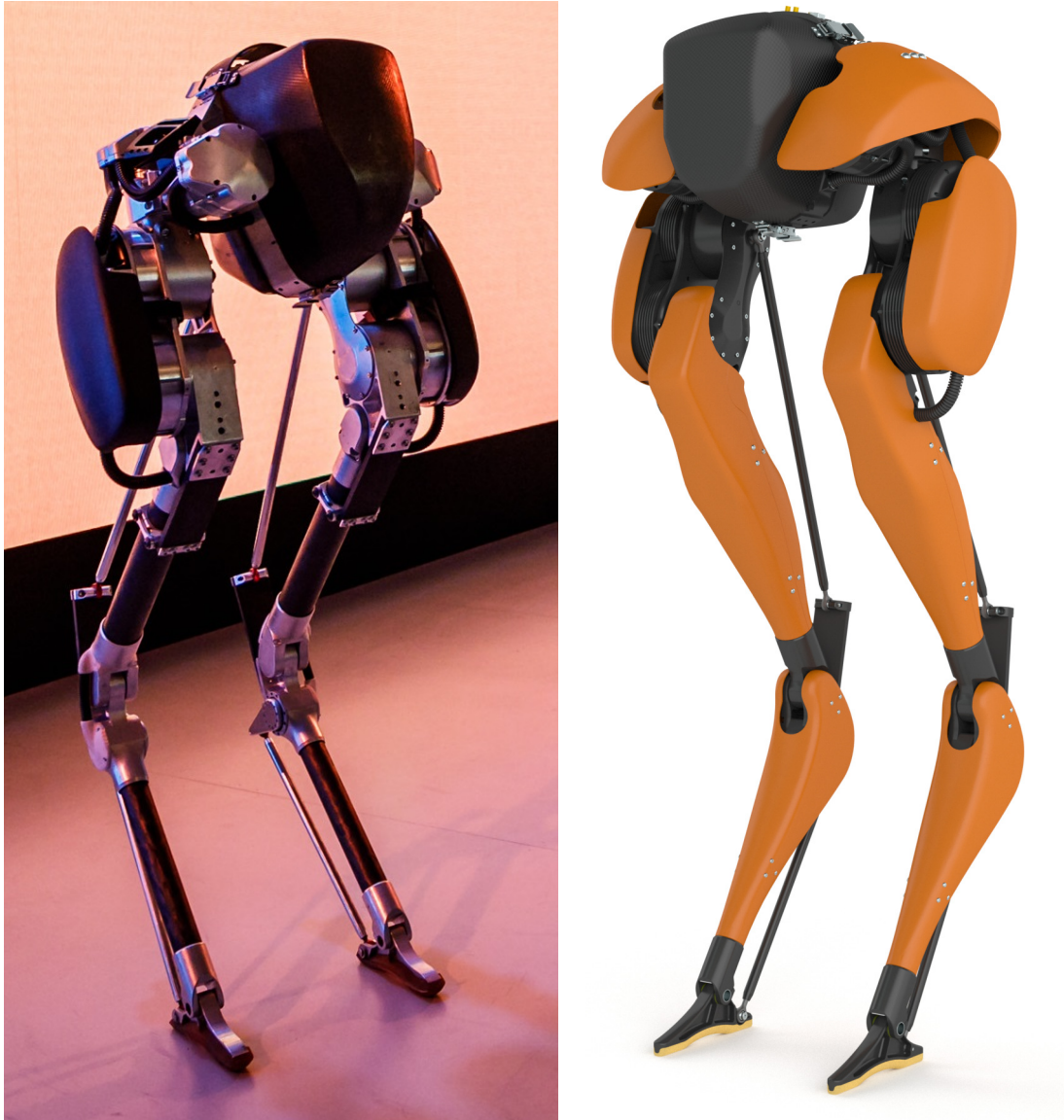


Figure 9: The OSU Cassie biped has abduction, long-axis rotation, hip, knee, and toe degrees of freedom. The knee and ankle have tuned series-compliant elements for efficient and robust walking.

of compliance, so a second spring is attached farther down the leg on an otherwise passive joint.

Making mechanically-centered design choices naturally made Cassie's kinematics and dynamics more complicated, but not insurmountable. The knee spring joint is not concentric with the knee motor, it is displaced distally to allow it to be connected with a plain bushing rather than heavier bearings. Also, the heel spring acts as a link in the pantograph mechanism and deflects under load. These two features make Cassie's pantograph mechanism act more like a six-bar linkage than a four-bar (Figure 10). Controllers can either neglect these effects, or they can apply loop-closure constraints to the full leg model.

ATRIAS had a serious mechanical flaw that was fixed in Cassie: an antagonistic work loop during walking and running gaits [3]. ATRIAS's parallel mechanism required the two leg motors to exert equal and opposite torque to lift the weight of the robot, while each motor moves in the same direction to sweep the leg underneath the body (Figure 11). Opposite torques and equal velocities results in equal-and-opposite power for the two leg motors, even for nominal spring-mass gaits that require zero net power. This antagonistic work loop unnecessarily dissipates power in the braking motor, and it increases motor power requirements.

Cassie uses serial leg motors to avoid antagonistic work for walking and running. Since the hip strictly swings the leg, it is the only motor that has to move during locomotion, while the knee just has to support the robot's weight at zero velocity. As such, for ideal spring-mass gaits, neither motor is doing any work, and the gait ideally requires no work, so there is no waste.

Cassie uses actuators optimized for efficiently performing walking gaits [122, 127]. ATRIAS used high-reduction 50:1 transmissions and 170 mm diameter pancake motors, resulting in substantial reflected inertia. With such high inertia, ATRIAS had to do a lot of work simply to swing the leg back and forth. Cassie uses an optimal combination of gear reduction and motor frame size to balance mechanical work output, resistive heating, and work to accelerate the reflected inertia.

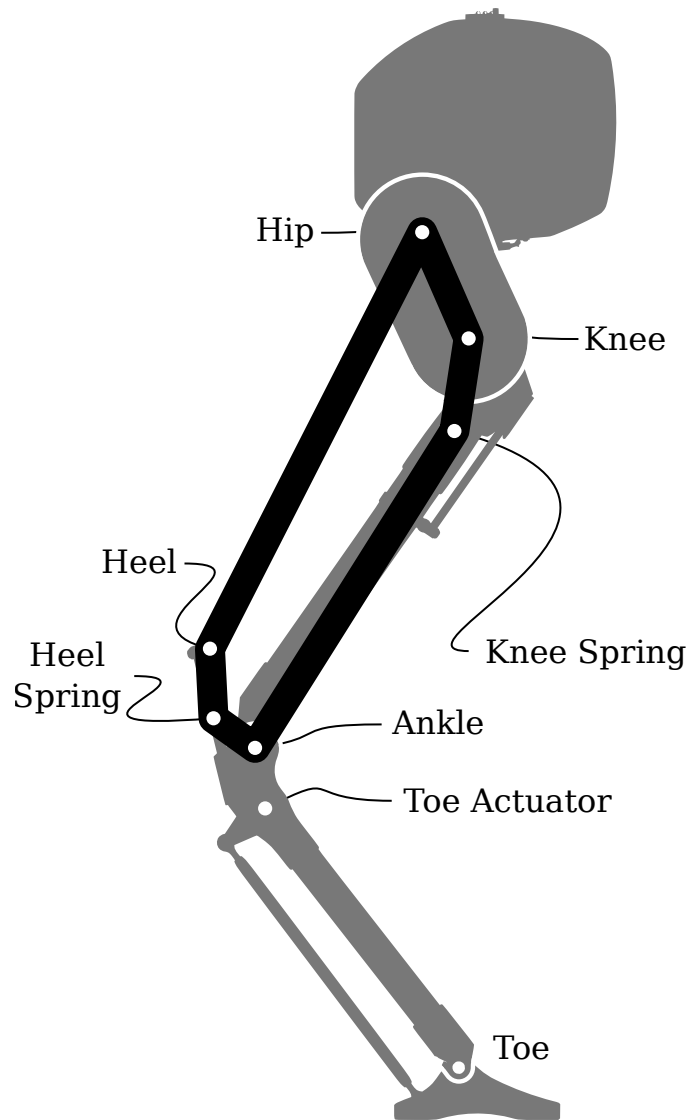


Figure 10: Cassie's two series springs are modeled as a planar six-bar linkage (black). The rigid mechanism moves as a four-bar connecting the hip, knee, ankle, and heel. The plate spring at the heel deflects with an approximate point of flexure along the spring (point labeled as "heel spring").

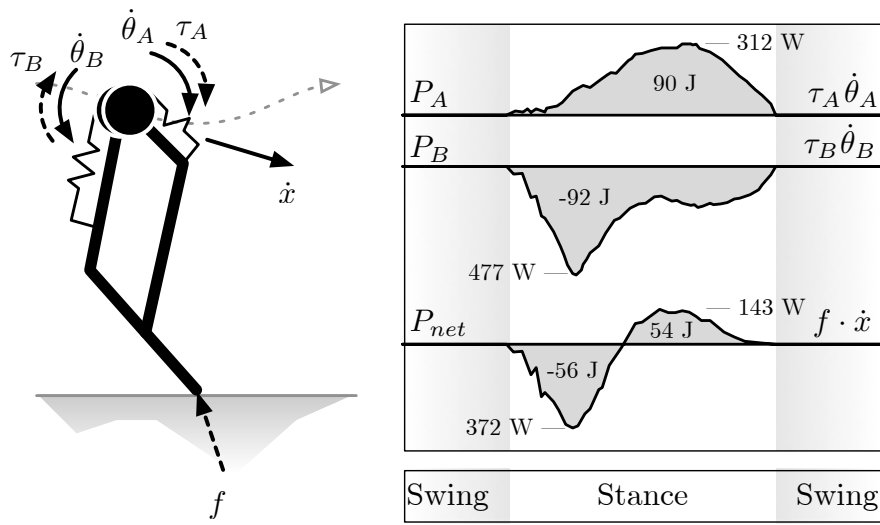


Figure 11: Motor power requirements for the ATRIAS robot during a walking task. The parallel mechanism decomposes leg-length forces into equal and opposite motor torques while the motors have equal velocity to sweep the leg underneath the torso. This results in the motors doing equal and opposite work during walking, dramatically reducing the efficiency of the robot and increasing peak power requirements above what is actually required by the task. Cassie was designed to correct this issue while maintaining simple passive dynamics.

3 MODELING VIRTUAL PROTOTYPES

All design processes require testing and iteration in order to discover and "hone" a good design. Building designs and testing them in the real world has the advantage of providing ground truths, but we do not always have the resources to do so. Certain tests require a real robot (strength and lifetime testing usually sacrifice real components), but often we can use mathematical models and simulation environments to test a "virtual prototype". It is often said that simulations are doomed to succeed, so we need to pay close attention to the models we use to simulate virtual prototypes. The models we use obviously have to represent reality to a certain extent, so the behavior of the simulated robot reflects the behavior of the robot as-built.

3.1 SOFTWARE, HARDWARE, AND THE WORLD

Locomotion exists as a dynamical process within the interactions between three dynamic subsystems: software algorithms, hardware mechanics, and world dynamics. Each component has its own internal state, laws governing its dynamical evolution, and interfaces through which they can effect change in the others (Figure 12). This work focuses on the rigid-body mechanics of a robot and touches on the software interface and contact mechanics in order to make informed, contextual mechanical design decisions. Classically, hardware and world dynamics are lumped together into the "plant" which interacts with the controller. However, since mechanical designers have quite a lot of control over the dynamics of a new robot's hardware, but they have relatively little control over what happens in the real world, there is a useful distinction between hardware and world dynamics.

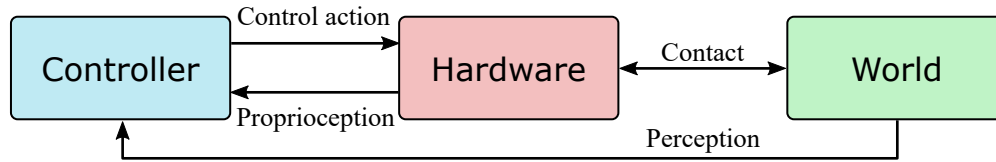


Figure 12: Hardware, software, world interaction

HARDWARE Hardware dynamics are governed by the laws of rigid-body mechanics, receiving internal forcing from the control system and external forcing from the world. Locomotion hardware typically does not have thrusters (although flying humanoids with hand and foot thrusters would be spectacular), so the control system has no means of affecting the robot’s linear and angular momentum directly. Only contact force with the world can produce changes in total momenta, so the control system must excite the desired contact forces.

CONTROL The control system takes sensor inputs, mixes it with some internal state, and produces torque commands for the robot’s active joints. The controller has no direct control over the world, but it can act indirectly through the robot’s hardware. Software can measure aspects of the world’s state through the robot’s sensors, but that has little bearing on hardware design so it will not be covered here. This work is concerned with potential control action on the hardware’s dynamics, not the implementation of specific algorithms.

THE WORLD Finally, world dynamics encompass everything happening outside of the robot: from the contact mechanics between a foot and the ground, to the behavior of a box that needs to be carried, or even an approaching vehicle that needs to be avoided. This work is concerned with the effect of external forcing on the robot’s hardware, not any aspects of mapping, localization, or planning.

3.2 CORE MECHANICAL SYSTEM

To build a virtual model of a robot, we must encode links and inertia, joint degrees of freedom, and reflected inertia as a multibody system (Figure 13 a). Multibody dynamics are mostly concerned with the inertial behavior of a system, since actuator forces, spring forces, damping, and friction can all be applied as forcing functions to the basic inertial dynamics. For this work, let us restrict members of the multibody system to be rigid (no soft bodies). Rigid bodies are *far* easier to model mathematically [38–41], and they are a reasonable representation of most real robotic links (excepting soft robots). Rigid body motion has been thoroughly studied and we have not only simple governing equations (Newton-Euler equations of motion), but we have sophisticated algorithms for quickly simulating arbitrary assemblies [43, 44, 80]. Keep in mind, though, that the mathematical concept of a rigid body is an approximation of real bodies; there is no such thing as a *real* rigid body, and all materials will bend and deform according to continuum mechanics when sufficiently stressed.

The most basic step in modeling a multibody system is drawing a system boundary: the imaginary envelope which distinguishes "robot" from "not robot" (Figure 13 b). The inscribed system has basic quantities associated with it: a center of mass (CoM), a total mass, and linear and angular momenta. Forces crossing the system boundary (the robot interacting with the world) modulate the system's momenta by definition. Forces within the system boundary (joint torques, for example) do not affect the system's momenta.

Robot topologies are constructed by connecting rigid bodies together with joints. The choice and number of joints (rotary, prismatic, multi-DoF) create the robot's configuration space, assigning numerical coordinates to poses and shapes the robot can take in the world. Joints do not always add degrees of freedom to the robot, since a closed loop of connected bodies has some joints acting as mobilizers and some joints acting as constraints. Robots with closed-chain topologies have both dependent and independent joints (which joints are which is a modeling choice), but only the independent joints form the configuration space. Constructing a model in this way allows us to express its kinematics and dynamics in *minimal* coordinates: the positions of the independent joints.

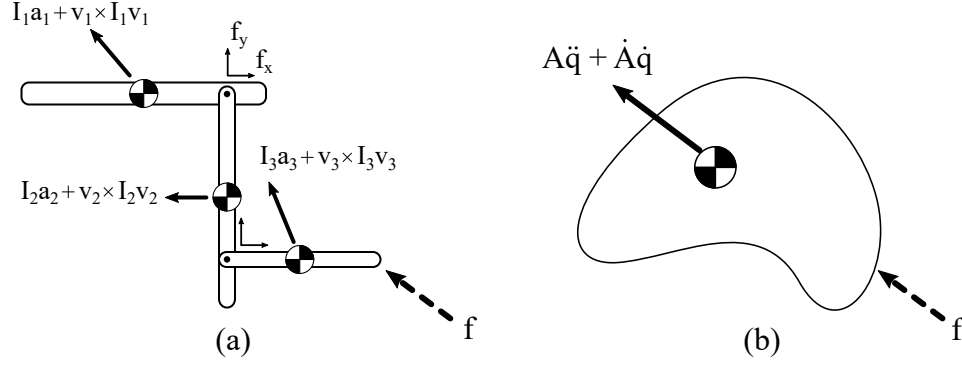


Figure 13: (a) shows a multibody system with an externally-applied force, link accelerations, and joint forces. By drawing a system boundary, (b) we can capture the relationship between external forces and gross system motion at/around the centroid. See [104, 105].

Computing kinematic quantities for a chosen topology is a matter of mapping position, velocity, acceleration, and force vectors between joint coordinates ($\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\tau} \in \mathbb{R}^n$) and world coordinates ($\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \mathbf{f} \in \mathbb{R}^m$). For any joint velocity of a manipulator, there is an associated end-effector velocity $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$. For any load applied to the end-effector, there is an associated set of joint torques $\boldsymbol{\tau} = \mathbf{J}^\top \mathbf{f}$. Such mappings are usually uniquely invertible if the dimension of the joint and task spaces are the same ($n = m$), but they are often underdetermined for articulated mechanisms. The Jacobian matrix $\mathbf{J}(\mathbf{q})$ can be interpreted as a row of task-space basis vectors for the joint variables q_i . These basis vectors \mathbf{e}_i are the partial derivative of the function mapping joint space to task space $\mathbf{x}(\mathbf{q})$.

$$\dot{\mathbf{x}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\partial \mathbf{x}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad \text{task-space velocity} \quad (1)$$

$$= \sum \frac{\partial \mathbf{x}}{\partial q_i} \dot{q}_i \quad \text{partial derivatives} \quad (2)$$

$$= \sum \mathbf{e}_i \dot{q}_i \quad \text{joint basis vectors} \quad (3)$$

These basis vectors are equivalently used to map task-space forces into joint space

$$\tau^\top \dot{q} = f^\top \dot{x} \quad \text{conservation of power} \quad (4)$$

$$\tau^\top \dot{q} = f^\top J \dot{q} \quad \text{replace task velocity} \quad (5)$$

$$\tau^\top = f^\top J \quad \text{must be valid for all } \dot{q} \quad (6)$$

$$\tau_i = e_i \cdot f \quad \text{joint torque as a scalar product} \quad (7)$$

where the torque applied to each joint in the task space is equivalent to the scalar product between that joint's basis vector and the applied task force. The basis-vector interpretation of the Jacobian matrix is used in Section 5 to derive visual representations of locomotion metrics.

This work models a robot's mechanical system using standard joint-space dynamic equations,

$$M(q)\ddot{q} + h(q, \dot{q}) = Bu + J(q)^\top f, \quad (8)$$

where M is the joint-space inertia matrix relating joint accelerations \ddot{q} to joint forces τ . h is the vector of velocity-product and gravitational forces ("bias" forces). B is a selector matrix mapping control action u into joint space. Unless otherwise stated, joint coordinates will always be those of the articulated body tree and not the input-side torque and velocity of actuators. So, actuator output torques in u will directly affect joint-space dynamics, and the B matrix will only have unit and zero entries.

Dynamics equations are written symbolically, but that is not to say that the terms must be symbolic. Lagrangian [15] and Hamiltonian mechanics work well for simple systems, generating symbolic equations of motion in minimal coordinates either "by hand" or through a computer-algebra system (e.g. MATLAB Symbolic Toolbox or Python's SymPy). Alternatively, systems with more complex dynamics and constraints are amenable to *numerical* dynamics (see Featherstone's book on dynamics algorithms [42]). Many simulation packages exist for calculating kinematic and dynamical properties such as the MATLAB Robotic Systems Toolkit, the Rigid Body Dynamics Library (RBDL [44]),

MuJoCo [145], and so on. There are also many special-purpose geometric, symplectic, and Lie-group integrators for generating accurate system trajectories [87, 110].

