

A Low Cost Modular Actuator for Dynamic Robots

by

Benjamin G. Katz

S.B., Massachusetts Institute of Technology (2016)

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

Masters of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Author
Department of Mechanical Engineering
May 11, 2018

Certified by
Sangbae Kim
Associate Professor
Thesis Supervisor

Accepted by
Rohan Abeyaratne
Quentin Berg Professor of Mechanics, Graduate Officer

A Low Cost Modular Actuator for Dynamic Robots

by

Benjamin G. Katz

Submitted to the Department of Mechanical Engineering
on May 11, 2018, in partial fulfillment of the
requirements for the degree of
Masters of Science in Mechanical Engineering

Abstract

This thesis details the hardware and control development for a low-cost modular actuator, intended for use in highly dynamic robots. A small 12 degree of freedom quadruped robot has been built using these actuators, on which several control experiments have been performed. Despite the relatively low cost of the actuators, the quadruped has demonstrated unprecedented dynamic behaviors for a robot of this scale and number of degrees of freedom, such as a full 360° backflip from standing on flat ground. Several other implementations of these actuators are also discussed, including bilateral teleoperation and haptic feedback system and a 6 degree of freedom lower-body biped robot.

Thesis Supervisor: Sangbae Kim
Title: Associate Professor

Acknowledgments

Many people helped to make this work possible. In particular I would like to thank:

My advisor, Sangbae Kim: for taking me as a UROP after my freshman year at MIT, getting me started down this path; for the freedom and resources to pursue this project; and everything else.

Jared Di Carlo, for the amazing work on the software and controls for both this robot and Cheetah 3, and for countless hours of debugging and fixing Cheetah 3 with me.

João Ramos for being an actuator guinea pig and trying these out on Little Hermes.

Bayley Wang, for many motor control discussions, help outsourcing machining, and wrangling Yubo MFG.

Alex Hattori, for help with design and fabrication of the quadruped.

Everyone else in the Biomimetic Robotics Lab: This has been an fantastic environment to work in. Thanks for making it awesome.

The wonderful people of the MIT Electronics Research Society. Thank you all for the inspiration, technical advice, friendships, and many ill-conceived adventures.

Andy B. for keeping me company during all those late nights of machining robot parts, assembling PCBs, writing this thesis, and so on.

And Ava Chen. Hey, Ava: You're great.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 17 |
| 1.1 | A Brief Overview of Actuation for Dynamic Legged Robots | 18 |
| 1.1.1 | Proprioceptive Electric Actuators | 19 |
| 1.1.2 | Hydraulic Actuators | 20 |
| 1.1.3 | Series Elastic Actuators | 20 |
| 1.2 | Existing Modular Robot Actuators | 21 |
| 2 | Modular Actuator | 23 |
| 2.1 | Performance Requirements | 24 |
| 2.2 | Mechanical Design | 26 |
| 2.2.1 | Electric Motor | 26 |
| 2.2.2 | Housing and Planetary Gearbox | 27 |
| 2.2.3 | Design for Impact: Estimating Transmission Loading During Collisions. | 30 |
| 2.2.4 | Bearing Loads | 34 |
| 2.3 | Motor Control Hardware | 37 |
| 2.3.1 | 3-Phase Inverter | 37 |
| 2.3.2 | Logic Power, Gate Drive, Current Sensing, and Microcontroller | 40 |
| 2.3.3 | Position Sensing | 40 |
| 2.4 | Control | 41 |
| 2.4.1 | Current and Torque Control | 41 |
| 2.4.2 | Position Sensor Calibration and Linearization | 45 |
| 2.4.3 | Actuator configuration and communication | 47 |

