Design, Construction, and Experiments with a Compass Gait Walking Robot

by

Zachary J Jackowski

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author . . . .
Department of Mechanical Engineering
May 6, 2011

Certified by . . . .
Russell L Tedrake
Associate Professor
Thesis Supervisor

Certified by . . . .
Sangbae Kim
Assistant Professor
ME Faculty Reader

Accepted by . . . .
David E. Hardt
Chairman, Department Committee on Graduate Theses

Design, Construction, and Experiments with a Compass Gait Walking Robot

by

Zachary J Jackowski

Submitted to the Department of Mechanical Engineering
on May 6, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

In recent years a number of new computational techniques for the control of nonlinear and underactuated systems have been developed and tested largely in theory and simulation. In order to better understand how these new tools are applied to real systems and to expose areas where the theory is lacking testing on a physical model system is necessary. In this thesis a human scale, free walking, planar bipedal walking robot is designed and several of these new control techniques are tested. These include system identification via simulation error optimization, simulation based LQR-Trees, and transverse stabilization of trajectories. Emphasis is put on the topics of designing highly dynamic robots, practical considerations in implementation of these advanced control strategies, and exploring where these techniques need additional development.

Thesis Supervisor: Russell L Tedrake
Title: Associate Professor

ME Faculty Reader: Sangbae Kim
Title: Assistant Professor

Acknowledgments

I'd like to thank Russ Tedrake for support, both in resources and direct help with this project in the form of countless discussions and late night code contributions. I'd also like to thank Anirudha Majumdar for a summer of great help getting the robot take its first steps and Amy Qian for her help early on with design work and fabrication of the robot's foot actuators. A great deal of thanks is also deserved by everyone else at the Robot Locomotion Group, Michael Levashov for lots of help with dynamics and modeling, Ian Manchester for the many consultations on control theory and direct help with making it work, John Roberts for many useful discussion both on robotics and everything else in life and everyone else for making my years at the lab so enjoyable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As a robot designer and fabricator I come to the area of controls mostly as a consumer but working closely with real theorists. The Robot Locomotion Group has been producing a tremendous amount of promising new ideas in the field of applied nonlinear control, specifically LQR Trees [19] and transverse stabilization [13]. While mathematically well grounded results are important one must look to the actual purpose of the research in order to see that it's only half the story. The other half is how well those techniques work to solve real problems with real hardware and whether they can be implemented successfully and efficiently. This is the end that the compass gait walker project has been working toward. We as a research lab believe that all of the tools required to effectively solve the control of the compass gait currently exist, especially with the development of our LQR Trees method and wanted to develop a hardware platform that can show that fact without a doubt.

The work involved in implementing a full demonstration of our methods on real hardware is also important in and of itself. Previously a great amount of effort has gone into making our methods theoretically sound and working in simulation, but the real goal is almost always making real physical systems work better. Making this happen brings to light a wide variety of new considerations such as how the control theory fits in software system architecture, the constraints of running in real time, and the amenability of common physical systems to the precise modeling our methods require. It is by design that difficulties in this project drive the development of our work in the future, if not explicitly finding ways to deal with them, knowing of their importance and severity while investigating new methods to steer us away from fragile theories.

While the goals stated above apply for the project as a whole, my goal in this thesis is to record as many of the little bits that fall through the cracks of documentation as possible along with the big ideas. I've been surprised and saddened to find a general absence of design information for those getting started in building highly dynamic robots and lot of effort is lost learning the right order of priorities and rules of thumb. While in design there's no replacement for personal experience, knowing pitfalls ahead of time can mean the success of an entire project.

I think it's safe to say that any robot designer knows, as a few trivial examples, to make structures lightweight, to validate what can be before committing to build, and to keep control loop delays small. How these individual factors should play in an ecosystem of many competing priorities is usually completely unknown. The designer's art is to figure out how to spread the resources available around to produce something that works. Figuring out the right places to break the bank can mean the difference between a successful ten thousand dollar robot and a failed fifty thousand dollar robot. The fact is that when you're trying to push the boundaries of control something as simple as a bargain bin inertial measurement unit can keep you from solving the problems that actually matter. The same goes for time of course, a week spent simulating the most critical parts of a robot could save months effort later, not only from the first order effects of having to fix what's wrong, but the second order effects of wasted time leading up to deciding band-aid fixes won't cut it and that something needs to be fixed in the first place. I hope to provide an artifact of my design process and how I balanced these competing factors.