It is easiest to calculate the terms in the joint-space dynamics equation for a tree topology, but not all robots are open chains. To include constraints, we start with an open-chain system and include Lagrange Multipliers λ as constraint forces as well as the restriction that the constraint's position, velocity, and acceleration must be zero,

$$M\ddot{q} + h = Bu + J^T f + G^T \lambda \quad \text{dynamics with constraint forces} \quad (9)$$

$$g(q) = 0 \quad \text{constraint position} \quad (10)$$

$$\dot{g} = G\dot{q} = 0 \quad \text{constraint velocity} \quad (11)$$

$$\ddot{g} = G\ddot{q} + \dot{G}\dot{q} = 0 \quad \text{constraint acceleration} \quad (12)$$

where G is the Jacobian of the holonomic constraint function $g(q)$. q and \dot{q} must be initialized so as to satisfy Equations 10 and 11. Terms in this equation can be found symbolically or by using the Composite Rigid Body Algorithm to find M and the Recursive Newton-Euler Algorithm to find h and $J^T f$. Solving the constrained system can be done by formulating a system of linear equations in \ddot{q} and λ ,

$$\begin{bmatrix} M & -G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Bu - h + J^T f \\ -\dot{G}\dot{q} - \alpha g - \beta \dot{g} \end{bmatrix}, \quad (13)$$

where α and β are Baumgarte stabilization parameters which drive $g(q)$ to zero when the robot's configuration does not completely satisfy the constraints [45].

Alternatively, we can project the open-chain dynamics onto the constrained dynamics to yield a system in minimal coordinates. The projection approach yields a single dynamic equation that is constraint-agnostic

$$M_\gamma \ddot{q}_{\text{ind}} + h_\gamma = B_\gamma u + J_\gamma^T f. \quad (14)$$

We can arrive at this equation by finding a matrix γ such that $G\gamma = 0$. One way of generating γ is to partition G into columns associated with the independent joints q_{ind} and the dependent joints q_{dep} , then writing \dot{q} in terms of \dot{q}_{ind} alone

$$G\dot{q} = \begin{bmatrix} G_{\text{ind}} & G_{\text{dep}} \end{bmatrix} \begin{bmatrix} \dot{q}_{\text{ind}} \\ \dot{q}_{\text{dep}} \end{bmatrix} = 0 \implies \dot{q} = \begin{bmatrix} \dot{q}_{\text{ind}} \\ \dot{q}_{\text{dep}} \end{bmatrix} = \gamma \dot{q}_{\text{ind}}, \quad (15)$$

$$\text{where } \gamma = \begin{bmatrix} I \\ -G_{\text{dep}}^{-1} G_{\text{ind}} \end{bmatrix}. \quad (16)$$

So long as the constraints are not redundant, there will be as many dependent joints as constraint equations, so G_{dep} will be square and invertible. Another way of finding γ is to write a function mapping independent coordinates into full coordinates as in $q = y(q_{\text{ind}})$. γ is then simply $\partial y / \partial q_{\text{ind}}$ since $\dot{q} = \dot{y} = \gamma \dot{q}_{\text{ind}}$.

We can use γ^\top to project the unconstrained dynamics into the constrained dynamics (removing the Lagrange multipliers) as in

$$\gamma^\top M \ddot{q} + \gamma^\top h = \gamma^\top B u + \gamma^\top J^\top f + \gamma^\top G^\top \lambda \quad (17)$$

$$= \gamma^\top B u + \gamma^\top J^\top f + 0. \quad (18)$$

where $G\gamma = \gamma^\top G^\top = 0$, which is proven simply as

$$G\gamma = \begin{bmatrix} G_{\text{ind}} & G_{\text{dep}} \end{bmatrix} \begin{bmatrix} I \\ -G_{\text{dep}}^{-1} G_{\text{ind}} \end{bmatrix} = G_{\text{ind}} - G_{\text{ind}} = 0. \quad (19)$$

We can cast \ddot{q} in terms of \ddot{q}_{ind} by seeing that

$$\ddot{g} = G\ddot{q} + \dot{G}\dot{q} = 0 \implies \begin{bmatrix} G_{\text{ind}} & G_{\text{dep}} \end{bmatrix} \begin{bmatrix} \ddot{q}_{\text{ind}} \\ \ddot{q}_{\text{dep}} \end{bmatrix} + \dot{G}\dot{q} = 0 \implies \quad (20)$$

$$\ddot{q} = \begin{bmatrix} \ddot{q}_{\text{ind}} \\ \ddot{q}_{\text{dep}} \end{bmatrix} = \begin{bmatrix} I \\ -G_{\text{dep}}^{-1} G_{\text{ind}} \end{bmatrix} \ddot{q}_{\text{ind}} + \begin{bmatrix} 0 \\ -G_{\text{dep}}^{-1} \end{bmatrix} \dot{G}\dot{q} = \gamma \ddot{q}_{\text{ind}} + \dot{\gamma} \dot{q}_{\text{ind}}. \quad (21)$$

$\dot{G}\dot{q}$ is a vector which is self-explanatory to calculate symbolically, but it may be more intimidating when using dynamics algorithms. The knee-jerk reaction may be to use finite-differencing on G , but evaluating the constraint acceleration function $\ddot{g}(q, \dot{q}, \ddot{q})$ with $\ddot{q} = 0$ returns the vector $\dot{G}\dot{q}$.

Combining Equations 18 and 21 produces

$$\gamma^\top M \gamma \ddot{q}_{\text{ind}} + \gamma^\top M \dot{\gamma} \dot{q}_{\text{ind}} + \gamma^\top h = \gamma^\top B u + \gamma^\top J^\top f, \quad (22)$$

giving us the terms for the projected dynamic equation

$$M_\gamma = \gamma^\top M \gamma, \quad \text{constrained inertia} \quad (23)$$

$$h_\gamma = \gamma^\top (M \dot{\gamma} \dot{q}_{\text{ind}} + h), \quad \text{constrained bias force} \quad (24)$$

$$B_\gamma = \gamma^\top B, \quad \text{constrained actuator matrix} \quad (25)$$

$$J_\gamma = J \gamma, \quad \text{constrained Jacobian} \quad (26)$$

$$M_\gamma \ddot{q}_{\text{ind}} + h_\gamma = B_\gamma u + J_\gamma^\top f. \quad (27)$$

The rest of this text discards the γ subscript and treats both constrained and unconstrained dynamics with the same equation, simply requiring that if a design contains constraints that its dynamics be projected into the constrained subspace.

3.3 CONTROL INTERFACE

It is the control system's job to excite motion and generate locomotion gaits, moving the robot through the world. Hardware dynamics and the control system have been deliberately partitioned to allow for passive mechanical components to be included in the "control system", since their behavior is very much like a simple, fixed controller acting on the joints. This reduces the complexity of the hardware's dynamical equations and reinforces the use of "mechanical intelligence" [21].

The control system interacts with the hardware by generating internal forces using active mechanical actuators and passive components like springs and dampers. This work focuses

on electrical actuators (brushless DC motors with gearheads) and joint-level passive compliance. Including actuators in a mechanical system introduces acceleration-proportional force at the joint (reflected inertia), and including springs introduces deflection-proportional force. Passive compliance is nearly lossless, but generating torque with actuators carries with it an energy-loss model.

3.3.1 Actuators

Actuators give the control system a channel through which to apply arbitrary forces (within saturation limits) to the mechanical system. Adding actuators comes at a cost, however: not only do they increase the weight of the system, but they add reflected inertia and require power to exert forces. They are required for machines that do work and adapt to their environment, but in some cases passive components are more appropriate.

Electric actuators have a relatively simple loss model, requiring electrical power to be converted into mechanical work (that which is necessary to accelerate the reflected and link inertia, resist viscous friction, and do work on the world) and also to generate holding torque.

$$v = k_b \dot{\theta} + iR \quad \text{winding voltage} \quad (28)$$

$$i = \tau_b / k_b \quad \text{winding current} \quad (29)$$

$$\tau_b = \tau + I\ddot{\theta} + c\dot{\theta} \quad \text{magnetic torque} \quad (30)$$

$$P_{\text{elec}} = iv = \tau_b \dot{\theta} + \tau_b^2 / k_W^2 \quad \text{electric power} \quad (31)$$

$$= \tau \dot{\theta} + I\ddot{\theta} \dot{\theta} + (\tau + I\ddot{\theta} + c\dot{\theta})^2 / k_W^2 \quad (32)$$

where k_b is the winding constant (identically the torque and back-EMF constants) of the motor and k_W is the motor constant relating torque to winding losses in resistive heating.

τ is the actuator's output torque, $\dot{\theta}$ and $\ddot{\theta}$ are the actuator's speed and acceleration, and I is the inertia of the actuator. I and k_W for a motor/gearhead pair are calculated simply,

$$I = I_{ls} + G^2 I_{hs} \quad (33)$$

$$c = c_{ls} + G^2 c_{hs} \quad (34)$$

$$k_W = G k_m \quad (35)$$

where G is the gearhead reduction factor, I_{ls} is the inertia of the low-speed side of the actuator (output, typically), I_{hs} is the inertia of the high-speed side (motor rotor), c is viscous friction, and k_m is the motor constant.

Geared motors introduce reflected inertia at the actuated joint, increasing that DoF's resistance to acceleration. This is not a spatial inertia that resists body acceleration, but strictly acts on the relative acceleration between two links connected by an actuator. Adding this reflected inertia to the mechanical model is as simple as appending an I_i to the diagonal element of M indexed by the joint to which the actuator i is attached (an actuator on the second joint of a 2-link serial manipulator would append $M_{2,2}$).

Actuators are treated as torque sources at chosen joints in an articulated rigid-body system. Since we have discussed loop-closure constraints, there is no reason for actuators to "bridge" multiple joints, since the constrained B_γ matrix will account for any multi-articular effort.

3.3.2 Passive compliance

Springs are best thought of as fixed proportional controllers on a robot's joints. The advantage of this over active compliance is that passive compliance does not have any energetic costs and no reflected inertia or other actuator dynamics (backlash, friction, etc). So, if part of the software control system would only *ever* generate a fixed force/deflection profile at a joint, it is better to use a physical spring at that joint. Springs introduce a predefined forcing function on the joints as a function of joint deflection. They can be

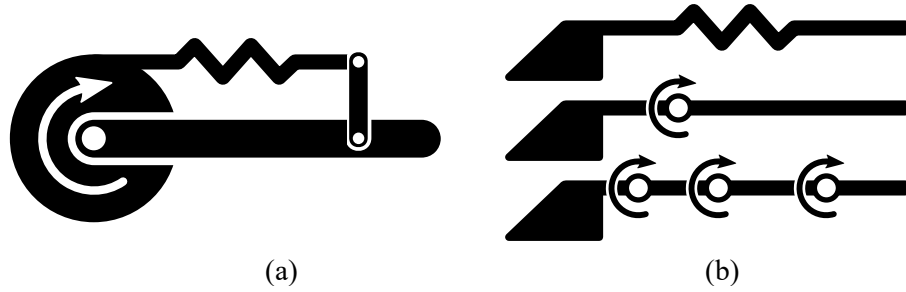


Figure 14: (a) uses nonlinear cantilever-beam deflection and a link to get approximately linear torque/deflection response around the compliant joint. (b) approximates the large-deflection beam behavior using one or more joints with linear spring functions.

in series or parallel with motors [46], and they can be mono- or multi-articular through loop-closure constraints.

The Cassie and ATRIAS robots use cantilevered fiberglass plate springs, posing an interesting modeling problem. Both ATRIAS and Cassie have spring linkages which make the cantilevered spring behave like a proportional spring around the compliant joint (Figure 14 a), but Cassie also uses a cantilever spring as a link with no fixed flexure joint (Figure 14 b). This may seem to introduce modeling difficulty, but there is a trick: such cantilever springs can be approximated as a series of rigid links with proportional springs at the joints [142].

We can identify the spring rate of compliant joints in the real robot by using a fitting process. One option is to identify the spring rate of each joint individually, and another is to identify all compliant joints at once by correlating force/deflection response of the end effector in task space. When using the multi-link approximation of a cantilever beam, the distances between flexure points are also open variables that need to be discovered based on test data. The test samples joint positions and contact forces (using a forceplate or load cell) for a range of feed-forward motor torques. The fitting process tries to find linear spring rates and flexure positions in order to minimize the error between the joint torque induced by the measured contact force and the joint torque modeled by the deflection of joint compliance.

3.4 WORLD INTERACTION

Since locomotors do not have thrusters, contact is the only source of controlled motion through the world. Unfortunately, we generally have no control over ground composition in the real world, and contact in general presents substantial modeling difficulty. However, we can still make useful mechanical design decisions by looking at the contact interface and leaving the contact model as a black box, permitting variable ground composition and friction. For example, we can devise control systems that are insensitive to touchdown timing and ground composition [62, 124].

Contact can be modeled as a discrete impact event followed by continuous unilateral forcing [138]. Before contact, the robot's end effector and the world have a relative velocity and a non-zero separation distance. At the moment of impact (separation reaches zero), each body experiences an impulse to bring their relative velocity to zero. The duration of contact (compression/restitution) occurs in continuous time, applying non-zero external force to the robot according to the contact model (rigid, soft [144], deformable, etc). Contact terminates at the moment when applied force crosses zero[†].

Contact force on a body is a 6-element spatial force (force and moment) [40, 41] which typically does not allow a Center of Pressure outside the contact area. Some sort of adhesion would be necessary between the end effector and the world to allow the CoP to be outside the contact area[‡]. The spatial contact force can be constructed using the sum of linear contact forces at different points on the contacting body. This is useful for robots like Cassie which have line-contact feet [1], since the zero-roll-torque constraint becomes implicit due to only two contact points on the foot.

The real robot must be built and controlled to permit uncertain contact models and touchdown timings (sand, mud, loose rubble, ice, etc). The only thing we generally know

[†] In general, the correct way to handle the contact state is to establish contact based on *distance* and to release based on *force* [70]. This prevents the accidental inclusion of adhesion effects where a velocity-dependent force balances a compression-dependent force before the compression distance becomes zero.

[‡] The Center of Pressure is really the intersection of the "line" aspect of a spatial contact force with a "support polygon". Support area makes less sense when hands and feet are used on terrain with changing or uncertain impedance. It makes even less sense when hands are allowed to grip the environment, presenting bilateral forcing. Dynamic locomotion is only concerned with the net contact force, not its intersection with an arbitrary polygon.

beforehand is that the contact state will "flow" in the direction of commanded contact force. That is, the robot's limbs will extend when the commanded force cannot be matched by the current state of contact. Rigid contact makes this instantaneous, supplying the restoring force immediately, but granular media and slippery surfaces will have some delay (or may not be able to support the commanded contact force due to friction or other limitations). Shaping this behavior through mechanical design allows us to use control schemes which "fail gracefully" when they encounter unexpected circumstances. In practice, friction cones are used more to keep contact forces within some safety margin, not to actually model contact friction.

Terrestrial systems are subject to gravity, but those systems only *feel* gravity when they resist it. It is the contact force, then, which couples the effect of gravity to acceleration of the robot's joints[†]. This fact presents many interesting opportunities for control, such as "gravity compensation" which accounts not just for a vertical 9.81 m/s acceleration, but also virtual forces resulting from non-inertial reference frames (e.g. traveling along a winding road in the back of a truck).

3.4.1 Touchdown event

For any given mechanism, we know the relationship between task impulse and task velocity delta as

$$H = J^T \Lambda \quad \text{joint-space impulse} \quad (36)$$

$$\Delta \dot{q} = M^{-1} H = M^{-1} J^T \Lambda \quad \text{joint-space velocity delta} \quad (37)$$

$$\Delta \dot{x} = J \Delta \dot{q} = J M^{-1} J^T \Lambda \quad \text{task-space velocity delta} \quad (38)$$

where $(J M^{-1} J^T)^{-1}$ (which can also be calculated using dynamics algorithms [155]) is the Operational Space Inertia Matrix (OSIM) [14] and Λ is the task-space impulse. This matrix (and the impulse-velocity relationship it represents) can be visualized as an ellipsoid in

[†] For example, arm swing is driven by ground-reaction force on the feet, and forceful arm motions spike the ground-reaction force.

task space, where major diameters represent the "heavy" directions and minor diameters represent the "light" directions.

3.4.2 Continuous forcing

Rigid ground implies a motion constraint that allows us to solve directly for contact force

$$\mathbf{x}(q) \quad \text{world position of contact} \quad (39)$$

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} = 0 \quad \text{contact velocity constraint} \quad (40)$$

$$\ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} = 0 \quad \text{contact acceleration constraint} \quad (41)$$

$$\mathbf{J}^\top \mathbf{f} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} - \mathbf{B}\mathbf{u} \quad \text{equation of motion} \quad (42)$$

$$\mathbf{M}^{-1}\mathbf{J}^\top \mathbf{f} = \ddot{\mathbf{q}} + \mathbf{M}^{-1}(\mathbf{h} - \mathbf{B}\mathbf{u}) \quad \text{invert mass} \quad (43)$$

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\top \mathbf{f} = \mathbf{J}\ddot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{h} - \mathbf{B}\mathbf{u}) \quad \text{left-multiply Jacobian} \quad (44)$$

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\top \mathbf{f} = -\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{h} - \mathbf{B}\mathbf{u}) \quad \mathbf{J}\ddot{\mathbf{q}} = -\dot{\mathbf{J}}\dot{\mathbf{q}} \quad (45)$$

$$\mathbf{f} = \left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^\top\right)^{-1} [\mathbf{J}\mathbf{M}^{-1}(\mathbf{h} - \mathbf{B}\mathbf{u}) - \dot{\mathbf{J}}\dot{\mathbf{q}}] \quad \text{rigid contact force} \quad (46)$$

Alternatively, we can use an augmented inertia matrix to simultaneously solve for joint acceleration and contact force

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^\top \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{B}\mathbf{u} - \mathbf{h} \\ -\dot{\mathbf{J}}\dot{\mathbf{q}} - \alpha(\mathbf{x} - \mathbf{x}_0) - \beta\dot{\mathbf{x}} \end{bmatrix}, \quad (47)$$

where the Baumgarte stabilization parameters α and β are necessary to stabilize contact distance and velocity during simulation.

4 DEFINING THE LOCOMOTION TASK

Our design process needs a precise definition of what candidate designs are supposed to *do*. With a clear definition of purpose, we can design the hardware to ensure the feasibility of a robot, increase efficiency, and reduce the stability burden placed on the control system. Additionally, specifying a task allows us to pick appropriate objectives for guiding our search through the design space.

4.1 ABSTRACT LOCOMOTION

In general, the purpose of a locomotor is to transport itself from one location to another over potentially rough or hazardous terrain. So, let us choose to initially define locomotion as the self-transport of a system of connected bodies. Self-transport means that the system should be able to move through space without the assistance of external devices. Everything not connected to the system is considered as "the world".

In order to transport its own mass through space, the locomotor system has to adjust its linear momentum. Simply, speeding up requires an impulse in the desired direction and slowing down requires the opposite impulse. Additionally, we can say that locomotion is associated with zero average angular momentum, since any undulation is periodic and not continuous (barring acrobatics). Only forces acting across the system boundary can affect the system's momentum, since forces within the system simply exchange momentum between components and will not affect the total.