| | | |
|-------------------|--|-----------|
| 2.5 | Actuator Testing and Characterization | 49 |
| 2.5.1 | Steady-State Performance | 49 |
| 2.5.2 | Torque Accuracy | 51 |
| 2.5.3 | Thermal Analysis | 53 |
| 3 | Quadruped Platform | 57 |
| 3.1 | Mechanical Design | 58 |
| 3.1.1 | Legs | 58 |
| 3.1.2 | Body | 65 |
| 3.1.3 | Feet | 67 |
| 3.1.4 | Wiring | 68 |
| 3.2 | Electronics and System Architecture | 69 |
| 3.2.1 | Embedded Linux Computer | 69 |
| 3.2.2 | SPIne | 70 |
| 3.2.3 | Battery and Power Supplies | 70 |
| 3.3 | Control Experiments | 74 |
| 3.3.1 | Flipping | 74 |
| 4 | Other Actuator Implementations | 81 |
| 4.1 | Bilateral Teleoperation and High-Force Haptics | 81 |
| 4.1.1 | Hardware | 81 |
| 4.1.2 | Bilateral Teleoperation | 82 |
| 4.1.3 | Virtual Environments | 83 |
| 4.2 | Biped for Human-Robot Balance Feedback | 85 |
| 5 | Conclusion | 87 |
| Appendix A | Code and Design Files | 93 |
| Appendix B | Videos | 95 |
| Appendix C | Discrete Time Current Control | 97 |
| C.1 | System Model in Continuous and Discrete Time | 97 |

| | |
|--|-----|
| C.2 Controller Design | 99 |
| C.3 Implementation in Code | 102 |
| C.4 MATLAB Gain Calculating Script | 103 |

List of Figures

| | |
|---|----|
| 2-1 Assembled actuator. | 24 |
| 2-2 Exploded view of the actuator. | 26 |
| 2-3 Torque vs current (Left) and torque constant vs current (Right). The motor exhibits some saturation, with torque constant dropping by 12% at maximum current. | 27 |
| 2-4 Stator (left) and rotor (right) of the brushless motor used in the actuator. The rotor has been modified to fit the sun gear of the planetary transmission. | 28 |
| 2-5 Cross-sectional views of the actuator. On the left, bearings are highlighted in red, rotor in blue, and planet carrier in green. | 29 |
| 2-6 Machined components of the housing and planetary gearbox | 29 |
| 2-7 Case 1: Actuator input compliance | 31 |
| 2-8 Case 2: Actuator output compliance | 32 |
| 2-9 Case 3: End effector compliance | 32 |
| 2-10 PCB layout, front, and back of the inverter. | 38 |
| 2-11 Estimated steady-state MOSFET temperature vs peak phase current. | 39 |
| 2-12 View of the actuator with the housing open. The encoder IC which measures rotor position can be seen at the center of the bearing on the left half of the housing. The diametrically magnetized magnet it uses is at the center of the rotor on the right. | 41 |

| | |
|---|----|
| 2-13 View of the connectors to the actuator. Two XT-30 connectors are used for DC input power, and two 3-Pin Molex SPOX connectors are used for CAN. Each pair of connectors is in parallel, so actuator modules can be easily daisy-chained. | 42 |
| 2-14 Phase current step response for 10 and 20 Amp steps, measured by external current probe. Current rise time is 75 μ s for 10A, and 110 μ s for 20A due to motor inductance and voltage limitation. | 43 |
| 2-15 Maximum torque vs speed (Left) and power vs speed (Right), with and without field weakening. | 45 |
| 2-16 1: Rotor angle and voltage vector angle during calibration. 2: Rotor angle minus voltage vector angle. 3: Averaged forwards and backwards error, with and without cogging torque FIR filter. 4: Calibration lookup table | 46 |
| 2-17 Custom 4-quadrant motor dynamometer used to measure steady-state performance and torque accuracy. | 50 |
| 2-18 Efficiency Map | 51 |
| 2-19 Loss Map | 52 |
| 2-20 Input Power Map. The red contour indicates the zero-power operating points. Outside this region, the actuator sinks electrical power, and inside this region the actuator sources power. | 53 |
| 2-21 Measured output torque vs torque command, averaged over several rotations, for both positive and negative work. | 54 |
| 2-22 Measured rotor cogging torque v s electrical angle. | 55 |
| 2-23 Thermal camera images at 5A (left) and 12.5A (right) phase currents. Colors do not correspond between images, as the temperature-color mapping automatically scales. | 55 |
| 2-24 Winding temperature during a 10A step. | 55 |
| 3-1 CAD Rendering of the quadruped | 58 |
| 3-2 CAD Diagram of one leg | 60 |