Designers work in a different currency from most scientists and engineers because their products are different. The working machine that's produced may not be a great intellectual work, but it's often the real world outlet and test of those works. I hope that the reader will come away from this not so much with an intimate understanding of all the wonderful things I've discovered, but with a feel for the design process of this kind of machine and the kinds of control methods that can bring it to life. Hopefully the theorists whose work I've drawn on will also be able to see the spots that are still sore and find inspiration for new work in them.

Why we chose to make a simple planar walking robot is an important topic when it seems like similar things have been done before [10] [22] [5] [11] and even more impressive walking robots have been demonstrated [8] [4] [2] [7]. We feel strongly that in order to understand the real theoretical problems at the heart of these complicated control problems concisely the target system needs to be exactly as complicated as necessary and no more so. The free walking compass gait concept takes a scalpel to complexity and reduces the physical system to the minimum necessary to bring forth all the problems we think matter. It represents highly nonlinear hybrid dynamics with realistic but simple ground contact. It also brings the issues of managing a full robotic system such as carrying its own computation resources into the picture without overshadowing what is going on at the lowest levels.

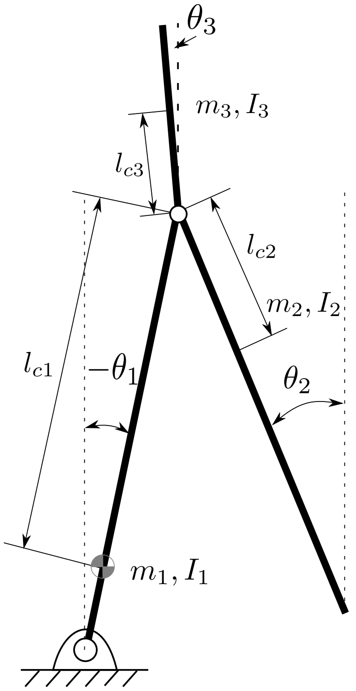

Figure 1-1: The compass gait with a body is a very simple model of walking which still reflects all the reasons why walking is a difficult problem. The model can be fully described by the continuous dynamics of these three links along with an instantaneous ground contact which switches the stance leg.

A system as complicated as ASIMO or BigDog with a very large state adds little of value as far as understanding goes while the system designed here with its small state and number of actuators still representing all the important problems that need to be solved on the control front. Building the robot in-house puts the expertise with the mechanical and software system in direct contact (often in the same person) with the people developing the control ideas that make it work which is important in highly dynamic systems such as this where things like loop delays and modelability can make or break experiments. The robot developed here is also in a performance regime that hasn't been entered by a similar robot before. Careful attention to maximizing actuator performance and ideality has enabled the boundaries of what the system is capable of to be pushed as far out as possible while working within the constraints of an academic lab.

Chapter 2

Physical System Design

With a relatively simple concept and purpose for the robot we thought the actual design of the robot would be very straightforward. The lab already had built a much smaller compass gait walker and I had personally built an acrobot, so we knew more than a few key design points for the robot. There's always more to learn, especially after making a jump in complexity as large as this one and we certainly learned a lot in the process. Hopefully the most important points will all be remembered and recorded here.

2.1 Robot Design

I mean to make a point with the layout of this chapter, that robot design is the most important design issue in a robot. The defining characteristic of a robot is the integration of sensing, actuation, and decision making into a singular fluid system. The mechanical parts of the robot are inseparable in design from the electrical system that brings them to life and the software systems that breathes intelligence into it all.

As a simple example of this, when we designed the acrobot the position encoder for the second link was originally on the back end of the motor, a very clean design decision that integrates the motor assembly and protects the encoder well. Previous experience had shown that kit encoders which rely on user provided bearings had been big sources of problems because of misalignment causing missed counts. Encoders that

Figure 2-1: The full robot, a free walking realization of the simple compass gait model with a body. The robot is a planar walker but has three legs and four feet, with good reason.