For terrestrial locomotion, the system will have no propellers or thrusters, so the only source of controlled impulse is the force of contact with the world. This allows us to say that **terrestrial locomotion is ultimately the act of exciting contact forces and impulses.**

This statement is made with no loss of generality relative to the original definition of mass self-transport.

Such a definition may appear to neglect attitude adjustment during free motion (no contact), but it is often necessary to reorient and reconfigure the robot in order to apply desired contact forces in the future. Reorientation takes advantage of the conservation of angular momentum to reorient the system rotationally [88]. Total angular momentum still can only be adjusted through contact force.

As a side note, controlling contact force is different than merely being able to *resist* contact forces; a chair's legs are in contact with the ground, and the combined impulse of their contact perfectly balances the impulse of gravity, but that does not make it a good locomotor. The difference lies in whether the commanded actuator torque resists applied loads or if it is the acceleration or compression/tension of passive components; torque due to magnetic flux in the air gap of an electric motor is controllable whereas torque due to other effects is not. Robots can gain efficiency by using passive components where possible and using active components only where control and adaptability are necessary. Robots can also exploit singularities (which are typically avoided because they limit the robot's control authority) to resist contact forces rather than spending energy on motor effort.

4.1.1 *Abstract control structure*

In order to control its motion, a system must be able to excite contact forces (the system's only source of impulse). By definition of the mass-center of a system, both controlling momentum and controlling contact force are equivalent to controlling the acceleration (or change in velocity) of the system's center of mass. It follows that, in general, the control system of a locomotor is interested in commanding both linear and angular centroidal (center of mass) accelerations.

Since locomotion is ultimately the combination of exciting contact forces and reconfiguring the robot's pose, "locomotion-level" commands can be sent to an interface layer which abstracts from the specific hardware design. A fair amount of research has been

done on such an interface as "whole-body control" [71, 105], "task-space control" [28, 89], or "operational-space control" [76]. Such control interfaces take acceleration and force commands for the centroid or other points on the robot and deduce the joint torques necessary to produce those accelerations given ground contact, friction cones, and other constraints. Whole-body locomotion control is only concerned with the CoM motion task, leaving room in the controlled dynamics for other tasks (e.g. manipulation, acrobatics, body language communication, etc). One could imagine this interface as a more sophisticated version of Jacobian-Transpose control [24].

In operation, the acceleration control interface abstracts higher-level controllers from the kinematics/dynamics of the hardware. Effects of hardware changes are bounded to this interface layer and do not require changes to the higher-level controllers acting on this layer. For example, the trajectory generator and path planner can operate on the reduced point-mass dynamics of the robot's CoM without needing to know anything about the robot's internal dynamics [33].

4.2 SPRING-MASS LOCOMOTION

Abstract locomotion places few limits on the dynamics of a locomotor, so we can use a specific model to usefully constrain the system's dynamics. At this point, locomotion only involves the motion of the CoM and the net contact force (i.e. CoP). Arguably, the simplest dynamic constraint is that the contact force is controlled to increase as the CoM gets closer to a desired contact point within reach of the system. This causes the system to "bounce" off of any surfaces with which it comes into contact (imagine potential-field obstacle avoidance [75]). The Spring-Loaded Inverted Pendulum (SLIP, Figure 15) [20] codifies these "spring-mass" dynamics.

There is substantial evidence that spring-mass locomotion has benefits in terms of energy economy [4, 125], self-stability [49], and impact robustness. The SLIP and other reduced-order models give insight into how robots may exploit natural dynamics for

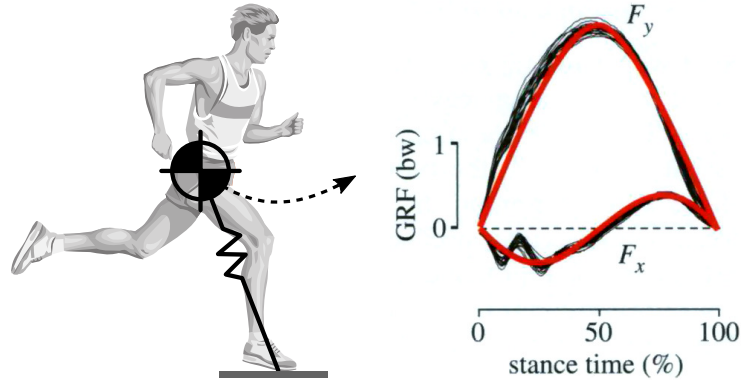


Figure 15: Spring-loaded inverted pendulum (SLIP) model of a human running gait. This minimal spring-mass model has been shown to produce both walking and running gaits for humans, as well as realistically model a number of terrestrial animals. Adapted from [50, 131].

locomotion. By designing a robot to *dynamically* resemble a SLIP, it can receive the same benefits and use similar control theories as the SLIP [63].

4.2.1 Reduced-order model

The SLIP can be seen as a simple, lumped-parameter representation of animal limbs. The SLIP comprises a single point-mass body and a massless linear spring for each leg. Being composed of only a point mass and springs, the SLIP is conservative and has no rotational inertia or angular momentum, and the only control signal for the unactuated SLIP is the leg angle at touchdown. This is not the only reduced-order model for locomotion [34, 59, 116, 130, 135], but it is arguably the most applicable [20, 96]. Additionally, there are quite a few locomotion control ideas stemming from simple leg angle control of the SLIP [11, 13, 37, 78, 79, 91, 112, 134, 148, 149, 157, 160]. SLIP dynamics require numerical integration (e.g. [128]), but there also exist closed-form approximations [131, 136].

Amazingly, the simple, unactuated, bipedal SLIP reproduces the CoM trajectory and ground-reaction forces of human walking and running [49, 50]. However, even though it is analogous to the CoM and series compliance in animals, the SLIP says nothing about how animals use physical compliance. We know that animals do indeed incorporate and exploit compliance [7, 8, 10], and it is possible that animals actively track a reduced-order

model [19, 47], but we do not know exactly why. So, we need to remember that robots are not animals, and biomimicry design arguments are no better than gut instinct.

The SLIP has been shown to have a fragment of self-stability when being controlled with feed-forward touchdown leg angle [48]. That is, the SLIP can reject small disturbances *passively* without feedback control and converge to the original equilibrium gait. Self-stability suggests that real robots which instantiate SLIP dynamics will have a reduced control burden due to the passive rejection of disturbances.

4.2.2 *Application in design*

Mechanical designs can physically approximate SLIP dynamics and receive a number of the SLIP's desirable properties. Of course, a real robot must have distributed mass, so no design will ever perfectly match the SLIP, but much of the intuition still applies. Many implementations simply copy the SLIP's "pogo-stick" morphology, thereby trivially approximating SLIP dynamics [5, 12]. However, articulated mechanisms have benefits over linear slides [109], so other robots have adopted non-obvious morphologies with the same dynamic properties [63, 109]. Approximating SLIP is another great example of designing a machine with dynamics and control in mind [2].

Although compliance for spring-mass gaits could be implemented as a proportional controller in software [35, 73, 154], it is better to utilize physical compliance due to the inefficiencies and reflected inertia of motors. Utilizing physical springs as inertialess, efficient proportional controllers carries a number of benefits (and also a few detriments).

- + Resonant locomotion following spring-mass gaits can recycle gait energy using passive springs as a high-efficiency energy store [4]. These gaits have a known positive and negative work requirement over time, and tuned series compliance can do this work with very little loss as opposed to electric motors which do not easily regenerate energy and require power simply to resist forces.
- + Passive springs decouple actuator and body inertia from ground impact, reducing force spikes from unexpected contact. Softening impacts can increase the lifetime of gears,

electronics, and other components, and it also reduces the disturbance (discrete velocity change) associated with large impulses.

- + Leg compliance makes the ground contact model matter less because the known leg model dominates. This reduces perception and planning requirements, making the controller simpler.
- + Spring-mass gaits can reduce simultaneous positive and negative work if at least one actuator operates the leg angle exclusively [3, 8, 81].
- + Electric motors cannot do work at max torque, but series springs can. The motor only needs to hold the neutral position of the spring, and the spring alone will store and release energy [55].
- +/- Parallel springs help energy efficiency because the motors do not have to exert themselves, but they limit possible tasks because the motors must fight to maintain poses away from the springs' neutral positions. This is a specialization/functionality trade-off.
- Too much compliance between various inertias (reflected, heavy links, etc) can lead to several vibration modes if not designed correctly [83], making the robot difficult or impossible to control.
- Since the goal of locomotion is to excite contact forces, the combination of compliance and reflected inertia hurts controller bandwidth. Acceleration is now a function of position, and thus not directly controllable by the actuators. This can lead to ringing in position and force commands. However, reducing reflected inertia brings the series-elastic actuators [114] back to being simple torque sources, just like a plain motor.

Encoding these properties into a limb design is a matter of finding the right "program" for intrinsic compliance (deflection vs force at the end-effector) and the right inertia distribution. Even when the right theoretical spring rates are found, they may need to be nonlinear or otherwise difficult to translate the spring function into physical components.

4.3 THROUGH-CONTACT TRAJECTORY OPTIMIZATION

Walking, running, or other tasks can be defined by objectives and constraints for a dynamical system. Obviously, simulating the full robot with final controller gives the best indication of the operating point of a design, but we are often forced to create initial machines that will work amicably with unspecified, future controllers. If we have access to a through-contact trajectory optimizer [98, 99, 113, 121] as a *stand-in* for a controller, then we can generate candidate trajectories on a design-by-design basis to maximize goals. However, gait optimization is quite the subject on its own, so this work only discusses it for completeness.

Given a sufficiently realistic simulation, a computer optimization process could find very concrete designs that exploit kinematic singularities, have autonomous swing-phase dynamics, use toe-off forces for leg recirculation, and other passive-dynamic features for very specific tasks. There has been substantial work in this area, especially in locomotion [36, 57, 60, 61, 153].

Optimizers find the theoretically-optimal path given perfect information. In reality, the trajectories will surely be worse, requiring feedback control to stabilize [68]. However, we are after the peak theoretical performance of a hardware design, so we are not as worried about tracking.

In the world of optimal control, it is either possible to meet constraints or it is not possible (feasibility), and it is up to the objective function to determine which feasible trajectory is *best*. The objective must provide enough regularity to find feasible solutions, but it must also lead to the desired solutions. Minimizing the integral of squared torque is a common objective, giving smooth trajectories fairly regularly (minimizing nondimensional work is also common [121, 141]). Squared-torque happens to measure the deviation from passive-dynamic paths (trajectory "error" from those which require no control effort), and is also proportional to resistive-heating losses in BLDC motors. So, this metric minimizes a large component of the robot's power requirements in addition to providing smooth, least-action trajectories.

5 PERFORMANCE METRICS

Objectives guide our search for better locomotor designs by indicating how well we expect possible designs to function relative to each other. Even without specifically setting these down, designers intuitively use objectives and metrics to judge designs. For example, we are commonly interested in reducing weight and component cost while increasing load capacity (making the robot lighter, stronger, and cheaper). Meticulousness encourages us to lay out all the factors at play in our design program, so that is what we will do.

5.0.1 *Properties of good performance metrics*

In general, a metric can be any qualitative or quantitative evaluation of a *real property* of a robot which provides the ability to rank candidate designs. A metric is typically a well-ordered quantity (like Integers or Real numbers), but they can be any formal set of *weakly-ordered* objects[‡]. We require the set-theory definition of metrics when we begin to combine disparate metrics and identify over-all better designs in Section 6.

Metrics should indicate tangible, real-world performance, so there must be a mapping from real designs into the metric space. Without the connection to a specific, real property of the system, metrics just create noise in the design process. So, it is important to *calibrate* metrics against real properties. Additionally, if the chosen metrics for a project lead to sub-par results, then the project metrics need to be re-evaluated.

Metrics must be normalized in order to fairly compare robots of different weights, sizes, and speeds. For example, Jacobian-based performance metrics are susceptible to unit inhomogeneity when both linear and rotary actuators are used simultaneously [90]. Without homogenization, robots of different types are not comparable. Similarly, it is

[‡] Weak ordering means each element of a set is either less than, equal to, or greater than any other element of the set.

much more impressive for a tiny, 10 cm-long hexapod to travel at 1 m/s than it is for a 1.5 m cheetah, so simply measuring top speed is not a good metric. In most cases, a properly-normalized metric has no units.

If it is possible, bounded metrics provide upper and lower limits which give a sense of where a particular design falls in the spectrum of possibilities. Of course, all numbers are bounded by infinities, but that is the default. If we can bound metrics between meaningful upper and lower limits (or at least provide reference values), we can gain a better insight into the possible designs.

Metrics should also be smooth, if possible. Smoothness implies that a subtle change in the input design produces a subtle change in the metric. Discontinuities in the mapping between designs and their metric value can lead to confusion and make it difficult to hone into an optimal design. For example, mechanical linkages have singularities which have a large impact on manipulability metrics. As a general rule, metrics should be functional up to and through singularities.

5.0.2 *Locomotion metric categories*

When creating a new hardware design for locomotion, there are four categories of objectives that have been described by various groups and are generally agreed upon:

1. The energetic **efficiency** of a design in performing its tasks, measured by the drain on its power supply. Energy efficiency is critical for mobile robots, because they must carry their own power source.
2. The **control authority** of a design, measuring how "effectively" the control system can affect change in its control target. Classical manipulability metrics are placed here and assume force, velocity, or acceleration control targets of the end-effector.
3. The **robustness** of a design to control errors and external disturbances. Slips, impacts, and uncertainty cannot be allowed to threaten the stability of the gait. Some designs are less robust than others, so we need to quantify that in our design process.

4. The **feasibility** of a design, making sure it is physically capable of walking, running, jumping, and lifting according to the task description. These objectives are a subset of the design constraints which are written as penalty functions.

These categories are fairly abstract (perhaps except for efficiency), but we can dive into each one to find concrete design metrics which promote the higher-level goal.

5.1 EFFICIENCY METRICS

Mobile robots must strive to increase their efficiency, since they must carry their own power supply. Higher power requirements lead to larger motors and power sources, increasing the weight of the robot and leading to even higher power requirements; it is a vicious cycle. The trouble is, a machine alone does not have power requirements, only does the combination of a machine and a motion/torque trajectory. As such, we cannot optimize the efficiency of a machine without specifying how it will be moving and interacting with the world. The task specification can be as complex as a distribution of speed/torque trajectories, or it can simply be a pair of force/velocity vectors at the end effector.

Given that power generally cannot be regenerated once spent, the loss model for an electric actuator is the integral of positive electrical power [141],

$$\int P_+ = \int [\tau\dot{\theta} + I\ddot{\theta}]_+ + c\dot{\theta}^2 + (\tau + I\ddot{\theta} + c\dot{\theta})^2/k_W^2 \quad (48)$$

We want to minimize mechanical power expenditure, and limit costs related to generating torque/motion. Better regenerative braking will certainly increase the efficiency of gaits [100, 158], but it is always better to keep power requirements low than rely on regeneration. Even if power regeneration was 100% efficient, motors still have to be physically larger to accommodate higher power requirements.

While total positive power output is useful for comparing robots with some commonality, the Total Cost of Transport (TCoT) is a unitless measure of power and work that can compare wholly different robots, species, and even vehicles (although there is a slight downward trend when comparing large differences in mass [81]). Measuring non-

dimensional work, TCoT is the ratio of work done by a robot's power source to the product of the robot's weight and distance traveled. Similarly, non-dimensional power is simply the ratio of power to the product of weight and velocity. TCoT is lower-bounded by zero, implying no cost to locomote. For reference, humans have a TCoT of 0.2, the ATRIAS robot has a TCoT of 1.0, ASIMO has approximately 3.2, Oregon State's Cassie and MIT Cheetah have 0.7, and the Collins walking robot matches humans at 0.2 [32].

Electric motors generally have to be geared down in order to produce enough torque to drive human-scale machines. Speed reduction at first approximation linearly exchanges speed for torque while keeping mechanical power constant. However, gear reduction also has an effect on resistive heating losses and viscous losses. In the sense of electrical power, the gearing nonlinearly exchanges mechanical work of rotor acceleration and viscous losses with torque-related electrical losses. Because this exchange is nonlinear and convex, there is a "sweet spot" when selecting a motor/gearhead pair for a particular application [22, 122, 127].

5.1.1 *Antagonistic work*

A robot's kinematics play a large role in determining power requirements for a machine, because they determine how the net task power is distributed among the individual motors [3]. Because locomotion is the self-transport of an inertial system, it has a defined net power requirement for modulating external kinetic energy (due to the velocity of the system's mass-center) and potential energy (raising and lowering the mass-center). And, because the robot is composed of individual joints which all have their own speed and torque requirements as defined by the robot's kinematics, this net power can be produced in a combination of different positive and negative components. In fact, the chosen kinematics of a robot as they relate to a task can cause some motors to do substantially more work than the task requires, and that excess work is offset by braking in other actuators (as noted since the inception of robotic limb design [58, 108, 139, 151]). The idea of **antagonistic work** offers a more complete view than simply "minimizing positive work". It describes

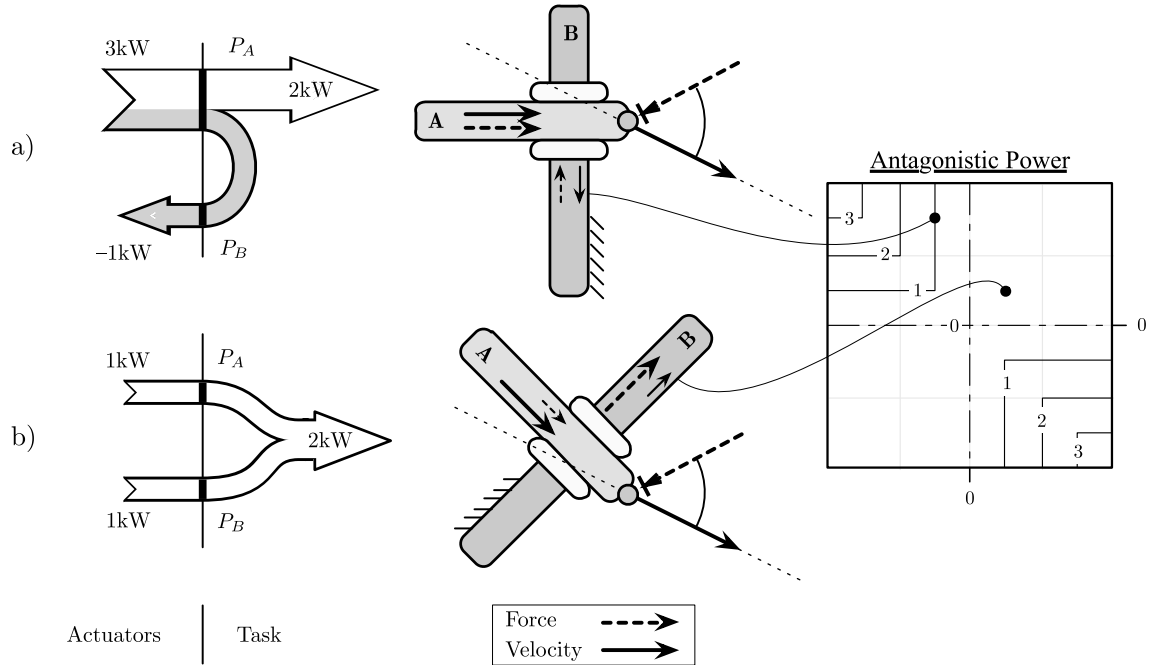


Figure 16: Hardware designs must be "aligned" with a task in order to perform optimally. Above is a simple demonstration where two identical x-y tables must resist an applied load while moving at constant velocity in the plane. The only difference between configuration (a) and configuration (b) is a 45 degree rotation of the table relative to the force and velocity vectors. As a result, table (a) must work harder to do the same task.

how much work is being done that does not benefit the task, whether the task requires positive or negative work. Figure 16 gives a simple example.