| | | |
|------|--|----|
| 3-3 | A variety of configurations possible with the wide range of motion of the legs. Top Left: Turn-on configuration. In this configuration the legs are correctly zeroed for turn-on. Top Right: The robot can be folded up for transportation. Bottom Left: The ab/ad joints have enough range of motion for the robot to walk with the body rolled $\pm 90^\circ$. Bottom Right: The legs can be lifted above the body, for climbing very tall obstacles or manipulation. | 61 |
| 3-4 | Completed lower link with bearings(Left), and cross section (Right) . . | 62 |
| 3-5 | One half of the upper link (Top), and the two assembled halves (Bottom) | 62 |
| 3-6 | Belt path through the upper link | 63 |
| 3-7 | Assembled knee joint. Cross section (Bottom) with the knee axle and bearing spacer in dark blue, bearings in red, upper link in light blue, lower link in orange, and belt in black | 64 |
| 3-8 | Abduction to hip link (Left), attached to both actuators (Right) . . . | 64 |
| 3-9 | Hip to leg link | 65 |
| 3-10 | Body components made from waterjet-cut and bent 1.5 mm 6061 sheet | 66 |
| 3-11 | Inside of the assembled body, showing the internal frame-stiffeners, battery mount, and handle on top. | 66 |
| 3-12 | Foot, and cross section(Left). The plastic insert has holes which are filled by the casting foam. | 68 |
| 3-13 | Wiring harness for one leg. Flexible wires are housed within protective corrugated sheaths, which are fixed by 3D printed guides at each end, strain-relieving the connectors. | 69 |
| 3-14 | Installed wiring harness for one leg. | 69 |
| 3-15 | Diagram of the robot's communication architecture | 71 |
| 3-16 | PCB Layout of the SPIne, on the left, and assembled board on the right. | 71 |
| 3-17 | SPIne and UP board assembled together | 72 |

| | |
|---|----|
| 3-18 Precharge circuit for turning on battery power. When the power switch is closed, a 12V supply is turned on which charges the gate of MOSFET Q1, turning it on. This allows the output capacitance C_{out} to be charged through power resistor R1, limiting the charging current to a maximum of V/R_1 . After a fixed delay set by the time constant of the RC filter made by R2, R3, and C1, the output of the comparator switches on, turning on MOSFET Q2, which bypasses the power resistor. Flyback diode D1 allows current to continue flowing when the switches have been turned off, so that inductance on the output does not cause voltage spikes which would damage the pass transistors or motor drivers. | 73 |
| 3-19 The Kobalt 24V 5 Ah battery. Housing removed on the right, showing the built-in BMS, output terminals, and 18650 cells. | 73 |
| 3-20 Robot power supply board, which generates an isolated 5V logic supply, distributes power to the legs and computer, and has pass transistors for turning on and off motor power and logic power. The underside of the power supply (right) has board mount FASTON terminals which mate with the spades built into the battery. | 74 |
| 3-21 Sequence of frames from the optimization on a 2D sagittal plane model of the robot. | 76 |
| 3-22 Time series of the quadruped doing a back-flip | 77 |
| 3-23 Joint torques (Left) and joint output power (Right) during the takeoff of the backflip. On top is the output from the 2D optimization, and on the bottom, the data collected from the actual robot during the flip. Values are for each pair of legs, so individual joint torques and powers are half those plotted. | 78 |
| 3-24 Joint position (Left) and joint velocity (Right) tracking during the takeoff of the backflip, for the front left leg. | 78 |

| | | |
|-----|---|----|
| 4-1 | Teleoperation and haptics setup consisting of two 3-DOF arms, with feet replaced by smooth spheres. | 82 |
| 4-2 | Little Hermes, a 6-DOF point-foot bipedal robot. Image used with permission | 86 |

List of Tables

| | | |
|-----|------------------------------------|----|
| 2.1 | Cheetah 3 Specifications | 25 |
| 2.2 | Actuator Specifications | 25 |
| 2.3 | CAN Data | 48 |

Chapter 1

Introduction

This thesis documents the development of a low cost, high performance, modular actuator, intended for use in legged robots and other machines which dynamically interact with the world, make and break contact and experience collisions.