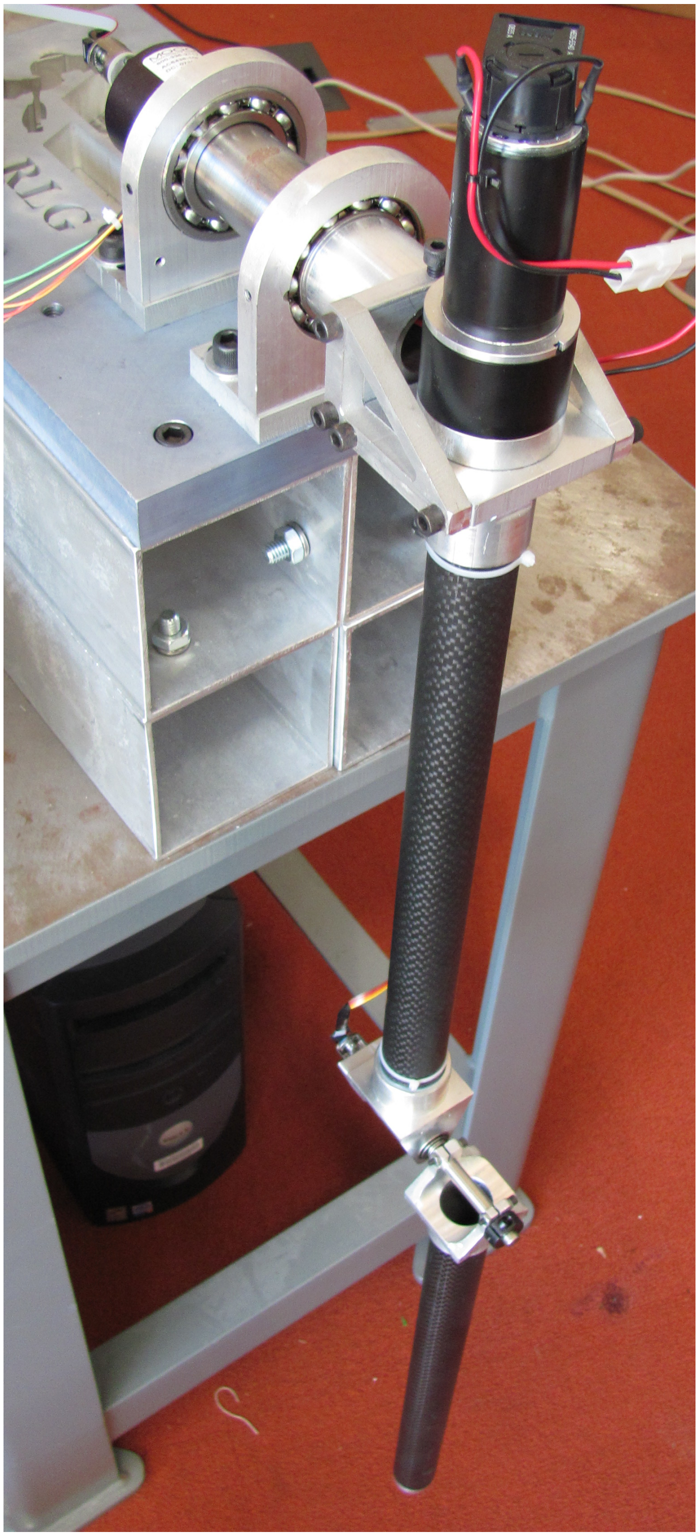

Figure 2-2: The acrobot built previous to the design of the compass gait walker. The motor on the first link is connected to the elbow by a long driveshaft. The encoders in question are located on the back of the motor at the top and on the elbow near the bottom of the picture.

integrate their own bearings and housing are much heavier and expensive, so using the encoder on the back of the motor is a big advantage, or at least seemed to be so until we realized the error we had made. The two foot long carbon fiber driveshaft that connected the motor at the shoulder of the robot to the elbow introduced a small amount of compliance which caused extreme issues for the high-gain control required to balance the arm around the upright position. The compliance of the shaft, combined with the small amount of backlash in the right angle gearbox at the elbow introduced extra dynamics that made control of the system near impossible. In order to make headway in control of the robot we had to sacrifice the integration of the encoder and move it down to the elbow on the other side of the gearbox. In addition to the compromise made in the beauty of the design, we also had to sacrifice some money in buying an encoder with its own integrated bearing set.