Measuring antagonistic work is a matter of comparing the *total power* $\sum |P_i|$ being generated and dissipated within a mechanism to the *net power* $|\sum P_i|$ required by the task. Antagonistic work is then half the difference between total and normed net power (the factor of two is there because antagonistic work is by-definition a pair of equal flows into and out of a system)

$$P_{\text{ant}} = \frac{1}{2}(\sum |P_i| - |\sum P_i|). \quad (49)$$

Since antagonistic work does not benefit the task, we need to design mechanism+task pairs which minimize the amount of antagonistic work. Minimizing antagonistic work is equivalent to minimizing mechanical cost of transport (MCoT) under a zero-regeneration motor power model.

Minimizing the amount of antagonistic work in a mechanism+task pair means finding a set of kinematics which require same-sign work from all actuators. Additionally, *balancing* power among all actuators reduces their individual sizes and makes gait variations less likely to require more power than necessary [3]. Balancing power among all actuators increases the **power quality** of a machine+motion pair. Power quality is a smoothed version of antagonistic work which promotes power sharing between actuators

$$Q = (\Sigma P_i)^2 - \Sigma (P_i^2). \quad (50)$$

which, as you may notice, is proportional to the *variance* in P around the mean $\Sigma P_i/n$, where n is the number of actuators. The less varied the actuator powers, the more likely they are to be the same sign, with the limiting case of zero variance distributing the task power evenly among all the actuators.

For a system with two actuators, power quality reduces to a saddle centered at zero (Figure 17). By maximizing power quality for a machine+task pair, we minimize antagonistic work as well as the total power that must be generated/dissipated by the actuators, thereby reducing the required size of the actuators and power supply [3]. Figure 18 shows how three different leg designs compare in following a spring-mass running gait.

Power quality optimization carries with it a helpful guarantee: finding power-optimal kinematics guarantees that each actuator will have the smallest power requirements, allowing joint-level actuator optimization on a joint-by-joint basis. Maximizing power quality is a kinematic operation which can be done early in the design process, before actuator internals are even specified. Doing so sets the mechanical work of each joint as low as possible, allowing joint actuators to be designed in parallel and still yield a power-optimal manipulator. Power is balanced between all actuators such that any differences in speed or torque for a single actuator can be adjusted from the motor's perspective by an appropriate gear reduction [122, 127, 143]. This removes the curse of dimensionality from large power-optimization problems, decomposing it into kinematic and actuator subproblems. This is in contrast to minimizing over-all power requirements

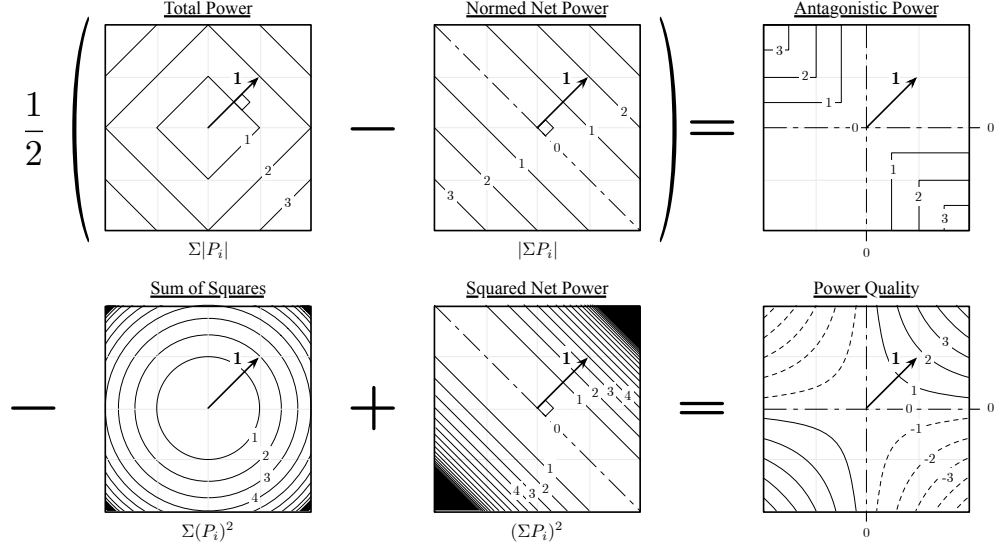


Figure 17: Antagonistic work compared to power quality in 2d. Each is a scalar field with level sets drawn as contour lines. In 2d, power quality is a saddle centered at zero $Q = 2P_1 P_2$. Maximizing power quality brings the operational point closer to the 1-vector (primary diagonal), reducing the chance that task distributions will require antagonistic work. Note the mathematical and visual similarities between antagonistic work (the measure of simultaneous opposite power) and power quality (the measure of power variance).

for the combination of possible kinematics with possible motor+gearhead pairs at all the robot's joints.

Knowing that contact forces for locomotion are predominantly along the leg-length, and end-effector velocities are predominantly in the leg-swing direction, we can determine the best kinematic designs without needing to touch an optimizer (Figure 19). The trivial solution for minimizing antagonistic work is to make sure there is at least one motor per limb that acts in the leg-swing direction *only*. Doing so guarantees that nominal contact forces have no component on that actuator, making its power requirement zero. This is a *marginal* power quality, however. So, all serially-connected limbs have at least marginal power quality, and no antagonistic work in the nominal case. Simply considering the signs of torques and speeds for different parts of the gait can show which designs have negative, marginal, or positive power quality.

The idea of power quality has been extended into a power-optimal pseudoinverse which is applied at run-time to minimize the amount of antagonistic work that a redundant manipulator performs [25].

Quantity	Value	Units	RESULTING GAIT		
Mass	75	kg	Peak Force	1560	N
Stiffness	1.1×10^4	N/m	Stride Period	0.3578	s
Apex height	0.93	m	Stride Length	1.036	m
Apex velocity	3	m/s	Average Speed	2.9	m/s
Neutral length	1.0	m	CoM Peak-to-Peak	0.0717	m
Touchdown angle	0.4	rad			
Gravitational constant	9.81	m/s ²			

Quantity	Parallel	Serial	Spider	Units
Nominal peak power	2.3	1.3	1.0	kW
Nominal MCoT	0.43	0.14	0.14	.
Average MCoT	0.45	0.18	0.16	.
StdDev MCoT	± 0.08	± 0.04	± 0.03	.
Mean power quality	-2,600	0.00	150	.

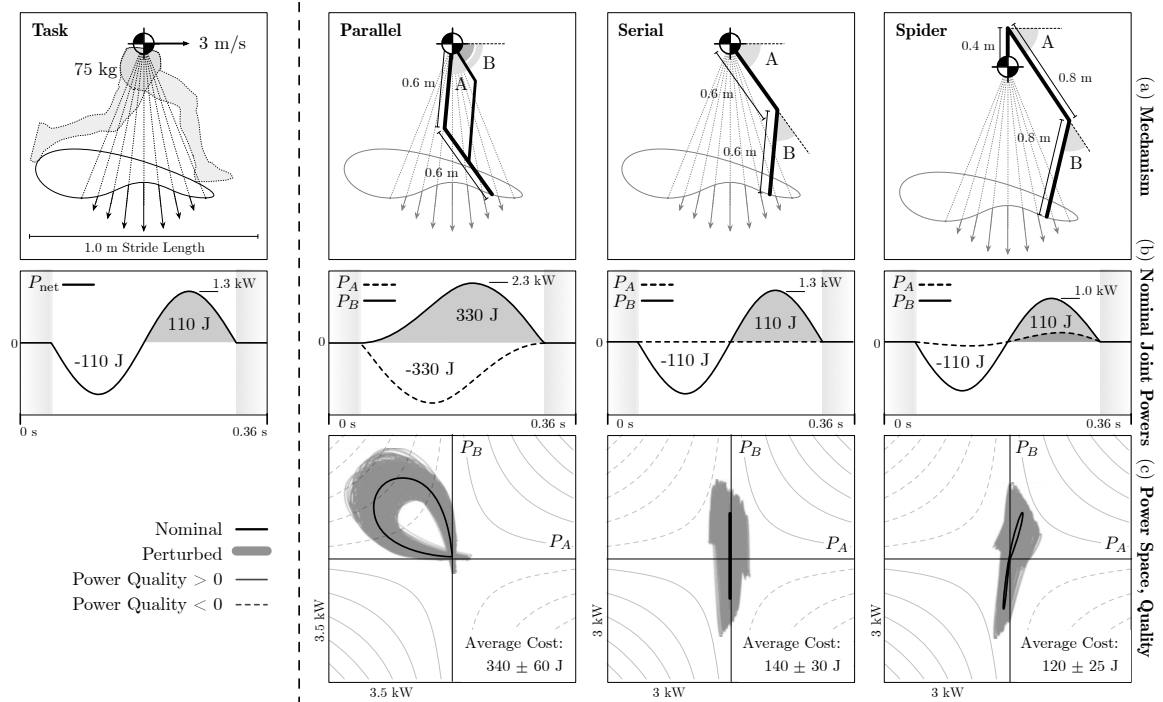


Figure 18: Comparing three leg designs performing the same spring-mass running task. In order to evaluate just the kinematics, none of these designs have components which store energy (springs, flywheels, etc). The *parallel* mechanism has the highest peak power requirements, does more work than the task requires, and therefore has the worst power quality. The *serial* and *spider* designs do no antagonistic work nominally, but the spider design requires less work for perturbed trajectories (a fact which is reflected in its higher power quality).

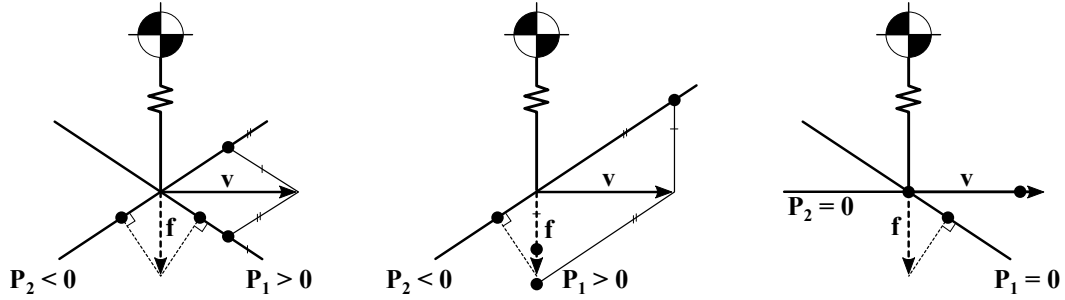


Figure 19: Decomposing force and velocity onto motor bases. Knowing that locomotion forces are predominantly directed along the leg-length (towards the CoM) and that end-effector velocities (for the actuators only) are predominantly perpendicular to the limb, we can differentiate good kinematics from bad by projecting the task force and velocity onto the actuators' basis vectors. Task force is projected onto joint torques by drawing a perpendicular line to each joint's basis. Task velocity must be a linear combination of joint bases, so the task velocity projects onto each basis using lines parallel to the other joint's basis (which is only true for invertible Jacobians). The resulting pair of joint torque/speed shows whether the power requirement is positive, negative, or zero. Since this task requires nominally zero power (task force and velocity are orthogonal), joint powers must cancel.

5.1.2 Matching SLIP

In addition to its other benefits, the SLIP model gives us good heuristics for maximizing efficiency. Bouncy, spring-mass trajectories solve the problem of antagonistic work during level walking [8, 81], and the inclusion of passive compliance alleviates the problem of *cyclic work*. Just like antagonistic work measures the amount of simultaneous, opposing powers, cyclic work measures the amount of energy that is injected into a task just to be removed at some point in the future. Cyclic exchange between gravitational potential energy and spring potential energy typifies spring-mass locomotion. The efficiency of the passive springs allows that gait energy to remain in mechanical form, not dissipated to heat by motor braking.

Practical engineering intuition, research in dynamic locomotion, and animal physiology all agree that series compliance is an important part of any limb design intended for locomotion [64], but where and how is that compliance supposed to be added? The standard approach is to use series-elastic actuators [114], which have compliant elements directly in series with actuators. However, this may lead to an unnecessary number of

springs in the system. In practice, springs can be placed anywhere in an articulated mechanism and lead to a broad array of nonlinear functions at the end effector.

We can match the compliance of the SLIP in robot designs by tuning stiffness to match that of the reduced-order model (thereby matching the gait's resonant frequency) [83], and making sure the directional behavior [85] of the task-space compliance is directed along the leg length (just like the SLIP) [2].

Compliant force is modeled as the gradient of the compliant limb's spring-potential well, and the force generated by the springs is equal to the negative gradient of this potential well at different deflected positions of the end-effector. Over large deflections, the well can have various nonlinear shapes, but locally, this well is modeled as a simple parabolic bowl and the force due to deflection can be modeled as a symmetric matrix in task space. Visually, the force-deflection behavior is an ellipsoid whose axes are eigenvectors and diameters are eigenvalues. So, matching the SLIP is a matter of tuning spring constants until an axis of the compliance ellipsoid is directed along the long axis of the limb. Doing so ensures that lengthwise forces result in lengthwise deflections, just as in the SLIP.

The potential well is a scalar field $V_s(q)$ measuring the spring potential of a mechanism deflected by Δq from some set point q_0 . Naturally, the set point $q = q_0$ represents the zero of the well, since this point corresponds to zero deflection in the springs. When a force is applied, the equilibrium position for the mechanism corresponds to a point $q^* = q_0 + \Delta q$ such that $\nabla V_s(q^*) = J^\top f$. This well-known relation comes from the Euler-Lagrange equation using only a spring-potential term in the action \mathcal{L} .

Equilibrium positions in arbitrary potential wells can be costly to compute, but quadratic wells give good local approximations. As such, we can replace the spring potential with the second-order Taylor series expansion around the set point,

$$V_s(q^*) = V_s(q_0 + \Delta q) \approx V_s(q_0) + \left. \frac{\partial V_s}{\partial q} \right|_{q_0} \Delta q + \frac{1}{2} \Delta q^\top \left. \frac{\partial^2 V_s}{\partial q^2} \right|_{q_0} \Delta q \quad (51)$$

$$\approx 0 + 0 + \frac{1}{2} \Delta q^\top \left. \frac{\partial^2 V_s}{\partial q^2} \right|_{q_0} \Delta q \quad (52)$$

Because the set point is a minimum for the potential field, the zeroth- and first-order terms vanish. With this local approximation, the equilibrium deflection of a mechanism becomes the deflection Δq that satisfies the equality

$$\nabla \left(\frac{1}{2} \Delta q^\top \frac{\partial^2 V_s}{\partial q^2} \Big|_{q_0} \Delta q \right) = \frac{\partial^2 V_s}{\partial q^2} \Big|_{q_0} \Delta q = J^\top f, \quad (53)$$

which is simply the equality of restitution and applied forces. When rearranged using $K^{-1} = C$ and $J\Delta q \approx \Delta x$ (first-order approximation of $f(q)$), we find the task-space compliance tensor, which relates applied loads to deflections,

$$K\Delta q = J^\top f \quad (54)$$

$$\Delta x = JCJ^\top f. \quad (55)$$

The tensor JCJ^\top represents the approximate force-deflection relationship surrounding a neutral position for the mechanism. Because it is a symmetric matrix, the task-space compliance is orthonormal and can be visualized using ellipsoids whose major and minor diameters are the eigenvalues of the matrix. Doing so allows for immediate feedback of the compliant behavior of a design, as well as the ability to design for specific force-deflection responses.

Task-space compliance can also be understood as the sum of symmetric dyads (a *dyadic*) mapping force vectors to deflection vectors

$$[e_1, e_2, \dots] = J \quad \text{columns of the Jacobian} \quad (56)$$

$$\tau_i = e_i^\top f \quad \text{joint torque from task force} \quad (57)$$

$$\Delta\theta_i = \tau_i/k_i \quad \text{joint deflection} \quad (58)$$

$$\Delta x = \sum e_i \Delta\theta_i \quad \text{first-order task deflection} \quad (59)$$

$$= \left(\sum \frac{e_i e_i^\top}{k_i} \right) f \quad \text{sum of dyads} \quad (60)$$

$$= \left(\sum \frac{\|e_i\|^2}{k_i} \hat{e}_i \hat{e}_i^\top \right) f \quad \text{normalizing bases} \quad (61)$$

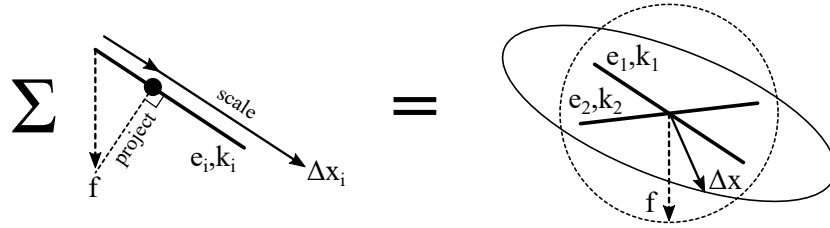


Figure 20: Constructing compliant behavior from the sum of joint dyads. The task force is projected onto each joint basis, then scaled by the squared length of the basis and the joint stiffness, $\frac{\|e_i\|^2}{k_i}$. Each component of the deflection Δx_i is added to the total deflection Δx . Projecting a unit circle of task forces produces the compliance ellipse.

There is one dyad per compliant degree of freedom. Geometrically, each dyad takes the projection of the task force onto that joint's basis vector in the task space, then scales the projection by the length of the basis and the stiffness of the joint (Figure 20). The total task deflection is the sum of these scaled projections of the task force, and the sum of dyads itself is the task-space compliance JCJ^T . Understanding the dyad representation gives much better intuition for the behavior of the mechanism by virtue of its geometric properties and the ability to "visually compute" force-deflection mappings. This also proves that serial mechanisms with hip and knee elasticity can **never** be tuned to give axial compliance, since the leg-length force does not project onto the hip actuator, meaning only the knee dyad participates, always deflecting partially in the leg-angle direction.

Physical compliance can be incorporated into Jacobian-transpose control [28], where the task-space proportional gain tensor is the sum of physical stiffness and controller stiffness, thus allowing for software-controlled impedance augmented with physical springs. Such a method combines impedance control [132] with inverse kinematics [24, 74] in a way that is appropriate for dynamic locomotion.

5.2 MANIPULABILITY METRICS

What "manipulability" means depends on how the controller is going to be interacting with the hardware design. It is a measure of how well a machine can translate control signals (in this case, electric motors as torque sources) into changes in a control target.

Different control targets necessarily lead to different mechanical designs, so we need to specify a reasonable control target for locomotion in general.

As discussed in Section 4.1, locomotion in general reduces to the control of contact force (equivalently the acceleration of the robot's mass-center). So, measuring the manipulability of a machine reduces to measuring the limits on contact force and CoM acceleration. Since locomotion is fundamentally mass transport, and eliciting changes in a mechanical system's motion necessarily requires center of mass velocity changes, we can say that the all abstract locomotion controllers act on centroidal impulse and acceleration[†] (or equivalently, the net contact force with the world). Control authority for a system with such a controller can be measured as the maximum acceleration it can achieve for its center of mass.