The MIT Cheetah robot introduced a new actuation paradigm for legged robots, demonstrating that impressively fast, dynamic legged machines could be actuated with electric motors, given the right design approach. The actuators in the MIT Cheetah 2 and 3 robots use custom-designed electric motors optimized for torque density, and custom single-stage planetary geartrains [1]. These actuators, when driving, light, low inertia legs, allow for fast, open-loop proprioceptive control of ground reaction forces. While the performance of these actuators is exceptional, they are also expensive, as they were designed with little concern for cost, and every component is custom and manufactured in small quantity.

In addition to the cost of robots like the MIT Cheetah, their physical size and power makes experimentation with such robots inherently high-risk - particularly for the robot itself, but also for those operating it. This risk makes testing some controllers, for example, doing a backflip on a quadruped, a very daunting task. Having a cheaper, smaller, but still high-performance robot platform on which to test hopefully will allow for much faster controller development, and experimentation with "exciting" controllers or behaviors which may be hard to safely try out on a larger, more powerful machine. These concerns were discussed in the context of the Super

Mini Cheetah robot [2], which was built in a similar spirit to the quadruped discussed in this thesis.

To these ends, we have developed an inexpensive modular actuator intended to enable rapid development of dynamic robots and their control systems. We take the same approach to actuation as the MIT Cheetah robots: using high torque density motors and low-ratio, backdriveable transmission to allow high bandwidth torque control. Using these actuators, we have built a 12 degree-of-freedom quadruped robot, roughly 60% the size of the MIT Cheetah 3 robot, as well as several other robotic platforms. Although we have only recently started performing experiments on the quadruped, the initial results are extremely promising, including a fully 3D, 360° backflip from the ground - which we believe is a first for a quadruped robot. Hopefully the results of this thesis demonstrate that, with the right approach, remarkable robot performance can be achieved with relatively simple, inexpensive actuators, and will encourage others to employ similar actuation architectures.

1.1 A Brief Overview of Actuation for Dynamic Legged Robots

Legged robots present an interesting actuation challenge, because of the many important and often conflicting performance characteristics to consider.

Clearly specific torque (or force, for linear actuators) is important - without sufficient specific torque, a legged robot will not even be able to support its own weight. High actuator speed is necessary for fast locomotion, quick footstep placement for recovering from disturbances, and explosive maneuvers like jumping. Similarly, high acceleration capability is also critical, i.e. high torque to inertia. Without high torque to inertia, much the actuator's torque ends up accelerating the moving components of the actuators themselves, rather than accelerating the robot. Low reflected inertia is also an important quality for force control [1] and impact robustness, reducing the forces seen by the transmission and by the body of the robot during impact, [7]. Ro-

bustness to external impacts is critical: Another way of putting it, actuators must be tolerant to a velocity steps on their output, as zero-velocity collisions with the world can never be guaranteed. Many control approaches for legged machines assume reasonable torque/force control capability, and for fast behaviors, force control bandwidth needs to be relatively high. For example, running at 6 m s^{-1} , the ground contact times for the MIT Cheetah 2 are only 80 ms [8], so force profiles must be controllable during that tiny window. High efficiency is also desirable, although, given the current state of legged robots, perhaps less critical than most other performance criteria (up to a point). We can start worrying about efficiency when our robots stop falling over.

Hopefully this overview of some of the desirable characteristics for actuators in legged robots makes it clear that the design space is challenging, and there is not a clear "best" solution. It is often easy to satisfy several of these characteristics, but challenging to satisfy all of them simultaneously.