One of the most important takeaway lessons from that sensor change is that modeling, control, and advanced sensing strategies are rarely the right solution for a problem that can be designed away from the start. This is a surprisingly difficult lesson to internalize, especially surrounded by people very good at these methods, and it took most of the compass gait project to get all the way there. Elements of this can be seen in almost every major design change on the robot if you look closely, for example modifications to the hip gearbox and the location of the middle toes which be looked at closely in their respective sections.

Structural components make up most of the 'robot' by weight and volume, but it's important to remember that they're the facilitators of the robot's function. In the case of a walking robot this is a little bit strange because the main function of the machine is to push the limits of what has been demonstrated by walking robots before, a nebulous goal that doesn't outwardly say anything about what the functional requirements are. As with many design problems the target specifications aren't known and may never be, the best we can do is make successive approximations and prototypes.

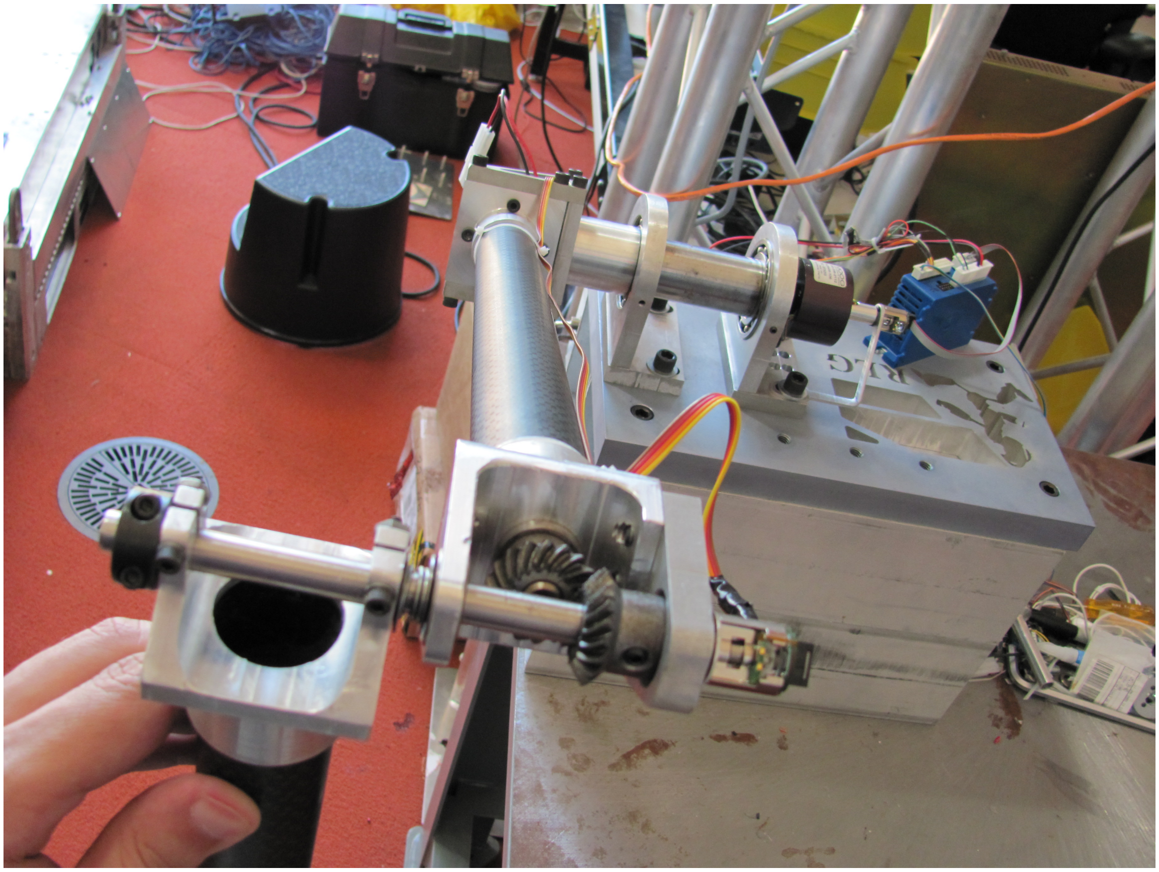

Figure 2-3: One of the several acrobot elbow iterations. In this case helical miter gears are used to achieve smoother operation along with a spring between the two links pushing them apart. This light preload keeps the gears tightly meshed which minimizes backlash while avoiding binding. The integrated encoder can be seen hanging off the backside.

2.1.1 Overall Concept

The initial germs of the idea to build this robot came from a few sources and past projects. The acrobot I had designed for the lab previously was the precursor for many of the design details and lessons, but our work in theory and on our small scale compass gait robot are where the desire for it actually came from. We would have kept working with this small robot, but it had a couple flaws.