5.2.1 "Manipulability" for locomotion

Kinematic and Dynamic Manipulability are classic metrics for fixed-base manipulators, describing the robot's kinematic and dynamic ability within the task space [86, 97, 159]. Peak force [101, 106], velocity, and acceleration [29] in the task space are directional quantities; a robot can be stronger or quicker in one direction than another. These metrics can be used to measure the robot's directional limit at the end effector [30], or they can measure how "isotropic" [90] or uniform is the robot's directional behavior[‡]. Additionally, manipulability can be used as a global performance index to grade a manipulator independently of its pose [52, 103]. Locomotors are different in that they have multiple end effectors, and it is actually the movement of the floating *base* that we care about[§]. Additionally, we care about how effectively the system can be reconfigured in anticipation of future locomotion needs.

[†] That is to say, all locomotion controllers reduce to eliciting CoM acceleration in one way or another. If they did not, the robot would have no control over its own motion through the world. Because of this, we are concerned with the controller's ability to command CoM accelerations, whether or not that is a direct control target of a particular algorithm. More CoM control authority necessarily leads to better manipulability of the hardware as it pertains to system transport.

[‡] Manipulability is sometimes regarded as a distance measure from singularities, since the isotropy of a manipulator will collapse to zero at degenerate configurations.

[§] Since the robot cannot modulate its momentum without contact, no contact necessarily implies no manipulability.

We can define the "manipulability" of a locomotor as its CoM **acceleration limit**. Acceleration is limited because electric motors are limited in how much torque they exert, and it is directional due to mechanism kinematics (the robot could be stronger in leg extension than in leg swing). This measure is normalized relative to the acceleration of gravity (i.e., measured in "g's"). Since CoM acceleration depends on contact, and limbed locomotion requires contact configuration to change frequently and in response to pushes or slips, we must also consider the ability for the robot to control its end-effectors in space. We can use the idea of Dynamic Manipulability to measure the maximum end-effector acceleration in different configurations. End-effector peak acceleration is normalized by step length times squared cycle frequency.

5.2.2 CoM acceleration limit

Calculating the acceleration limit for a floating-base robot in contact with the world is not trivial, since contact models are ill-defined. Given the best case of rigid contact and zero velocity, we can evaluate the acceleration limit of the system's CoM by first evaluating the maximum contact force

$$\mathbf{f} = - \left(\mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T \right)^{-1} \mathbf{J} \mathbf{M}^{-1} \mathbf{B} \mathbf{u}. \quad (62)$$

Using this equation, we project the joint-space hypercube of torque limits into the contact force polytope (Figure 21). We can apply friction cone constraints by "trimming" the contact force polytope by the friction cone geometry $|\mathbf{f}_T| \leq \mu |\mathbf{f}_N|$ (MATLAB has the function `polybool` for this). After scaling the force polytope by system mass to leave acceleration, we can shift the new acceleration polytope by the acceleration of gravity. We are left with the maximum acceleration the system can achieve for its mass-center in all directions. Non-zero joint velocities translate this polytope by Coriolis and velocity-product accelerations, but its shape remains unchanged.

Considering the acceleration limit automatically balances reflected inertia vs total mass of the robot. Too low of a gear ratio means the acceleration is sub-optimal because the

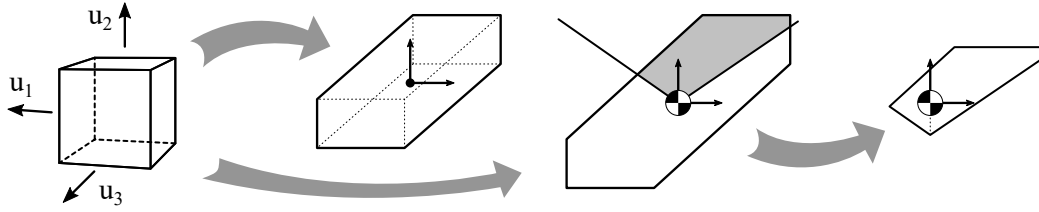


Figure 21: Computation of acceleration limits from actuator torque limits. The end-effector acceleration limit transforms as in $\ddot{\mathbf{x}} = \mathbf{J}\mathbf{M}^{-1}\mathbf{B}\mathbf{u}$ and the CoM acceleration limit transforms as in $-(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}\mathbf{B}\mathbf{u}/m$, where m is the total system mass. The CoM limit additionally must be trimmed by the contact friction cone and shifted by 1 g.

output torque of the actuators is not high enough to accelerate the robot's mass. Too high of a gear ratio means that spinning up the motor's reflected inertia requires too much torque, thus decreasing peak robot acceleration. This idea is analogous to the inertia ratio[†] for single-DoF joints.

We do not necessarily want to maximize the acceleration potential of a locomotor. When the system includes compliance (which all real systems do), having a 1:1 inertia ratio can lead to larger tracking errors and more overshoot/ringing. There are good reasons for intentionally including substantial compliance in a locomotor, so we want just enough acceleration for walking, running, jumping, and whatever else the robot is designed to do, but we still want to have as little reflected inertia as possible.

5.2.3 End-effector acceleration limit

Locomotion requires more than instantaneous centroidal acceleration; the system must be able to reconfigure itself in preparation for future contact. This is also important to position sensors, grasp an object, or prepare for future contact during a fall.

[†] Inertia ratio is a standard "rule of thumb" for specifying the ideal gear reduction for a motor driving an inertial load. This measure applies to single degrees of freedom and, as the name suggests, is the ratio between the motor's reflected inertia and the load inertia. For an ideal, rigid system, the inertia ratio should be identity, but flexible systems require a lighter reflected inertia than the load. The squared ideal gear reduction is equal to the ratio of motor to load inertia, producing a maximal peak acceleration for the motor/load pair. A lower gear ratio causes the motor torque to saturate due to the load inertia, while a higher gear ratio causes it to saturate due to reflected motor inertia.

We can measure the reconfigurability of a robot by computing the dynamic manipulability of its end effectors relative to the robot's root body

$$\ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \quad \text{task acceleration} \quad (63)$$

$$\ddot{\mathbf{x}} = \mathbf{J}\mathbf{M}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{h}) + \dot{\mathbf{J}}\dot{\mathbf{q}} \quad \text{joint-space forcing} \quad (64)$$

$$\ddot{\mathbf{x}} = \mathbf{J}\mathbf{M}^{-1}\mathbf{B}\mathbf{u} \quad \text{controllable term} \quad (65)$$

which can be used to generate a task-space polytope of the acceleration limit given the joint-space torque limits on \mathbf{u} . Computing this polytope is as simple as evaluating Equation 65 for all the positive and negative torque limits in joint space (can be generated as a Gray Code in n bits, where n is the number of actuators). The linear projection of a hypercube is guaranteed to be convex, so the acceleration limit polytope is the convex hull of the set of values of $\ddot{\mathbf{x}}$. Since the uncontrollable terms $\dot{\mathbf{J}}\dot{\mathbf{q}}$ and $\mathbf{J}\mathbf{M}^{-1}\mathbf{h}$ are independent of \mathbf{u} , they will only translate the acceleration limit polytope, not affect its shape. In some cases, we care about the directional dynamic manipulability, but we can reduce the polytope into a simple scalar by finding the least-maximum acceleration. This scalar represents the radius of the largest sphere we can inscribe within the polytope.

We can normalize the end-effector acceleration in the simple case of sinusoid trajectory tracking, which is applicable to walking and running gaits. For the end-effector to travel a distance L along a line in task space at frequency F , it requires a peak task acceleration of $\frac{\pi^2}{2}LF^2$. So, dividing the peak task acceleration of the manipulator by that factor gives multiple of length-root-frequencies the manipulator can accommodate. A reasonable normalization for jogging at 2 m/s would be $\frac{\pi^2}{2}(2\text{ m})(1\text{ Hz})^2 = 9.87\text{ m/s}^2$, which is interestingly very close to 1 g. For a normalized peak limb acceleration of 4, the limb can approximately handle a stride length 4 times longer than given or 2 times as frequent (not considering the workspace or motor speed limits of the machine).

5.3 ROBUSTNESS METRICS

We need to try and create designs which do not fail when the world is not exactly the way the robot thinks it is. This is not the same as component fatigue and mechanical failure, but a failure to maintain stability of a pose or gait when disturbed. A robot's hardware/-software combination must be robust to unexpected changes in ground composition and contact state. If the robot expects there to be a foothold where there isn't, the controller must "fail gracefully", performing autonomous IK to push the limb in the direction of contact. The robot cannot rely on contact sensing in the timeframe where the control system has little or no control over the robot's dynamics (passive dynamics dominate). Even more, there is no meaningful way to define "contact" for granular media, loose rubble, and other non-rigid surfaces. As such, passive dynamics play a large role in the control structure, since they define how the robot responds when disturbed or when the real state of the robot is different from the measured state. Quite a lot of the robustness of a system is determined by the controller, but there are situations which only the hardware can manage, namely impacts and uncertain contact.

5.3.1 *Impact inertia*

The inherent, repeated inelastic collisions during locomotion can be a problem for robots not designed to minimize impact mass. The intuition is clear: hammers have a substantial mass in order to generate large, impulsive forces, while cheetahs, cursorial birds, and many other animals have light, forward-facing tarsi and foot pads to minimize impacts [9]. If a leg is heavy or has a large unsprung mass, these collisions amount to a substantial loss of kinetic energy [64]. If gearheads are subjected to these collisions, the impact forces can be damaging. As such, it is important to shield inertias from the touchdown impact. We analyze the impact inertia as the directional, projected mass of the manipulator into task space (and simply normalized by the total system mass, leaving a percentage of unsprung mass).

The real world is a complex place, and locomoting through it means substantial impacts are going to occur. Since the robot will not always know about impending impacts, we must design the hardware system to minimize the force and impulse due to inevitable collisions between its end effectors and the world. Reducing impact forces reduces the chance of damaging components, and reducing impulses limits the discontinuous effect of impacts on the state of the robot, thereby reducing the control burden.

We may not be able to control the mass of objects in the world, but we can control how heavy limbs are. This is not just reducing over-all weight, but reducing the operational inertia of the end-effectors. Operational inertia is a function of the spatial inertia of the robot's links as well as reflected inertia at its joints. As is the case with large gear reductions, the joint reflected inertia can easily dominate the operational inertia of a limb.

Passive compliance can help reduce the effects of impact by limiting the operational inertia to distal masses only and buffering the transmission of force to the upstream components. Compliant distal joints work well for buffering impacts as well as recycling gait energy (an efficiency bonus). Additionally, compliant pads on the end-effectors are effective at further reducing impact loading, although they typically are viscoelastic and do not recycle energy.

We can derive a machine's impact inertia by assuming that acceleration forces will dominate during an impact. The resulting impact equation for a manipulator is

$$M\Delta\dot{q} = J^T \Lambda \quad (66)$$

$$J\Delta\dot{q} = JM^{-1}J^T \Lambda \quad (67)$$

$$\left(JM^{-1}J^T\right)^{-1} \Delta\dot{q} = \Lambda, \quad (68)$$

where $\Delta\dot{q}$ is the change in joint velocity for an impulse H , and J is the Jacobian matrix at the point of impact (see [2]). Equation 68 shows the projected impact mass in task space, relating instantaneous velocity changes to impulses. The matrix term is symmetric, representing an ellipsoid in task space which shows the directionality of impact inertia.

Projecting the joint-space inertia onto different points on the limb easily visualizes the directionally-dependent impact mass at those points [2, 14]. This leads to the simple

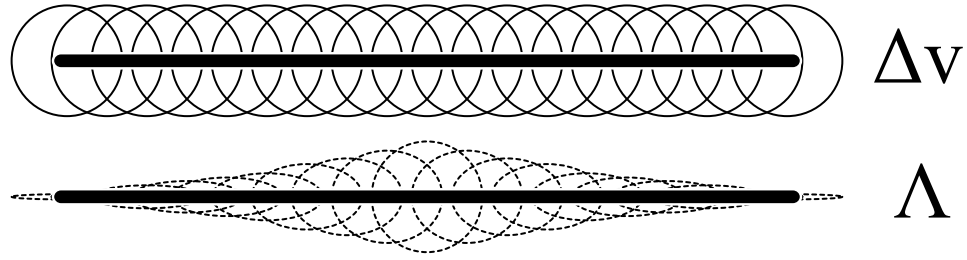


Figure 22: Relationship between uniform Δv and impulse Λ for a homogeneous bar. The impact inertia is defined at every point on any rigid body, whether free-floating or attached to an articulated body. Larger principle axes of the inertia ellipsoids represent heavier directions and shorter principle axes represent lighter directions.

identification of "light" and "heavy" directions for any point on a limb (Figure 22). One simple result that agrees with literature and biology is the forward-facing tarsus, which receives impacts more perpendicular to its length [2, 69].

This method respects the presence of elastic elements and transparent actuators, and therefore gives accurate visual feedback of the true impact response of a machine. As such, it is easy to see which actuators or degrees of freedom need compliance to buffer impacts and which ones don't. For example, high-transparency actuators (those with very low reflected inertia) do not contribute significantly to the impact ellipse and therefore do not need series compliance to reduce force spikes.

One necessary assumption to ensure the validity of this method is that we must make sure that the time period of impacts is far shorter than the natural period of the suspension system. In reality, stiff springs will transfer the ground reaction force to massive components almost as quickly as a "rigid" element would (there is no such thing as rigid in reality, just high stiffness). Impact inertia considers the very instant of impact, so any modeled springs, no matter how stiff, will buffer the upstream mass, so very stiff springs will give inaccurate results. This is because springs of any stiffness in the model are still much softer than the infinite stiffness of rigid links.

Minimizing the magnitude of the operational space inertia is effective at reducing impact effects, but it is not a well-normalized metric. The Impact Mitigation Factor is a recent unitless measure of backdrivability during impact, allowing us to compare robots of different sizes, weights, and joint types [154].

5.4 FEASIBILITY METRICS

Feasibility metrics are perhaps the most important, because they determine if the design can even meet task requirements. While constraints are common in numerical optimizations, it generally does not make sense to implement feasibility checks as hard constraints. Rather, we can take another common technique from computer optimization and add *penalty functions* as objectives [156]. Penalty functions tell us *how far* a design is from being feasible, giving us greater feedback on different designs. Hard constraints can apply when generating designs, where it is easy to say that links can be no shorter than x , link density must be at least ρ , the legs must have a total length ℓ , and so on.

Feasibility is about quantifying the degree to which robots have to violate physical constraints in order to follow a task. It may sound pedantic, but chairs are perfectly efficient and fairly robust to different methods of being sat on, but they are terrible locomotors and cannot carry anyone around (yet). Specifically to locomotion, a locomotor must be able to transport itself through the world through unilateral, friction-constrained contact while obeying motor torque/speed limits. If it cannot, then it is infeasible.

When designing a machine, we typically do not have hard constraints unless they are trivial to enforce (such as link length). It is better to know *how much* a constraint is violated in a design and use that information to migrate towards a more feasible design.

Measuring "how much" a constraint has been violated is exactly the purpose of penalty functions (which are not to be confused with barrier functions). The most basic penalty function is the *quadratic loss*, where the penalty for violating an equality constraint is simply the square of the error (Figure 23). Violating an inequality constraint results in the same quadratic loss, but unviolated inequality constraints carry no cost

$$\phi_{\text{eq}}(e) = e^2 \tag{69}$$

$$\phi_{\text{ineq}}(e) = \begin{cases} 0, & e \leq 0 \\ e^2, & e > 0 \end{cases} \tag{70}$$



Figure 23: Rather than enforcing hard constraints, the *quadratic loss* penalty function encodes "how much" an equality or inequality constraint is being violated.

When trying to optimize or follow a trajectory, we can include a physics violation cost

$$L_{\text{physics}} = \int \left(J^{\top} f + Bu - M\ddot{q} - h \right)^2 dt \quad (71)$$

which accumulates the error between system dynamics and those required by the trajectory [99].

Friction cone and motor speed/torque limits are unilateral constraints, but they can be handled in the same way. The constraints

$$|f_T| \leq \mu f_N \quad \text{tangential force limit} \quad (72)$$

$$k_b |\dot{\theta}| + \frac{R}{k_b} |\tau| \leq V_{\max} \quad \text{winding voltage limit} \quad (73)$$

become the penalty functions

$$L_{\text{cone}} = \int \max(0, f_T^2 - \mu^2 f_N^2) dt \quad (74)$$

$$L_{\text{voltage}} = \int \max(0, k_b |\dot{\theta}| + \frac{R}{k_b} |\tau| - V_{\max})^2 dt \quad (75)$$

such that L_{cone} and L_{voltage} are the feasibility metrics for a particular gait according to friction cone and torque-speed limitations.

6 METRIC-DRIVEN DESIGN PROCESS

Using several unrelated metrics prevents over-specialization of designs, but they need to be combined using some exterior structure. For this, we can use the notion of Pareto optimality [27], where a design "dominates" another only when it is better according to every metric defining the objective space. If designs do not dominate each other, then they are considered equal and merely make different trade-offs between their metrics. This method effectively creates a singular, weakly-ordered metric by combining two or more existing metrics.

6.0.1 *Corner sort*

The multi-objective space has a necessary weak ordering, so members cannot be trivially compared. Pareto-optimality tells us that there exists a set of non-dominated solutions which are better in at least one respect than every other member. In order (no pun intended) to rank designs in a multi-objective space, we can use the Corner Sort algorithm to iteratively find the non-dominated set, assign a rank, then find the non-dominated set of the remaining dominated solutions [152]. This partitions the designs into "shells" of increasing rank, allowing any pair to be meaningfully ordered even if one does not completely dominate the other (Figure 24).

6.1 AUTOMATED DESIGN GENERATION

We can partially automate candidate design generation by stochastically searching the design space [140]. This is no Grand, Unified Optimizer[‡], but it works sufficiently well for

[‡] It is a very exciting idea to have one single optimization program which translates a task specification into a hardware design and a corresponding optimal controller, but that goal is well out of the scope of this work.

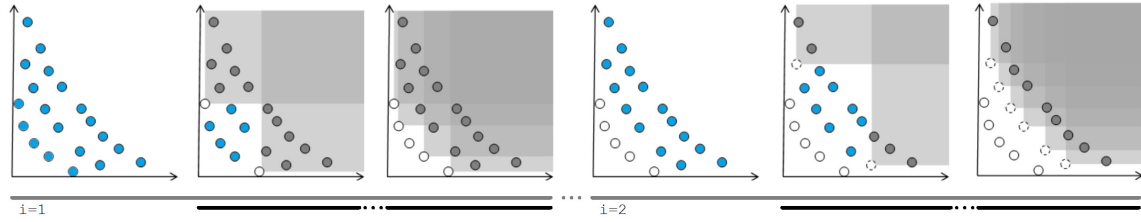


Figure 24: Corner sort algorithm (graphic adapted from [152]). All solutions begin unranked (rank = 0). Keeping track of iterations i , do the following: 1) unmark all unranked solutions, 2) find the unmarked corner solutions, set their rank to be i , mark them, and mark all solutions they dominate, 3) repeat at 2 until all solutions are marked, 4) repeat at 1 until all solutions are ranked.

helping direct the design process. Stochastic search is a simple pattern that uses modular components (Figure 25):

1. Factories for randomizing continuous parameters of given template designs.
2. A population of instantiated designs.
3. A set of metrics which are evaluated for each member of the population.
4. Corner sorting algorithm for providing the Pareto rank of the members.
5. A selection scheme for reducing the population (we cannot exhaustively search the design space).
6. A "temperature" schedule for determining how much to randomize each member of the population when replacing culled members. High initial temperatures allow the population to cover the design space, and slowly-cooling temperatures allow the population to refine towards the true Pareto frontier.