1.1.1 Proprioceptive Electric Actuators

The actuator discussed in detail in this thesis falls under the category of "proprioceptive" electric actuators. By designing to maximize electric motor torque density and minimize parasitic motor, transmission, and leg dynamics, simultaneously high torque density, high speed, robustness to external impacts, and high bandwidth torque control can be achieved [1]. This actuation strategy was first and most notably used in legged robots on the MIT Cheetah series of robots. In addition to the desireable performance characteristics of this type of actuator, they are also relatively simple compared to other acutation techniques, in terms of both mechanical design and control. Each actuator requires only one position sensor (at the rotor of the electric motor), no additional compliant elements built into the actuator, and relatively simple transmisisons. Torque control is achieved simply by current control of the electric motor. This actuation strategy has also demonstrated remarkable efficiency, with Cost of Transport (CoT) of the MIT Cheetah 2 being similar to that of a biological quadrupeds [8].

1.1.2 Hydraulic Actuators

The hydraulic legged robots from Boston Dynamics [32], starting with Big Dog, have set the standard for the performance capabilities of modern legged robots. Hydraulic actuators tend to have high force density and high robustness to impacts, as impact loads are distributed over the large surface area of the hydraulic channels, rather than, for example several small gear teeth. Another compelling reason to use hydraulics, especially for high degree-of-freedom machines, is the relative ease of adding high-force degrees of freedom. For an electric motor driven robots, each actuator needs to be sized for its peak performance, which makes building systems with many degrees of freedom needing high peak power and force at all the joints (and especially at distal joints) very challenging. With a hydraulic system, it is easier to build high-force distal links (ankles, wrists, fingers, etc) without adding significant mass and inertia to the limbs. For quadrupeds with point-feet (no actuated ankles), however, it is relatively easy to design high force, high power, low inertia limbs with all the actuation mass located at the body, making hydraulics less advantageous for this type of robot. Hydraulic actuators also tend to have poor efficiency, from viscous losses in the moving fluid, servovalve internal leakage and pressure drop[24].

1.1.3 Series Elastic Actuators

Series elastic actuators (SEA) take a conventional highly-gearred electric motor, and add a spring and spring displacement sensors at the output [5]. Joint torque can be measured at the output by measuring spring deflection, and equivalently, output torque can be controlled by controlling spring deflection. In addition to (and perhaps even more importantly than) allowing closed-loop output torque control, the spring serves to decouple the gearbox from impacts seen at the output of the actuator. This allows high gear ratios to be used without risk of damaging the gears upon impact. In some SEA implementations, the spring serves to store significant energy, increasing the peak power output of the actuator when the spring and electric motor do work in the same direction. Perhaps the best examples of SEA's use in legged robots are

quadrupeds StarlETH [3], and its successor, ANYMal [4], from the Robotic Systems Lab at ETH Zurich. While impressive work has been done on these robots in terms of control of quadruped locomotion, planning, and navigation, they simply do not have the actuation performance to execute the sorts of dynamic maneuvers like high-speed running which have been demonstrated on the MIT Cheetah, and the hydraulic robots from Boston Dynamics.

While their lack of actuation performance may be a result of the specific implementation, not an inherent limitation of SEAs, several other factors make SEAs less attractive for use in a low-cost but high performance actuator. Spring deflection control turns the torque control problem into a position control problem, so SEAs often use expensive, low-backlash, high accuracy transmissions like harmonic drives. And because of transmission backlash and compliance, many SEA implementations use three position sensors: One at the electric motor for commutation; one after the transmission, on the input of the spring; and one on the output of the spring. The addition of a spring and two additional position sensors substantially increases both the mechanical complexity and the cost of the actuator, and sacrifices weight and volume of the actuator to non-power and torque producing components.

1.2 Existing Modular Robot Actuators

Several modular actuators with built-in electronics and controls are already commercially available, however none of these actuators meet desired performance requirements, let alone the low cost. Two actuators of particular note, which are not conventional high gear ratio, un-backdriveable actuators, are the ANYDrive, and the actuators from HEBI Robotics.