Most important was the boom. Originally the robot was put on a boom in order to keep it from falling over sideways, but the boom turned out to have another bad effect: it allowed the robot's key inertias and masses to be changed at will. When the robot was originally built it wasn't actually able to walk effectively until weight was added to the end of the boom on the other side of the fulcrum from the robot. This can be seen in the system parameters provided with one of the papers on experiments with the robot [10], the counterweight provided almost enough force to cancel out gravity, providing a moon-like bounce and slowness in the robot's steps. The time constants associated with the robot falling over without the boom were too fast to be worked with with the available actuators and sensing. This produced experiments which were overly optimistic and appeared nonphysical.

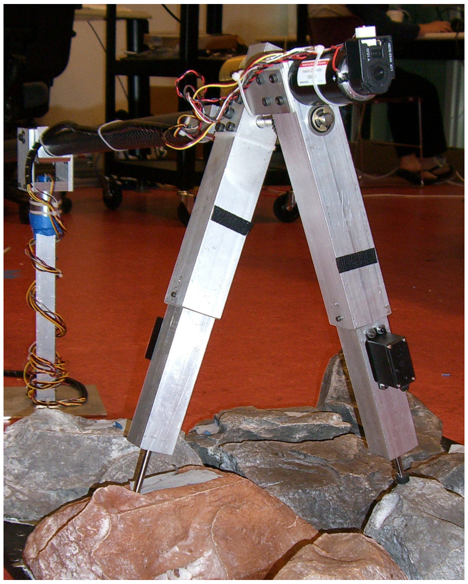

Figure 2-4: The lab's old compass gait walker on its boom.

This robot design works to deal with all the issues that the boom was used to compensate for so that it could be eliminated, producing a much more honest and believable demonstration, over more impressive terrain.

One way to change the inertial characteristics of the system without making the legs heavier and therefore more difficult to move themselves is to introduce a bisecting body to the robot. The example of this that the most inspiration was drawn from was the robot 'Max' from Martijn Wisse et al at TU Delft [22]. Because the body only travels half the angle that a leg does when its moving it only appears half the size dynamically to the leg, but because the falling over action involved both legs moving together the body's mass is fully represented in that portion of the dynamics. Adding a bisecting body was also important because we planned for the new robot to be fully autonomous, requiring it to carry its own power and computation equipment. It's advantageous to locate all this mass in the body for the same inertial reasons, helping keep the legs as easy to move as possible.

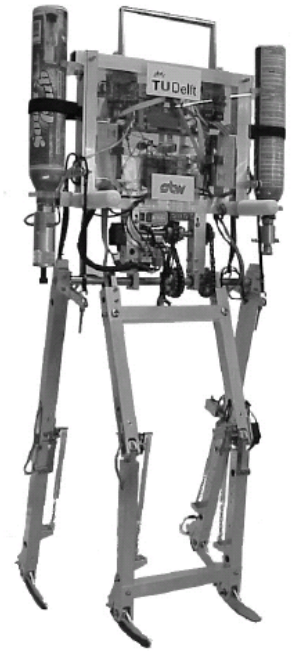

Figure 2-5: 'Max', an example of a dynamic walking robot with a bisecting body. [22]