We are dealing with trade-offs, so we need a sufficiently large population which can cover the Pareto frontier in the design space (which is projected into the metric space). The design space is sufficiently complicated that we do not want to risk falling into local optima by using hill climbing. Using just a single candidate with random hill climbing is practically guaranteed to be suboptimal, and it gives us no information about the trade-offs that can be made in the metric space.

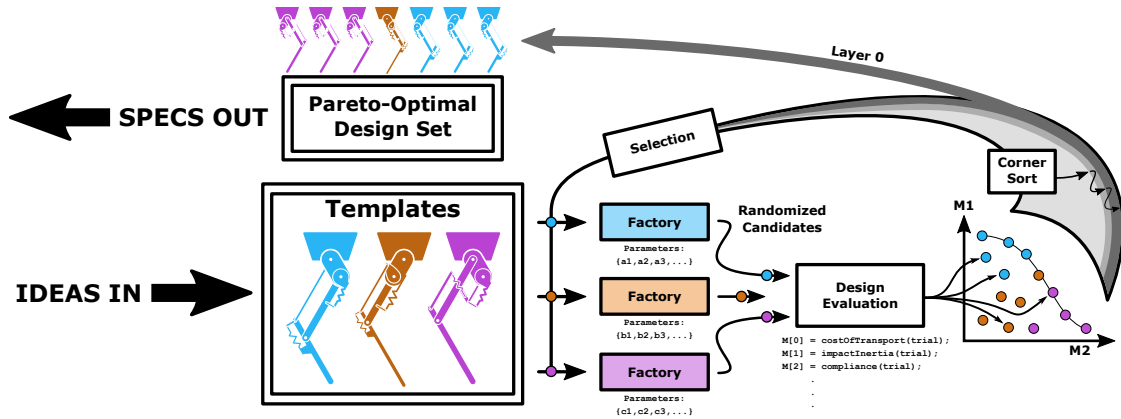


Figure 25: Diagram of the iterative design process. Topologies are passed in as templates with simple parameters such as link lengths and gear ratios. Each template has a corresponding factory which knows how to stochastically vary that template's parameters. A population of candidate designs is evaluated according to a set of metrics, mapping the design space into the objective space. The corner sort algorithm finds the "layers" of non-dominated candidates in the population, and binary tournament selection removes dominated designs while maintaining diversity. This process is both a formal multiobjective optimization algorithm and an analogy for practical iterative design processes used by engineers.

6.1.1 Template designs, factories

We can use human creativity to generate a handful of different design templates, each with a different link topology, motor locations, and so on. These templates have numerical parameters for link-lengths, gear-reductions, motor modules, and so on that are trivial to vary automatically within a set of allowed values.

Factories randomly generate candidate designs that meet design plausibility constraints such as nonzero link length, nonzero mass, etc. For the stochastic search algorithm, the factories produce randomized designs from existing designs until the population has grown by some factor (usually double).

6.1.2 Binary tournament selection with redraw

We want to limit the population of designs while also maintaining diversity, so we need an appropriate rule for removing members. There are many heuristics for this [51], but

binary tournament selection is a simple rule which does not limit diversity[†]. BTS is as simple as drawing two designs randomly from the sorted population and removing the one with lower rank (redraw is allowed, for simplicity). Do this until the population size has been reduced by some factor (usually half).

6.1.3 *Simulated annealing*

We can use a temperature schedule to gradually decrease the amount of randomness in the design factories [84]. This amounts to large steps through the design space during the first iterations, attempting to rapidly approach the Pareto frontier, and small steps during the later iterations (gradually refining the approximate frontier).

We can also incorporate the annealing temperature into the binary tournament selection algorithm, keeping the worse design with some probability. This helps maintain a diverse population at high temperatures.

[†] We could just remove all the dominated solutions from a population, but that decreases coverage of the design space. We want to leave as many varied designs on the table as we can while still hunting for the Pareto frontier.

7 EXAMPLE DESIGNS

To illustrate the use of locomotion metrics, we evaluate select metrics for three designs: a simple 2-link serial mechanism, an ATRIAS-like parallel mechanism, and a polar mechanism. These designs all have invertible kinematics (neglecting spring deflection), but all the metrics presented operate on any general combination of redundant serial and parallel kinematics (including bi-articulation). See Section 3 for details on modeling.

In a complete design process, the results of these evaluations would "feed back" into the next iteration of designs. Designers would adjust link lengths, shift mass, choose different motors, and so on to try to improve the next round of metrics. This iterative design process teases us with the potential for automation via stochastic search. Section 6 discusses such a method, but that method will not be followed here, since we are focusing on the metrics themselves.

Each design can be evaluated using the *visual* form of numeric metrics to get a *feel* for how well a design will behave to the point of sanity-checking numerical values or optimization results (Section 5). This is a transferable technique for guiding the design process when creatively generating new designs. Visual representations of different limb properties can be quickly and easily interpreted even with little or no understanding of the underlying mathematics.

The example designs we evaluate here are constructed using uniform-density bars for links, and all designs use the same density material in their construction. As such, the inertial properties of the links are derived from the area of "ink" used to draw them. Actuators are represented by additional masses at the actuated joint and the associated reflected inertia in the system's mass matrix diagonals. All designs have a leg length of 1 meter, use the same motors (with possibly different gearhead reductions), and have series compliance tuned for the same nominal toe stiffness. Each leg design is attached to an identical floating base. Even though these designs are made out of the same "stuff", they

are used as an exhibition of metrics only and not compared in detail, since real designs will be significantly different in their weight distribution. Table 1 shows the collected numerical evaluation of all metrics and Figures 26, 27, and 28 show the visual representations.

We evaluate each designs' CoM acceleration limit to track their ability to generate motion, their end-effector acceleration limit to track their ability to reconfigure, their compliant behavior as appropriate for spring-mass locomotion, their impact mass (smaller is better), and their power quality (positive is better).

COM ACCELERATION LIMIT This polytope represents the robot's ability to change its motion when in contact with rigid ground (making it the "manipulability" of floating-base manipulators). We draw a convex envelope around the robot's mass-center to show the boundary of possible accelerations the robot can achieve given motor torque and friction-cone constraints. CoM acceleration limits are directional, so a robot may be able to accelerate straight up more easily than it can accelerate laterally (represented by a thin and tall polytope). All floating-base robots on Earth will have the same downward acceleration limit: 1 g (unless it can pull itself toward the ground using a bilateral contact constraint). We are not always interested in the isotropy of manipulability; many more fitting measures could be drawn from the acceleration envelope such as vertical limit while standing, forward limit in a crouched position, lateral limit in a split-leg pose, and so on. We primarily affect the CoM acceleration limit by changing gear ratio, since this will balance reflected inertia of the actuators against the total mass of the robot.

END-EFFECTOR ACCELERATION LIMIT Just as the CoM has a directional acceleration limit, the robot's end-effectors also have such a limit. Here, we consider this acceleration relative to the robot's root frame (making this metric very much like the Dynamic Manipulability of a fixed-base manipulator). The end-effector acceleration is also directional, where the envelope's distance from the end-effector represents a higher peak acceleration in that direction. General, directionally-uniform behavior is not necessarily the goal, since the acceleration requirement for locomotion is very much anisotropic and configuration-dependent. While manipulability is independent of the task, the project/task requirements

necessarily determine different "cross-sections" of manipulability which matter more than others (the anisotropies).

END-EFFECTOR COMPLIANT RESPONSE Compliant behavior may be difficult to grasp when thinking in terms of individual joint stiffnesses, but the whole-limb behavior is succinctly represented as an ellipse centered at the end-effector. This ellipse is the result of a circle of unit forces being scaled along the principle axes of the ellipse (major and minor diameters). Each unit force on this circle maps to the scaled point on the ellipse representing the deflection of the end-effector under that force. Since this is a scaling operation, the directions of forces along the major and minor diameters remain unchanged. So, a unit force along the long axis of the ellipse generates a large equilibrium deflection in that direction, and that same force along the short axis results in less deflection. For spring-mass locomotion, we desire leg-length forces to result in leg-length deflections, so we want to tune the leg designs to have a principle compliant axis along the leg length at the desired spring rate.

END-EFFECTOR IMPACT MASS Impact with rigid ground is modeled as a linear relationship between the end-effector's velocity delta (before and after contact) and the impulse required to generate that delta. Just like compliance, this relationship is directional and can be visualized as an ellipse with major and minor axes. Long axes represent a heavier impact mass (more impulse to elicit a unit velocity delta in that direction), and short axes represent a lighter impact mass.

POWER QUALITY We can use a simple task definition to evaluate how well different designs decompose the net power required onto individual actuators. It is important to limit actuators from contributing simultaneous positive and negative powers due to the lack of regeneration and larger power requirements. Visually, task velocity and force decompose onto the design's actuator basis vectors (lines representing the direction/speed of the end-effector given unit velocity each actuator, fixing all other actuators). Force projects literally (as a scalar product with each basis), and velocity projects as the sum of

scaled basis vectors. Power quality can be coarsely-graded as positive or negative, or it can be evaluated using the products of speed and torque for each actuator (area of the rectangle with side-lengths of the force and velocity projections).

7.0.1 *Polar Design*

The first design (Figure 26) is simply a prismatic leg-length actuator pivoted by a revolute leg-angle actuator and connected to an unconstrained torso. Each actuator includes a serially-actuated spring to add compliance. This design has the same toe kinematics (e_1 and e_2 basis vectors in the figure) as various straight-line linkages, and thus can represent the compliance, power quality, and manipulability of any other revolute-prismatic design (but not impact inertia). Due to symmetry, the compliant and impact response are symmetrical around the leg-length, and so is the manipulability. Such a design has marginal power quality, due to the decomposition of walking/running forces onto a single actuator alone, meaning only one motor does any work (although this neglects inertial forces, which certainly have a leg-angle component). It is trivial to tune the compliant response of this design to match the SLIP, and achieving the desired stiffness is as simple as setting the prismatic spring to exactly that stiffness. Since leg-length forces decompose onto the prismatic spring only, walking/running forces will only ever elicit a leg-length response.

7.0.2 *2-link Serial Design*

The second example design has two links connected rotationally at the knee and hip. Like the other leg designs, this one is connected to a floating base. Since the links are connected serially, the hip acts to swing the leg as a whole, and the knee acts to swing just the second link. Each actuator has a serially-connected spring at its output, so as to add compliant behavior. The 2-link design is asymmetric, meaning its compliant, impact, and manipulability behavior is asymmetric. There is more impact mass along the length of the shin, due to the actuator located at the knee. Actuator 1 (which has e_1 as its basis vector)

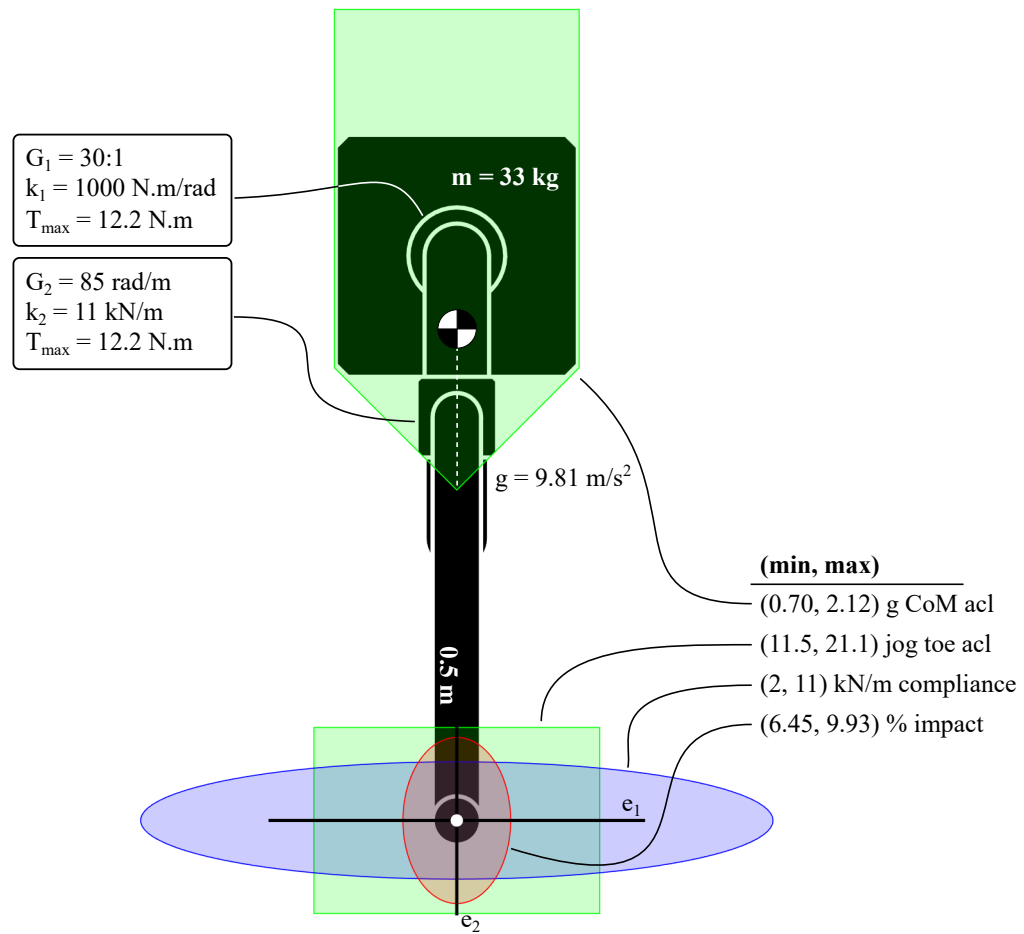


Figure 26: The first leg design is a revolute-prismatic "pogo stick" design. The hip includes a gear-motor and a series spring, and the leg-length is a sprung slide actuated by linear actuator (let's say it is a ballscrew driven by a rotary motor). Locomotion forces decompose onto the leg-length actuator only, and swing velocity onto the leg-angle only, giving this design a marginal (zero) power quality.

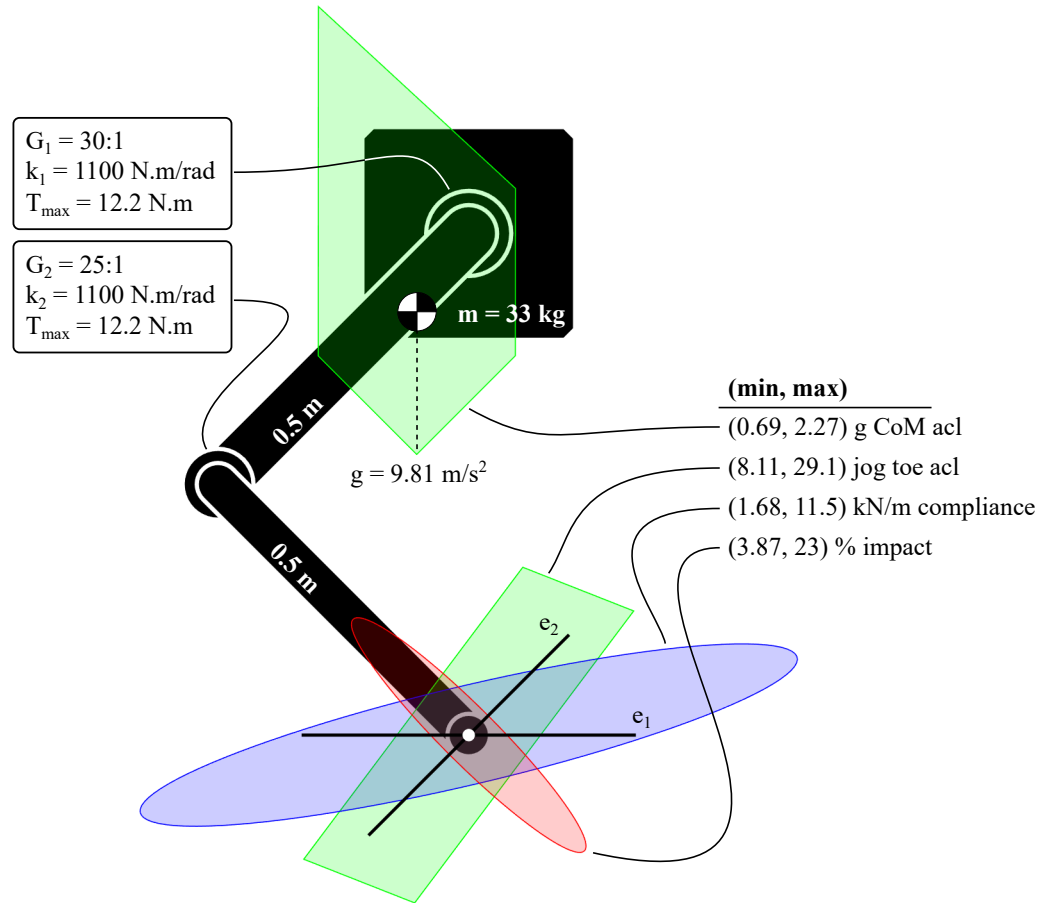


Figure 27: The second leg design is a 2-joint serial mechanism. Each joint includes a gearmotor and a series spring. Just like the pogo stick, locomotion forces act at the knee only and leg swing is controlled by the hip, giving the serial design a marginal (zero) power quality. Unfortunately, since there is no series compliance along the length of the shin, the knee motor mass participates in impacts directed along the shin. As such, the max impact percentage is higher for this design than the others. Also, since leg-length forces do not decompose onto the hip actuator, only the knee spring will ever deflect under load, meaning this design can never be tuned for a leg-length principal compliance.

does not act in the leg-length direction, so walking/running forces decompose only onto the knee actuator (e_2). As such, this design can never have a compliant SLIP response, since leg-length forces never generate leg-length deflections, no matter the spring constants used at the knee and hip (you can use the dyadic representation of compliance to prove this graphically).

7.0.3 4-bar Parallel Design

Lastly, we have a pantograph leg with four equal-length links. Because this design has a loop-closure, we project the open-chain dynamics of two, two-link chains using the constraint that the ends of these chains share position, velocity, and acceleration (see Section 3). The two series-compliant actuators are located at the hip and drive each of the proximal links. As shown in Figure 28, the parallel design is highly symmetric in its dynamic behavior. It is very good at accelerating its mass-center vertically (which was one of the intuitive reasons for using a parallel mechanism in ATRIAS, although it was just gut instinct at the time). It has isotropic impact and compliant responses in the pose shown, but these responses scale along the length of the leg as the leg extends and retracts (and retain a mirror symmetry about the length of the leg). The critical flaw with this design, as described earlier, is the antagonistic work that must be done during walking and running. Because leg-length forces decompose into equal and opposite actuator torques, and leg-swing velocity decomposes into equal actuator speeds, there is *nominal* equal and opposite work being done by the actuators. As such, this design has very poor power quality.

7.1 COMPARISON

Table 1 shows the collected evaluation of the metrics shown in the leg design figures. Since most of these metrics are directional, the numerical quantities are taken as the maximum value in any direction, along with an isotropy value (which is simply the least-maximum value divided by the limiting value). Isotropy isn't important if a particular project is not concerned with the uniformity of a design's dynamic behavior. Similarly, only some of these metrics may be important, depending on the goals of different projects. For example, fixed-base manipulators aren't concerned with power quality, because they are attached to wall power. If a design is focused on actuator transparency, then compliance becomes less important.