The ANYDrive [4] shares a number of goals with the actuator presented in this thesis. As the ANYDrive was also intended for use in legged robots (ANYMal), it is designed for torque control and robustness to impacts. The ANYDrive is a series elastic actuator, consisting of a brushless motor, harmonic drive, and series spring used for torque measurement and control. The design goals of ANYDrive diverge from

ours in several aspects, however. It is capable greater static torque accuracy, thanks to its closed-loop torque control, but at the cost of torque bandwidth. Although it can achieve 70 Hz closed-loop *small signal* torque bandwidth, for large torque steps its bandwidth is greatly reduced, dropping to 24 Hz at 10 N m, and continues to fall roughly linearly with torque, due to the large motor displacements necessary to displace the output spring for high torques. With 40 N m peak torque and 1 kg mass, the actuator has similar torque density to the actuator in this thesis, however, it is much slower, with a maximum speed of 12 rad s⁻¹, and lower power density.

Another commercially available SEA-based actuator are the HEBI X5 and X8 series of actuators [30], originally from CMU. These are very similar in size to the actuator in this thesis, but they are too slow and power limited for very dynamic machines - the X5 actuators are only capable of 20 W peak output power. The most notable feature of the HEBI actuators is the software and API for controlling these actuators - it is well documented, easy to use, and feature rich, making it easy to get up-and-running with their actuators.

Chapter 2

Modular Actuator

A primary contribution of this thesis is the design of a cheap, easy to use, modular actuator suited for the demanding requirements of dynamic legged locomotion. Cost was a primary design concern: For the MIT Cheetah robots, the hardware cost is dominated by the electric motors and their transmissions. Many design choices of the actuator were made to reduce the cost of the unit. As a result, this design certainly is not the most performant design possible given its dimensions and weight. The actuator has a BOM of around \$300 in sub-50 quantity, representing between 1 and 2 orders of magnitude lower cost than the actuators used on the MIT Cheetah. In fact, the entire hardware cost of the quadruped discussed in Chapter 3 is less than the cost of a single actuator use in the MIT Cheetah. So far, 26 of these actuators have been used in the lab, for a 6-DOF bipedal robot, a haptic feedback and bilateral teleoperation system, and the quadruped previously mentioned.

This actuator builds off the actuation paradigm used in the MIT Cheetah series of robots, using a high torque density electric motor, coupled to a low gear ratio transmission to achieve high torque density, high backdriveability, and high bandwidth force control through proprioception [1]. While the MIT Cheetah uses custom-designed motors optimized for torque density, the design presented here leverages the proliferation of high performance brushless motors for RC drones and airplanes, which are manufactured overseas in huge quantities, at very low cost. These motors have been tightly integrated into an actuator which also includes a 6:1 single-stage

planetary gear reduction, motor controller with built-in position sensor and joint-level control capabilities, output which can handle substantial moment loads for directly attaching limbs to the actuators, and daisy-chainable power and communication to simplify wiring.



Figure 2-1: Assembled actuator.

2.1 Performance Requirements

One of the intended applications of the actuator was building a smaller-scale quadruped with similar dynamic capabilities to the MIT Cheetah 3 robot. While Cheetah 3 is a fantastic platform, because of its size and power, it requires several people to run experiments on safely. It is powerful enough to severely damage itself or hurt the operators should anything go catastrophically wrong, and is a very expensive machine. Having a smaller, safer, cheaper, but similarly high-performance platform to experiment on will hopefully allow for faster controller development, and testing higher-risk controllers without fear of damaging the expensive robot or risk of injury.

Rough actuation requirements for a small high-performance quadruped can be estimated by applying scaling laws to Cheetah 3. Relevant specifications of Cheetah 3 are summarized in Table 2.1

For a quadruped 60% the size of Cheetah 3, assuming linear scaling in every dimension, the mass of the robot will be 9.5 kg, and link length 0.2 m. To produce the

Table 2.1: Cheetah 3 Specifications

| | |
|------------------------|------------------------|
| Mass | 41 kg |
| Maximum Joint Torque | 230 N m |
| Maximum Joint Velocity | 18 rad s ⁻¹ |
| Link Length | 0.34 m |

same ground reaction force per mass, joint torque should be around 30 N m, scaling as $length^4$. To achieve similar locomotion speed, joint angular velocities should scale by $length^{-1}$. These numbers serve as rough guidelines for the sizing of the actuator, not firm design targets, as a variety of factors like electric motor availability influenced the size of the final actuator and the the quadruped built using them.