In deciding to make a planar dynamic walker one of the other main design questions is whether the robot should have knees. To us this was as much a question of research philosophy as mechanical design. In order to avoid hitting the ground as the swing leg moves it must get shorter somehow. In most animals and on many walking robot this is accomplished with active or passive knees, making the leg to break during swing allows the distance between the hip and toe to get shorter. This action, however, greatly increases the complexity of the dynamic model (remember that point about the integration of design and control?). The canonical compass gait model only deals with one collision, impact of the swing toe with the ground, because the robot is assumed to be symmetric. Knees introduce a new hybrid transition into the model: knee locking (knee unlocking usually happens in conjunction with impact). Because this robot will be asymmetric this means that the introduction of knees would change the system from two modes with two transitions, to at least four modes with four transitions, many more if one wishes to account for breaking the knee of the stance leg or impacting with a broken knee. In many cases this wouldn't be much of a concern because the necessary simulation would be performed by an automatic dynamics solver, but we prefer to keep the number of plants as small as possible in order to keep analysis clean. The simpler the analysis the system is the more we likely we are to understand it at a fundamental level.

The alternative to this is prismatic feet, where the feet are actuated in and out parallel to the leg, something that doesn't happen often in biological systems. As long as the actuators don't hit their limits then the number of plant modes and transitions doesn't increase and the complexity of the continuous plant modes doesn't increase too badly. Prismatic feet also have the advantage of being able to push off using the same actuator which would be impossible with a passive or clutch based knee.

The overall size of the robot is one of the really free variables in the design. There are a couple considerations that push it to be large: computation and system time constants. Ideally the robot should be able to carry all the computer power it needs to function, which we decided up front to be a mini-ITX based computer running a modern x86 processor which means the robot's body will at least need to accommodate the volume and weight of the motherboard and batteries to feed a 65W processor. The mechanical time constants determine how fast and precise the control action needs to be, getting easier with the robot getting larger. This should intuitively make sense thinking of the simplest case, the point mass pendulum, whose natural frequency is $\omega = \sqrt{\frac{g}{l}}$. On the other hand, the smaller the robot, the smaller the motors required will be, along with being more safe and less expensive. We decided the scale of an adult human is a nice compromise between these competing objectives, with one more advantage: it's easy to take measurements of adult humans in order to help nail down further design aspects. This scale helps set reasonable expectations for walking and running speed and is also visually impressive when demonstrated in person. While humans don't normally perform the compass gait, it isn't too hard to imitate it for the purpose of collecting some rough data.

Much of the initial rollout of the specifications that individual components were built to was based on an initial guess for the weights of all the different robot parts. This is one area where some design intuition is extremely important. The process of doing things like picking how powerful the hip actuator should be is iterative, it's impossible to know exactly what will be expected of the actuator until the entire robot is designed, and even worse, functioning. Much of the intuition in designing a system like this comes down to having a feel for how much assemblies should weigh before detailed design has been done, in essence seeding the design optimization problem. There are also a lot of constraints that can be exploited if you know what to look for. For example, we know going into the design that the robot will need to carry a mini-ITX form factor computer, various standard sensors, and a battery pack to power at least the computer. These weights can help a lot to set the scale of the design problem.

2.1.2 Inertial Measurement Unit

Because the robot isn't fixed to the ground it's surprisingly difficult to figure out the current position of all its links. The angle between the legs is simple to measure, but figuring out the angle of the legs relative to gravity is more difficult and extremely important because control of the robot depends on knowing where the robot is and how the dynamics are moving. If the robot were on a boom this would not be a concern because the boom would be able to provide a reference to the ground. If the robot was always in contact with the ground it would be possible to attach a plate to the toes that could measure the ground angle, but that option falls apart if the robot has an aerial phase where neither foot is in contact with the ground.

An inertial measurement unit or IMU uses a collection of accelerometers, gyroscopes, and often magnetometers to produce orientation estimates without a ground reference besides gravity and an initial zeroing. A tremendous amount of development has gone into these systems due to interest from military and commercial aerospace applications and even a protracted discussion is well beyond the scope of this paper. This also means that there's a diverse field of off the shelf hardware available for the job.