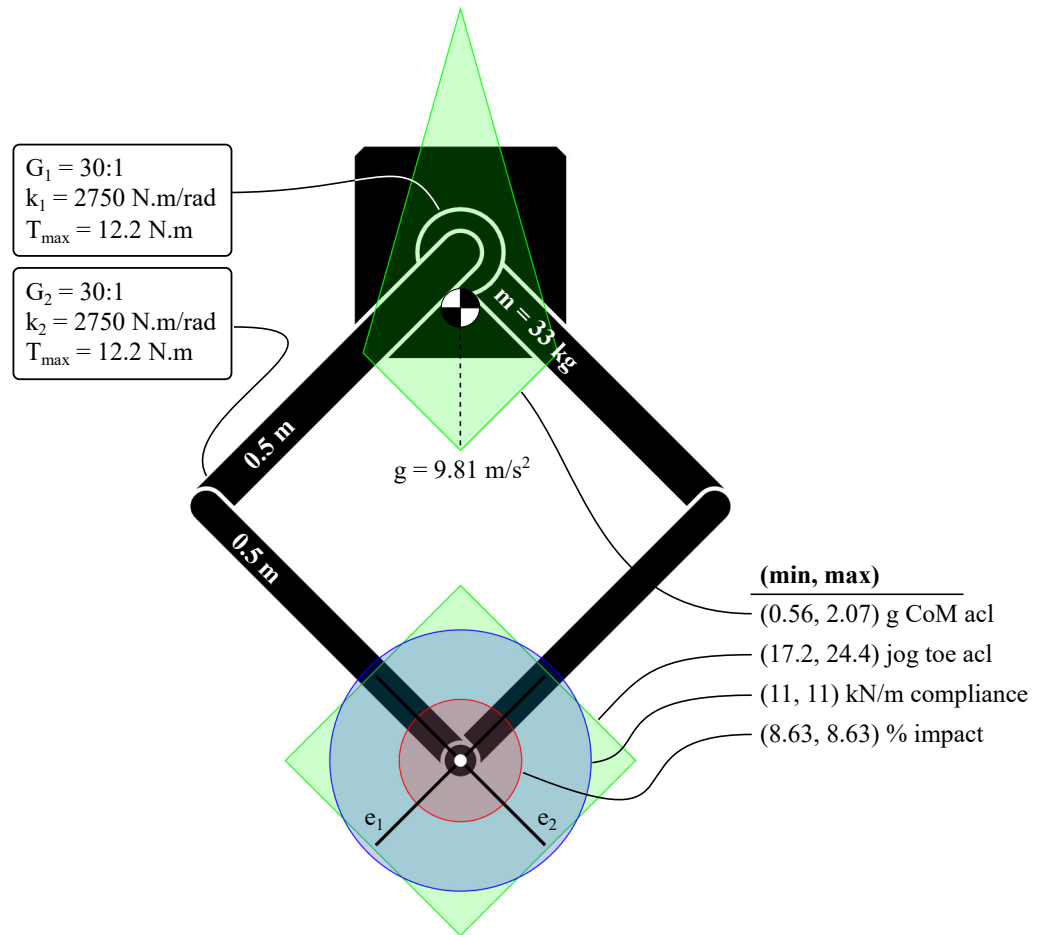


Figure 28: The third leg design is a parallel mechanism with two series-elastic motors mounted on the body. Two branches of links require a loop-closure constraint at the toe to enforce the four-bar linkage kinematics. Because leg-length forces decompose equally onto both motors, which also both have to rotate to swing the leg, the parallel design has a *negative* power quality, indicating antagonistic work requirements.

Three designs are too small a set to draw any general conclusions, but we will compare these designs nonetheless. In reality, the results shown here would feed back into the design process, directing modifications to these designs, or necessitating the addition of completely new designs into the mix. Further designs could include straight-line mechanisms, articulated mechanisms with more links (the metrics have no issue with redundancy), 3D designs (using ellipsoids and polytopes with volume instead of area), and so on.

Tuning gear reductions has an effect on impact mass (unless there is a series spring below the high-reduction actuator) and trades off CoM acceleration for toe acceleration (just like inertia matching for a single DoF system). The serial design has the best combination of CoM and toe acceleration limits, but the isotropy is low at the toe. The CoM acceleration isotropy is fairly low across the board due to the friction cone constraint on contact forces (all robots can only accelerate at 1 g downward). Leg stiffness is tuned to the same value for all designs, but the asymmetry of the serial design means it will never have pure SLIP compliant behavior. The serial design has the worst unsprung mass along the length of its shin, due to the actuator mass at the knee, but it actually has the lowest unsprung mass perpendicular to the shin. Using the predominantly leg-length forces and leg-angle velocities of walking and running, we see that the prismatic and serial designs have nominally zero antagonistic work (marginal power quality), and the parallel design has significant antagonistic work (negative power quality).

Quantity	Pogo	Serial	Parallel	Units	Order
CoM acceleration limit	2.12	2.27	2.07	g	>
	0.33	0.30	0.27	"iso"	>
Toe acceleration limit	21.1	29.1	24.4	"jogs"	>
	0.55	0.28	0.70	"iso"	>
Leg stiffness (tuned)	11.0	11.5	11.0	kN/m	=
Unsprung mass	9.93	23.0	8.63	%	<
	0.65	0.17	1.00	"iso"	>
Power quality	=0	=0	<0	W ²	>

Table 1: Properties of three leg designs. Metrics are evaluated at a nominal leg length of 0.707 m. CoM acceleration limit is normalized by the acceleration due to gravity, toe acceleration is normalized by the peak acceleration required by a 2 m/s jogging gait (hence "jogs", 9.87 m/s^2), and impact is normalized as a the percent of robot mass which is involved in the impact (unsprung). "Iso" is short for "isotropy" (the ratio of least to greatest measurement for directional quantities), where 1 is completely uniform. The least-maximum value of the metric is the limit times the isotropy value. These values are represented visually in the following figures.

8 CONCLUSION

This work has been collected for lack of a detailed roadmap in mechanical design for locomotion. It is an intricate problem, rooted in dynamics and with close connections to control design and contact mechanics. Such interconnection makes locomotion a particularly interesting problem to analyze, because it resists established norms of controller-dominated, just-needs-to-exist hardware designs in favor of "mechanical intelligence" and the teaming of hardware and software.

Clearly defined task parameters and design goals guide us in our search for better hardware designs for locomotion systems. Competing design goals lead to trade-offs in the design space, and the iterative nature of design means that the only "solution" to the design problem is continual refinement of existing designs.

We have attempted to render the given limb-design metrics in geometric forms such that the behavior of the limb design is visually apparent. Compliance and impact behavior are represented by ellipsoids indicating the deflection and impulse response for applied forces and velocity deltas, respectively. Floating-base acceleration limits are inherently convex polytopes (intersection of half-spaces) showing the directional limit of controlled acceleration for a locomotion system. Power quality can be evaluated by drawing actuator basis vectors in the task space and projecting the task's force and velocity onto those bases, indicating the sign of actuator powers and showing any antagonistic work. Some of these processes can be largely performed on paper, allowing designers to roughly evaluate candidate designs without any modeling or simulation.

A structured design process encodes what is fundamentally an optimization program, where our design process is a human-in-the-loop algorithm designed to solve that program. While what we have is no Grand, Unified Optimizer, the metrics can be evaluated in simulation, and the design process is written in a way that is amenable to computer

automation. The largest hurdles are through-contact trajectory optimization with realistic physics and limb topology grammars to encode and randomize design candidates.

8.1 DISCLOSURE

Andrew Abate is an employee of Agility Robotics, a company that licenses technology derived from Andrew’s research at Oregon State University and sells versions of the Cassie bipedal robot for locomotion research.

8.2 ACKNOWLEDGMENTS

People are products of their environment, therefore academics are products of each other. This work is naturally a shared work of myself and all the smart, driven co-workers I have had the pleasure to know. Namely my labmates at Oregon State over the years: Christian Hubicki, Hamid Vejdani, Siavash Rezazadeh, Daniel Renjewski, Jesse Grimes, Mikhail Jones, Brent Piercy, Kevin Kemper, Devin Koepl, Andrew Peekema, and Ryan Van Why. I also deeply appreciate the time of my formal (and ad-hoc) advisors: Jonathan Hurst, Ross Hatton, Matt Campbell, Burak Sencer, and C. David Remy.

This material is based upon work supported by the National Science Foundation under Graduate Research Fellowship Program Grant No. 1314109, and also under DARPA award number W911NF-16-1-0002.

BIBLIOGRAPHY

- [1] Andy Abate. “Preserving the Planar Dynamics of a Compliant Bipedal Robot with a Yaw-Stabilizing Foot Design.” In: (2014).
- [2] Andy Abate, Ross L Hatton, and Jonathan W Hurst. “Passive-dynamic leg design for agile robots.” In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 4519–4524.
- [3] Andy Abate, Jonathan W Hurst, and Ross L Hatton. “Mechanical Antagonism in Legged Robots.” In: *Robotics: Science and Systems (RSS)*. 2016.
- [4] Jeffrey Ackerman and Justin Seipel. “Energy efficiency of legged robot locomotion with elastically suspended loads.” In: *IEEE Transactions on Robotics* 29.2 (2013), pp. 321–330.
- [5] Mojtaba Ahmadi and Martin Buehler. “Controlled passive dynamic running experiments with the ARL-monopod II.” In: *Robotics, IEEE Transactions on* 22.5 (2006), pp. 974–986.
- [6] Antonio Alba, Francesco Bucchi, Francesco Frendo, and Marco Gabiccini. “Kinematic Optimization of the Arm of a Working Machine.” In: *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. 2014, V006T10A071–V006T10A071.
- [7] R. McNeill Alexander. *Elastic mechanisms in animal movement*. Cambridge University Press, 1988.
- [8] R. McNeill Alexander. “Three uses for springs in legged locomotion.” In: *International Journal of Robotics Research* 9.2 (1990), pp. 53–61.

- [9] R. McNeill Alexander, MB Bennett, and RF Ker. "Mechanical properties and function of the paw pads of some mammals." In: *Journal of Zoology* 209.3 (1986), pp. 405–419.
- [10] R. McNeill Alexander and Alexandra Vernon. "The mechanics of hopping by kangaroos (Macropodidae)." In: *Journal of Zoology* 177.2 (1975), pp. 265–303.
- [11] Emanuel Andrada, Christian Rode, and Reinhard Blickhan. "Grounded running in quails: simulations indicate benefits of observed fixed aperture angle between legs before touch-down." In: *Journal of theoretical biology* 335 (2013), pp. 97–107.
- [12] Ben Andrews, Bruce Miller, John Schmitt, and Jonathan E Clark. "Running over unknown rough terrain with a one-legged planar robot." In: *Bioinspiration & biomimetics* 6.2 (2011), p. 026009.
- [13] Ömür Arslan and Uluç Saranlı. "Reactive planning and control of planar spring-mass running on rough terrain." In: *Robotics, IEEE Transactions on* 28.3 (2012), pp. 567–579.
- [14] Haruhiko Asada. "Dynamic analysis and design of robot manipulators using inertia ellipsoids." In: *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*. Vol. 1. IEEE. 1984, pp. 94–102.
- [15] David Baraff. "Linear-time dynamics using lagrange multipliers." In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM. 1996, pp. 137–146.
- [16] DN Beal, FS Hover, MS Triantafyllou, JC Liao, and GV Lauder. "Passive propulsion in vortex wakes." In: *Journal of Fluid Mechanics* 549 (2006), pp. 385–402.
- [17] JE Bertram, Andy Ruina, CE Cannon, Y Hui Chang, and Michael J Coleman. "A point-mass model of gibbon locomotion." In: *Journal of Experimental Biology* 202.19 (1999), pp. 2609–2617.
- [18] Pranav A Bhounsule, Jason Cortell, and Andy Ruina. "Design and control of Ranger: an energy-efficient, dynamic walking robot." In: *Proc. CLAWAR*. 2012, pp. 441–448.

- [19] Aleksandra V Birn-Jeffery, Christian M Hubicki, Yvonne Blum, Daniel Renjewski, Jonathan W Hurst, and Monica A Daley. "Don't break a leg: running birds from quail to ostrich prioritize leg safety and economy on uneven terrain." In: *Journal of Experimental Biology* 217.21 (2014), pp. 3786–3796.
- [20] Reinhard Blickhan. "The spring-mass model for running and hopping." In: *Journal of biomechanics* 22.11-12 (1989), pp. 1217–1227.
- [21] Reinhard Blickhan, Andre Seyfarth, Hartmut Geyer, Sten Grimmer, Heiko Wagner, and Michael Günther. "Intelligence by mechanics." In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 365.1850 (2007), pp. 199–220.
- [22] Edgar Bolívar, Siavash Rezazadeh, and Robert Gregg. "A General Framework for Minimizing Energy Consumption of Series Elastic Actuators With Regeneration." In: *ASME 2017 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2017, V001T36A005–V001T36A005.
- [23] Ben Brown and Garth Zeglin. "The bow leg hopping robot." In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*. Vol. 1. IEEE. 1998, pp. 781–786.
- [24] Samuel R Buss. "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods." In: *IEEE Journal of Robotics and Automation* 17.1-19 (2004), p. 16.
- [25] Nathan M Cahill, Thomas Sugar, Matthew Holgate, and Kyle Schroeder. "Understanding Power Loss due to Mechanical Antagonism and a New Power-Optimal Pseudoinverse for Redundant Actuators." In: *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers. 2017, V05BT08A077–V05BT08A077.
- [26] Matthew I Campbell, Jonathan Cagan, and Kenneth Kotovsky. "A-design: an agent-based approach to conceptual design in a dynamic environment." In: *Research in Engineering Design* 11.3 (1999), pp. 172–192.

- [27] Yair Censor. "Pareto optimality in multiobjective problems." In: *Applied Mathematics and Optimization* 4.1 (1977), pp. 41–59.
- [28] Chien-Chern Cheah. "Task-space PD control of robot manipulators: unified analysis and duality property." In: *The International Journal of Robotics Research* 27.10 (2008), pp. 1152–1170.
- [29] D-Z Chen and L-W Tsai. "The generalized principle of inertia match for geared robotic mechanisms." In: *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE. 1991, pp. 1282–1287.
- [30] Hee-Byoung Choi and Jeha Ryu. "Convex hull-based power manipulability analysis of robot manipulators." In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 2972–2977.
- [31] Steven H Collins, Peter G Adamczyk, and Arthur D Kuo. "Dynamic arm swinging in human walking." In: *Proceedings of the Royal Society of London B: Biological Sciences* 276.1673 (2009), pp. 3679–3688.
- [32] Steven H Collins and Andy Ruina. "A bipedal walking robot with efficient and human-like gait." In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE. 2005, pp. 1983–1988.
- [33] Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. "Whole-body motion planning with centroidal dynamics and full kinematics." In: *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE. 2014, pp. 295–302.
- [34] Tomas De Boer. *Foot placement in robotic bipedal locomotion*. TU Delft, Delft University of Technology, 2012.
- [35] Avik De and Daniel E Koditschek. "The Penn Jerboa: A platform for exploring parallel composition of templates." In: *arXiv preprint arXiv:1502.05347* (2016).
- [36] Tom Erez and Emanuel Todorov. "Trajectory optimization for domains with contacts using inverse dynamics." In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 4914–4919.

- [37] Michael Ernst, Hartmut Geyer, and Reinhard Blickhan. "Spring-legged locomotion on uneven ground: a control approach to keep the running speed constant." In: *International Conference on Climbing and Walking Robots (CLAWAR)*. World Scientific. 2009, pp. 639–644.
- [38] Roy Featherstone. "The acceleration vector of a rigid body." In: *The International Journal of Robotics Research* 20.11 (2001), pp. 841–846.
- [39] Roy Featherstone. "Plucker basis vectors." In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE. 2006, pp. 1892–1897.
- [40] Roy Featherstone. "A beginner's guide to 6-D vectors (part 2)." In: *IEEE robotics & automation magazine* 17.4 (2010), pp. 88–99.
- [41] Roy Featherstone. "A beginner's guide to 6-d vectors (part 1)." In: *IEEE robotics & automation magazine* 17.3 (2010), pp. 83–94.
- [42] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.
- [43] Roy Featherstone and David Orin. "Robot dynamics: equations and algorithms." In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*. Vol. 1. IEEE. 2000, pp. 826–834.
- [44] Martin L Felis. "RBDL: an efficient rigid-body dynamics library using recursive algorithms." In: *Autonomous Robots* 41.2 (2017), pp. 495–511.
- [45] Paulo Flores, Margarida Machado, Eurico Seabra, and Miguel Tavares da Silva. "A parametric study on the Baumgarte stabilization method for forward dynamics of constrained multibody systems." In: *Journal of computational and nonlinear dynamics* 6.1 (2011), p. 011019.
- [46] Gerrit A Folkertsma, Sangbae Kim, and Stefano Stramigioli. "Parallel stiffness in a bounding quadruped with flexible spine." In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 2210–2215.
- [47] Robert J Full and Daniel E Koditschek. "Templates and anchors: neuromechanical hypotheses of legged locomotion on land." In: *Journal of Experimental Biology* 202.23 (1999), pp. 3325–3332.

- [48] Hartmut Geyer, Reinhard Blickhan, and Andre Seyfarth. "Natural dynamics of spring-like running: Emergence of selfstability." In: *5th International Conference on Climbing and Walking Robots*. Suffolk, England: Professional Engineering Publishing Ltd. 2002, pp. 87–91.
- [49] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. "Spring-mass running: simple approximate solution and application to gait stability." In: *Journal of theoretical biology* 232.3 (2005), pp. 315–328.
- [50] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. "Compliant leg behaviour explains basic dynamics of walking and running." In: *Proceedings of the Royal Society of London B: Biological Sciences* 273.1603 (2006), pp. 2861–2867.
- [51] David E Goldberg and Kalyanmoy Deb. "A comparative analysis of selection schemes used in genetic algorithms." In: *Foundations of genetic algorithms*. Vol. 1. Elsevier, 1991, pp. 69–93.
- [52] Clement Gosselin and Jorge Angeles. "A global performance index for the kinematic optimization of robotic manipulators." In: *Journal of Mechanical Design* 113.3 (1991), pp. 220–226.
- [53] Jesse A Grimes. "ATRIAS 1.0 & 2.1: enabling agile biped locomotion with a template-driven approach to robot design." In: (2013).
- [54] Jesse A Grimes and Jonathan W Hurst. "The design of ATRIAS 1.0 a unique monopod, hopping robot." In: *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 548–554.
- [55] Duncan W Haldane, MM Plecnik, Justin K Yim, and Ronald S Fearing. "Robotic vertical jumping agility via series-elastic power modulation." In: *Science Robotics* 1.1 (2016), eaag2048.
- [56] R van Ham, Thomas G Sugar, Bram Vanderborght, Kevin W Hollander, and Dirk Lefeber. "Compliant actuator designs." In: *Robotics & Automation Magazine, IEEE* 16.3 (2009), pp. 81–94.