The specifications of the actuator developed are summarized in Table 2.2.

Table 2.2: Actuator Specifications

| | |
|---------------------------|--------------------------------|
| Mass | 480g |
| Dimensions | 96 mm O.D., 40 mm axial length |
| Maximum Torque | 17 N m |
| Continuous Torque | 6.9 N m |
| Maximum Output Speed | 40 rad/s@24 volts |
| Maximum Output Power | +250/ - 680 watts |
| Current Control Bandwidth | 4.5kHz@4.5N m, 1.5kHz@17N m |
| Output Inertia | 0.0023 kg m ² |

Several qualities typically sought after in actuator design which were *not* of great concern in this actuator design are positioning accuracy, backlash, and static torque accuracy. The actuator has roughly 0.005 rad (0.28°) of backlash at the output from its planetary geartrain, as it uses off-the-shelf gears and no mechanisms for compensating backlash. Similarly, because of the cogging torque at the motor (which has not yet been compensated for, although this is possible), the actuator has roughly

0.25 N m of position-dependent torque ripple at the output. For many tasks, and particularly legged locomotion, these effects are of little importance. With respect to positioning accuracy, foot positioning better than a centimeter or so is highly unlikely to be critical. In all likelihood, there will be far greater error in the rest of the robot’s state estimation, particularly for gaits or behaviors involving periods of flight, with no feet in contact with the ground. Similarly, static torque accuracy is of little importance, as over any appreciable rotation of the actuators, the power and average torque delivered is not affected by cogging torque ripple. On Cheetah 3, we have observed roughly 10% static torque error from gearbox friction and other effects, which we have not needed to consider for control, and does not limit the performance of the robot. Some of these effects however do negatively impact performance in other applications of these actuators, such as the haptic feedback interface discussed, where cogging torque noticeably affects the user-perceived forces.

2.2 Mechanical Design

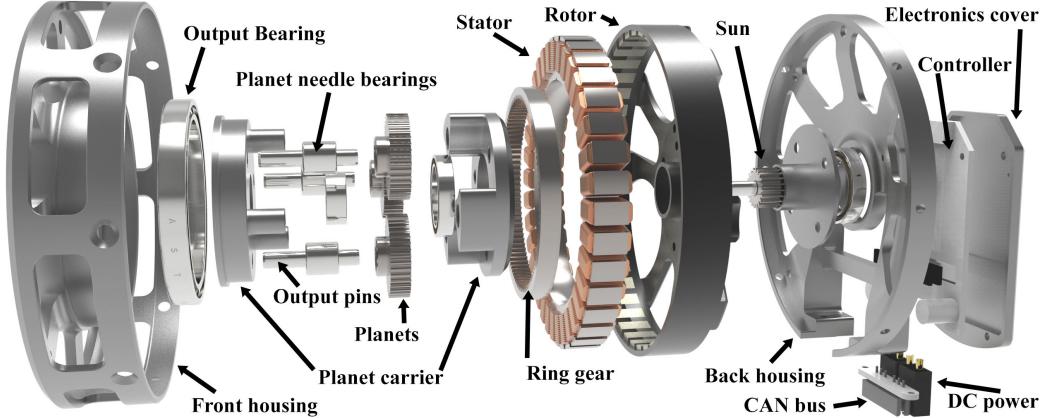


Figure 2-2: Exploded view of the actuator.

2.2.1 Electric Motor

The actuator uses the rotor and stator from an off-the-shelf brushless motor designed for large RC drones. The particular model used nearly identical in shape and

performance to the T-Motor U8, which was used in the direct-drive legged robot Minitaur[20], but available for between \$60 and \$90, less than 1/3 the cost of the U8. Peak torque and torque constant are identical to the U8, although continuous torque is slightly lower due to lower fill-factor windings, resulting in slightly higher winding resistance and higher power dissipation for a given torque. This particular motor was chosen for its geometry: A large airgap diameter of (81 mm), a stack length of 8.2 mm, and a large number of pole-pairs (21). These features give it particularly high torque density for an off-the-shelf motor. Within the operating range of the actuator, some torque saturation of the motor is observed, with the torque constant dropping by 12% at maximum torque, as shown in Figure 2-3.