There's a wide divide between the top of line MEMS IMUs available and the the bottom end of the next step up, laser ring based units which offer much better accuracy and resolution. The smallest laser ring based IMUs are designed for large aircraft or wheeled ground based vehicles and still weigh several kilograms and cost tens of thousands of dollars as they're largely targeted at military and aerospace applications so the choice here is easy, the top of the line MEMS device.

The best MEMS based IMU available at the time is the Microstrain 3DM-GX3 [14], which costs about \$1.9k and weighs 18 grams. According to the sensor specifications it has an accuracy of 2 degrees in dynamic conditions and 0.5 degrees in static conditions which is strikingly close to what we experienced in actual use. Most of the inaccuracy is in slow drift which can be pulled out using extra sensing with a little effort which we ended up needing to do for the balancing experiment. The other major nonideality we experienced is that the orientation estimate lags quite badly in fast motions, for example, taking about a second to settle after a 90 degree sensor rotation at 300 degrees per second. This turned into a nagging issue as we worked on control for walking.

2.1.3 Hip Sensing

Measuring the angle between the two legs is plainly a place for a shaft encoder, but there are thousands of encoders available. Absolute, relative, transmissive, reflective, magnetic? We need to look at the functional requirements: Reliability, resolution, interface, minimum volume and weight, availability, and cost.

Reliability first, because without being able to trust the most basic of sensors there really is nothing that can be accomplished. Every other requirement really can be pushed, we can always carve out a little bit of weight elsewhere, but lost ticks means unpredictable behavior. That's worse than no behavior. Previous experience indicates that encoders that rely on user-implemented bearings to constrain the code wheel are susceptible to lots of problems. Misalignment between the code wheel and shaft axes will produce wobbling over the reading head resulting in sporadic lost ticks, axial alignment with distance misalignment will produce lost ticks more consistently. The worst case is that these misalignments result in the code wheel crashing into the read head which will permanently damage it. The solution to all of these issues is to use an encoder with its own integrated shaft and bearing set.

Second, resolution, is relatively easy to deal with. Encoders are usually available in a wide range of resolutions, but the higher it goes the more fragile and expensive the encoder gets because the lines that must be sensed are so much smaller. Once they get small enough even a piece of dust can wreak havoc on reliability. For that reason it's smart to pick a resolution that isn't excessive compared to its purpose. So how does the resolution of an encoder relate to the robot it's attached to? It's hard to tell how much resolution is needed in position to accomplish the control objectives, but it's pretty easy to figure out what level won't be necessary in a feasible system. The best IMU available before moving into the laser ring range, coincidently the one selected for this robot already, has an orientation repeatability of 0.2 degrees[14] which translates to 1800 counts per revolution. Because the position estimate for the whole robot is driven off the IMU the encoder will effectively inherit its flaws, so anything much more than that is wasted, at least in terms of position.

It would be nice if there were enough ticks to go around to provide a fine estimate of velocity too. The way this is usually done is by looking at the difference in position over some very small amount of time, either the control loop rate, or faster than that if it's done on lower level hardware, and then applying a low pass filter to smooth out the signal. Figuring out this specification requires knowing a little bit about the velocities the robot will be seeing and how the actual encoder velocity estimate is produced. Because of the uncertainty of all of these pieces (and the fact that in the end we won't even actually use the encoder's velocity estimate thanks to better methods) we'll leave this one at just the mention of it.

When the encoder is maintained as a closed structure it maintains the accuracy that it passed quality assurance with at the factory. As the encoder shaft is fully constrained with respect to the reader structure in all degrees of freedom except a single rotation it's important not to overconstrain it when mounting the encoder to the robot. This is almost always accomplished with a flexure bracket. When the encoder is attached to the shaft to measure, usually by a combination of a loose slip fit and a set screw the whole assembly loses all of its translational degrees of freedom. The job of the bracket is fix the angular position of the encoder body while allowing it a small amount of motion in the two other rotational directions. An appropriate design for the bracket is shown in Figure 2-6.

Figure 2-6: Encoder flexure bracket.