- [57] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. "3D Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics." In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016.
- [58] Shigeo Hirose. "Some considerations on a feasible walking mechanism as a terrain vehicle." In: *3rd CISM-IFTOMM Int. Symp. on Theory and Practice of Robots and Manipulators*. 1978, pp. 357–375.
- [59] Philip Holmes, Robert J Full, Dan Koditschek, and John Guckenheimer. "The dynamics of legged locomotion: Models, analyses, and challenges." In: *Siam Review* 48.2 (2006), pp. 207–304.
- [60] Christian M Hubicki and Jonathan W Hurst. "Running on soft ground: Simple, energy-optimal disturbance rejection." In: *International Conference on Climbing and Walking Robots (CLAWAR)*. 2012.
- [61] Christian Michael Hubicki. "From running birds to walking robots: optimization as a unifying framework for dynamic bipedal locomotion." In: (2014).
- [62] Christian Hubicki, Andy Abate, Patrick Clary, Siavash Rezazadeh, Mikhail Jones, Andrew Peekema, Johnathan Van Why, Ryan Domres, Albert Wu, William Martin, et al. "Walking and running with passive compliance: Lessons from engineering a live demonstration of the ATRIAS biped." In: *IEEE Robotics and Automation Magazine* ().
- [63] Christian Hubicki, Jesse Grimes, Mikhail Jones, Daniel Renjewski, Alexander Spröwitz, Andy Abate, and Jonathan Hurst. "ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot." In: *The International Journal of Robotics Research* 35.12 (2016), pp. 1497–1521.
- [64] Jonathan W Hurst. "The role and implementation of compliance in legged locomotion." PhD thesis. Carnegie Mellon University, 2008.

- [65] Marco Hutter, C David Remy, Mark A Hoepflinger, and Roland Siegwart. "Scarleth: Design and control of a planar running robot." In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 562–567.
- [66] Marco Hutter, Michael Gehring, Michael Bloesch, C David Remy, Roland Yves Siegwart, and Mark A Hoepflinger. "StarLETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion." In: (2012).
- [67] Sang-Ho Hyon and Tsutomu Mita. "Development of a biologically inspired hopping robot-" Kenken"." In: *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*. Vol. 4. IEEE. 2002, pp. 3984–3991.
- [68] Mikhail S Jones. "Optimal control of an underactuated bipedal robot." In: (2014).
- [69] Mikhail S Jones and Jonathan W Hurst. "Effects of leg configuration on running and walking robots." In: *Proceedings of the 5th international conference on climbing and walking robots and the support technologies for mobile machines, Baltimore*. 2012, pp. 519–526.
- [70] Anders Jönsson, J Bathelt, and G Broman. "Implications of modelling one-dimensional impact by using a spring and damper element." In: *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 219.3 (2005), pp. 299–305.
- [71] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum." In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 2. IEEE. 2003, pp. 1644–1650.
- [72] Carl T Kelley. *Iterative methods for optimization*. SIAM, 1999.
- [73] Gavin Kenneally, Avik De, and Daniel E Koditschek. "Design principles for a family of direct-drive legged robots." In: *IEEE Robotics and Automation Letters* 1.2 (2016), pp. 900–907.
- [74] Ben Kenwright. "Inverse kinematics with dual-quaternions, exponential-maps, and joint limits." In: (2013).

- [75] Oussama Khatib. "Real-time obstacle avoidance for manipulators and mobile robots." In: *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.
- [76] Oussama Khatib. "Inertial properties in robotic manipulation: An object-level framework." In: *The international journal of robotics research* 14.1 (1995), pp. 19–36.
- [77] Sangbae Kim and Patrick M. Wensing. "Design of Dynamic Legged Robots." In: *Foundations and Trends® in Robotics* 5.2 (2017), pp. 117–190. DOI: [10.1561/23000000044](https://doi.org/10.1561/23000000044). URL: <http://dx.doi.org/10.1561/23000000044>.
- [78] Devin Koepl and Jonathan Hurst. "Force control for planar spring-mass running." In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 3758–3763.
- [79] Devin Koepl and Jonathan Hurst. "Impulse control for planar spring-mass running." In: *Journal of Intelligent & Robotic Systems* 74.3-4 (2014), pp. 589–603.
- [80] Evangelos Kokkevis. "Practical physics for articulated characters." In: *Game Developers Conference*. Vol. 2004. 2004.
- [81] Arthur D Kuo. "Choosing your steps carefully." In: *IEEE Robotics & Automation Magazine* 14.2 (2007), pp. 18–29.
- [82] Michael LaBarbera. "Why the wheels won't go." In: *The American Naturalist* 121.3 (1983), pp. 395–408.
- [83] Dominic Lakatos, Werner Friedl, and Alin Albu-Schäffer. "Eigenmodes of Nonlinear Dynamics: Definition, Existence, and Embodiment into Legged Robots With Elastic Elements." In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1062–1069.
- [84] Jimmy Kwok-Ching Lam. "An efficient simulated annealing schedule." PhD thesis. Yale University, 1989.
- [85] David V Lee and Sanford G Meek. "Directionally compliant legs influence the intrinsic pitch behaviour of a trotting quadruped." In: *Proceedings of the Royal Society of London B: Biological Sciences* 272.1563 (2005), pp. 567–572.

- [86] Jihong Lee. "A study on the manipulability measures for robot manipulators." In: *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*. Vol. 3. IEEE. 1997, pp. 1458–1465.
- [87] Sigrid Leyendecker. "Mechanical integrators for constrained dynamical systems in flexible multibody dynamics." PhD thesis. 2006.
- [88] Zexiang Li and Richard Montgomery. "Dynamics and optimal control of a legged robot in flight phase." In: *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE. 1990, pp. 1816–1821.
- [89] Zhibin Li, Nikos G Tsagarakis, and Darwin G Caldwell. "A passivity based admittance control for stabilizing the compliant humanoid COMAN." In: *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE. 2012, pp. 43–49.
- [90] Imed Mansouri and Mohammed Ouali. "The power manipulability—A new homogeneous performance index of robot manipulators." In: *Robotics and Computer-Integrated Manufacturing* 27.2 (2011), pp. 434–449.
- [91] William C Martin, Albert Wu, and Hartmut Geyer. "Robust spring mass model running for a physical bipedal robot." In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 6307–6312.
- [92] H-M Maus, SW Lipfert, M Gross, J Rummel, and A Seyfarth. "Upright human gait did not provide a major mechanical challenge for our ancestors." In: *Nature communications* 1 (2010), p. 70.
- [93] Tad McGeer. "Powered flight, child's play, silly wheels and walking machines." In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE. 1989, pp. 1592–1597.
- [94] Tad McGeer et al. "Passive dynamic walking." In: *I. J. Robotic Res.* 9.2 (1990), pp. 62–82.

- [95] Robert B McGhee and Geoffrey I Iswandhi. "Adaptive locomotion of a multilegged robot over rough terrain." In: *IEEE transactions on systems, man, and cybernetics* 9.4 (1979), pp. 176–182.
- [96] Andreas Merker, Dieter Kaiser, and Martin Hermann. "Numerical bifurcation analysis of the bipedal spring-mass model." In: *Physica D: Nonlinear Phenomena* 291 (2015), pp. 21–30.
- [97] Jean-Pierre Merlet. "Jacobian, manipulability, condition number, and accuracy of parallel robots." In: *Journal of Mechanical Design* 128.1 (2006), pp. 199–206.
- [98] Igor Mordatch. "Automated Discovery and Learning of Complex Movement Behaviors." PhD thesis. 2016.
- [99] Igor Mordatch, Emanuel Todorov, and Zoran Popović. "Discovery of complex behaviors through contact-invariant optimization." In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), p. 43.
- [100] Xiaohong Nian, Fei Peng, and Hang Zhang. "Regenerative braking system of electric vehicle driven by brushless DC motor." In: *IEEE Transactions on Industrial Electronics* 61.10 (2014), pp. 5798–5808.
- [101] Ryuma Niiyama and Yasuo Kuniyoshi. "Design principle based on maximum output force profile for a musculoskeletal robot." In: *Industrial Robot: An International Journal* 37.3 (2010), pp. 250–255.
- [102] Ionut Mihai Constantin Olaru, Chris Schmidt-Wetekam, Nicholas Payton, Gray Thomas, Johnny Godowski, Jerry Pratt, and Sebastien Cotton. "Mechanical Design of the FastRunner Leg." In: *HIP* 10.90 (), p. 100.
- [103] Kevin C Olds. "Global Indices for kinematic and force transmission performance in parallel robots." In: *Robotics, IEEE Transactions on* 31.2 (2015), pp. 494–500.
- [104] David E Orin and Ambarish Goswami. "Centroidal momentum matrix of a humanoid robot: Structure and properties." In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE. 2008, pp. 653–659.

- [105] David E Orin, Ambarish Goswami, and Sung-Hee Lee. "Centroidal dynamics of a humanoid robot." In: *Autonomous Robots* 35.2-3 (2013), pp. 161–176.
- [106] Toru Oshima, Tomohiko Fujikawa, Osamu Kameyama, and Minayori Kumamoto. "Robotic analyses of output force distribution developed by human limbs." In: *Robot and Human Interactive Communication, 2000. RO-MAN 2000. Proceedings. 9th IEEE International Workshop on*. IEEE. 2000, pp. 229–234.
- [107] F Ozguner, SJ Tsai, and RB McGhee. "An approach to the use of terrain-preview information in rough-terrain locomotion by a hexapod walking machine." In: *The International Journal of Robotics Research* 3.2 (1984), pp. 134–146.
- [108] Jim M Papadopoulos. "Forces in bicycle pedalling." In: *Biomechanics in Sport* (1987), pp. 26–33.
- [109] KV Papantoniou. "Electromechanical design for an electrically powered, actively balanced one leg planar robot." In: *Intelligent Robots and Systems' 91. Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*. IEEE. 1991, pp. 1553–1560.
- [110] Frank C Park, James E Bobrow, and Scott R Ploen. "A Lie group formulation of robot dynamics." In: *The International Journal of Robotics Research* 14.6 (1995), pp. 609–618.
- [111] Hae-Won Park, Koushil Sreenath, Jonathan W Hurst, and Jessy W Grizzle. "Identification of a bipedal robot with a compliant drivetrain." In: *IEEE Control Systems Magazine* 31.2 (2011), pp. 63–88.
- [112] Frank Peuker, Christophe Maufroy, and André Seyfarth. "Leg-adjustment strategies for stable running in three dimensions." In: *Bioinspiration & biomimetics* 7.3 (2012), p. 036002.
- [113] Michael Posa. "Optimization for Control and Planning of Multi-contact Dynamic Motion." PhD thesis. Massachusetts Institute of Technology, 2017.

- [114] Gill A Pratt and Matthew M Williamson. "Series elastic actuators." In: *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*. Vol. 1. IEEE. 1995, pp. 399–406.
- [115] Jerry E Pratt and Gill A Pratt. "Exploiting natural dynamics in the control of a planar bipedal walking robot." In: *Proceedings of the Annual Allerton Conference on Communication Control and Computing*. Vol. 36. UNIVERSITY OF ILLINOIS. 1998, pp. 739–748.
- [116] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. "Capture point: A step toward humanoid push recovery." In: *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE. 2006, pp. 200–207.
- [117] Marc H Raibert, H Benjamin Brown, and Michael Chepponis. "Experiments in balance with a 3D one-legged hopping machine." In: *The International Journal of Robotics Research* 3.2 (1984), pp. 75–92.
- [118] Marc H Raibert, H Benjamin Brown Jr, Michael Chepponis, Jeff Koechling, Jessica K Hodgins, Diane Dustman, W Kevin Brennan, David S Barrett, Clay M Thompson, John Daniell Hebert, et al. "Dynamically Stable Legged Locomotion (September 1985-September 1989)." In: (1989).
- [119] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, and Rob Playter. "Bigdog, the rough-terrain quadruped robot." In: *IFAC Proceedings Volumes* 41.2 (2008), pp. 10822–10825.
- [120] Jacob Reher, Eric A Cousineau, Ayonga Hereid, Christian M Hubicki, and Aaron D Ames. "Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS." In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1794–1801.
- [121] C David Remy, Keith Buffinton, and Roland Siegwart. "A matlab framework for efficient gait creation." In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 190–196.

- [122] Siavash Rezazadeh and Jonathan W Hurst. "On the optimal selection of motors and transmissions for electromechanical and robotic systems." In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE. 2014, pp. 4605–4611.
- [123] Siavash Rezazadeh and Jonathan W Hurst. "Toward step-by-step synthesis of stable gaits for underactuated compliant legged robots." In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 4532–4538.
- [124] Siavash Rezazadeh, Christian Hubicki, Mikhail Jones, Andrew Peekema, Johnathan Van Why, Andy Abate, and Jonathan Hurst. "Spring-mass walking with ATRIAS in 3D: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot." In: *ASME 2015 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2015, V001T04A003–V001T04A003.
- [125] Thomas J Roberts and Emanuel Azizi. "Flexible mechanisms: the diverse roles of biological springs in vertebrate movement." In: *The Journal of experimental biology* 214.3 (2011), pp. 353–361.
- [126] David W Robinson, Jerry E Pratt, Daniel J Paluska, and Gill A Pratt. "Series elastic actuator development for a biomimetic walking robot." In: *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*. IEEE. 1999, pp. 561–568.
- [127] Fredrik Roos, Hans Johansson, and Jan Wikander. "Optimal selection of motor and gearhead in mechatronic applications." In: *Mechatronics* 16.1 (2006), pp. 63–72.
- [128] Martin Rutschmann, Brian Satzinger, Marten Byl, and Katie Byl. "Nonlinear model predictive control for rough-terrain robot hopping." In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 1859–1864.
- [129] Uluc Saranli, Martin Buehler, and Daniel E Koditschek. "RHex: A simple and highly mobile hexapod robot." In: *The International Journal of Robotics Research* 20.7 (2001), pp. 616–631.

- [130] John Schmitt, Mariano Garcia, RC Razo, Philip Holmes, and Robert J Full. "Dynamics and stability of legged locomotion in the horizontal plane: a test case using insects." In: *Biological cybernetics* 86.5 (2002), pp. 343–353.
- [131] William J Schwind and Daniel E Koditschek. "Approximating the stance map of a 2-DOF monopod runner." In: *Journal of Nonlinear Science* 10.5 (2000), pp. 533–568.
- [132] Claudio Semini, Victor Barasuol, Thiago Boaventura, Marco Frigerio, Michele Focchi, Darwin G Caldwell, and Jonas Buchli. "Towards versatile legged robots through active impedance control." In: *The International Journal of Robotics Research* 34.7 (2015), pp. 1003–1020.
- [133] Sangok Seok, Albert Wang, Meng Yee Michael Chuah, Dong Jin Hyun, Jongwoo Lee, David M Otten, Jeffrey H Lang, and Sangbae Kim. "Design principles for energy-efficient legged locomotion and implementation on the MIT Cheetah robot." In: *IEEE/ASME Transactions on Mechatronics* 20.3 (2015), pp. 1117–1129.
- [134] André Seyfarth, Hartmut Geyer, and Hugh Herr. "Swing-leg retraction: a simple control model for stable running." In: *Journal of Experimental Biology* 206.15 (2003), pp. 2547–2555.
- [135] Zhuohua Shen and Justin Seipel. "A spring-mass model of locomotion with full asymptotic stability." In: *ASME 2011 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers. 2011, pp. 301–306.
- [136] Zhuohua Shen and Justin Seipel. "A Piecewise-Linear Approximation of the Canonical Spring-Loaded Inverted Pendulum Model of Legged Locomotion." In: *Journal of Computational and Nonlinear Dynamics* 11.1 (2016), p. 011007.
- [137] Manuel F Silva and JA Tenreiro Machado. "A historical perspective of legged robots." In: *Journal of Vibration and Control* 13.9-10 (2007), pp. 1447–1486.
- [138] Peng Song. "Modeling, analysis and simulation of multibody systems with contact and friction." PhD thesis. University of Pennsylvania, 2002.
- [139] Shin-Min Song and Jong-Kil Lee. "The mechanical efficiency and kinematics of pantograph-type manipulators." In: *KSME Journal* 2.1 (1988), pp. 69–78.

- [140] James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. Vol. 65. John Wiley & Sons, 2005.
- [141] Manoj Srinivasan and Andy Ruina. "Computer optimization of a minimal biped model discovers walking and running." In: *Nature* 439.7072 (2006), pp. 72–75.
- [142] Hai-Jun Su. "A pseudorigid-body 3R model for determining large deflection of cantilever beams subject to tip loads." In: *Journal of Mechanisms and Robotics* 1.2 (2009), p. 021008.
- [143] NA Titus and CH Spenny. "Power metrics for robot planning and redundancy resolution." In: *Intelligent Control, 1994., Proceedings of the 1994 IEEE International Symposium on*. IEEE. 1994, pp. 153–159.
- [144] Emanuel Todorov. "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo." In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 6054–6061.
- [145] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control." In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 5026–5033.
- [146] Nikos G Tsagarakis, Stephen Morfeý, Gustavo Medrano Cerda, Li Zhibin, and Darwin G Caldwell. "Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control." In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 673–678.
- [147] James R Usherwood and John EA Bertram. "Understanding brachiation: insight from a collisional perspective." In: *Journal of Experimental Biology* 206.10 (2003), pp. 1631–1642.
- [148] Hamid Reza Vejdani Noghreiyani. "Control of spring-mass running robots." In: (2013).
- [149] Hamid Reza Vejdani, Albert Wu, Hartmut Geyer, and Jonathan W Hurst. "Touch-down angle control for spring-mass walking." In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 5101–5106.

- [150] Miomir Vukobratović and Branislav Borovac. “Zero-moment point—thirty five years of its life.” In: *International journal of humanoid robotics* 1.01 (2004), pp. 157–173.
- [151] KJ Waldron and GL Kinzel. “The relationship between actuator geometry and mechanical efficiency in robots.” In: *Fourth, symposium on Theory and Practice of Robots and Manipulators. Poland* (1981).
- [152] Handing Wang and Xin Yao. “Corner sort for Pareto-based many-objective optimization.” In: *IEEE transactions on cybernetics* 44.1 (2014), pp. 92–102.
- [153] Patrick M Wensing. “Optimization and control of dynamic humanoid running and jumping.” PhD thesis. 2014.
- [154] Patrick M Wensing, Albert Wang, Sangok Seok, David Otten, Jeffrey Lang, and Sangbae Kim. “Proprioceptive actuator design in the MIT cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots.” In: *IEEE Transactions on Robotics* (2017).
- [155] Patrick Wensing, Roy Featherstone, and David E Orin. “A reduced-order recursive algorithm for the computation of the operational-space inertia matrix.” In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 4911–4917.
- [156] Yonas Gebre Woldesenbet, Gary G Yen, and Biruk G Tessema. “Constraint handling in multiobjective evolutionary optimization.” In: *IEEE Transactions on Evolutionary Computation* 13.3 (2009), pp. 514–525.
- [157] Aimin Wu and Hartmut Geyer. “The 3-d spring–mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments.” In: *Robotics, IEEE Transactions on* 29.5 (2013), pp. 1114–1124.
- [158] Ming-Ji Yang, Hong-Lin Jhou, Bin-Yen Ma, and Kuo-Kai Shyu. “A cost-effective method of electric brake with energy regeneration for electric vehicles.” In: *IEEE Transactions on Industrial Electronics* 56.6 (2009), pp. 2203–2212.
- [159] Tsuneo Yoshikawa. “Manipulability of robotic mechanisms.” In: *The international journal of Robotics Research* 4.2 (1985), pp. 3–9.

- [160] Petr Zaytsev, S Javad Hasaneini, and Andy Ruina. “Two steps is enough: no need to plan far ahead for walking balance.” In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 6295–6300.
- [161] Garth John Zeglin. “Uniroo—a one legged dynamic hopping robot.” PhD thesis. Massachusetts Institute of Technology, 1991.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst’s seminal book on typography “*The Elements of Typographic Style*”. `classicthesis` is available for both L^AT_EX and L^yX:

<http://code.google.com/p/classicthesis/>

Final Version as of May 22, 2018 (`classicthesis` version 4.2).