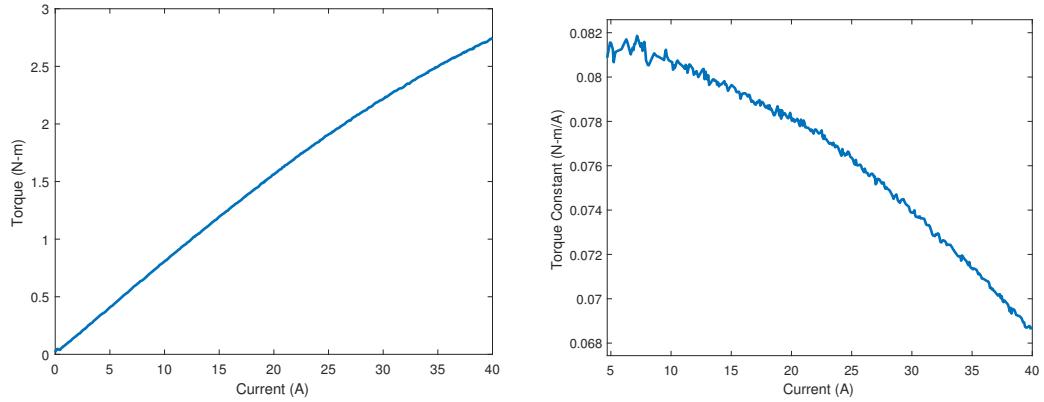


Figure 2-3: Torque vs current (Left) and torque constant vs current (Right). The motor exhibits some saturation, with torque constant dropping by 12% at maximum current.

The rotors have been post-machined to fit the sun gear for the planetary gearbox and a diametrically magnetized cylindrical magnet for position measurement. The large through-bore of the stator allows the entire planetary gearbox to fit within the center of the stator, making the final packaging of the actuator extremely compact. The rotor and stator together constitute about half the total mass of the actuator.

2.2.2 Housing and Planetary Gearbox

Between the rotor and output of the actuator is a 6:1 planetary gear reduction. To keep cost down, all the gears are stock parts. The 3 planets and sun gear are con-



Figure 2-4: Stator (left) and rotor (right) of the brushless motor used in the actuator. The rotor has been modified to fit the sun gear of the planetary transmission.

figurable parts from Misumi (part numbers GEFHB0.5-40-5-8-W3 and GEABN0.5-20-8-K-4 respectively), and the ring gear is made by KHK Gears (part number KHK SI0.5-100). The ring gear required post-machining to a 55 mm outer diameter. The ring gear is press-fit into the front half of the actuator housing, and the sun gear is press-fit to the rotor, with the interference fit assisted by Loctite 648 high-strength retaining compound. Each planet gear has a HK0408 needle roller bearing pressed into the bore. The needle bearings ride on precision 4 mm dowel pins hardened to 58 HRC. These pins are pressed through the front half of the planet carrier and extend past the output of the actuator, serving as both locating and torque-transferring features on the output.

A second revision of the housings was designed, to make the assembly process more tolerant of loose machining tolerances. In the first batch of housings we sent out for, fixturing errors on the part of the manufacturers lead to concentricity errors on several parts. In particular, on the front half of the housing, many parts showed concentricity errors between the output bearing bore and the ring gear bore. Similar errors were made on parts of the planet carrier, as well. When assembled, these would cause the planetary gearboxes to bind or not operate smoothly. For the initial round of actuators, this issue was avoided by measuring the runout of all the parts, binning the parts, and throwing out those which were severely out of tolerance. To avoid

having to do this, a second revision of the housings was designed which, wherever possible, moved concentric features to the same side of the parts, so that when the parts are machined on a 3-axis milling machine, these features are all produced in the same operation. This eliminated the possibility of setup or fixturing error causing concentricity errors, and allowed us to use very low-cost machining services to produce working parts, even with their relatively loose tolerance guarantees.

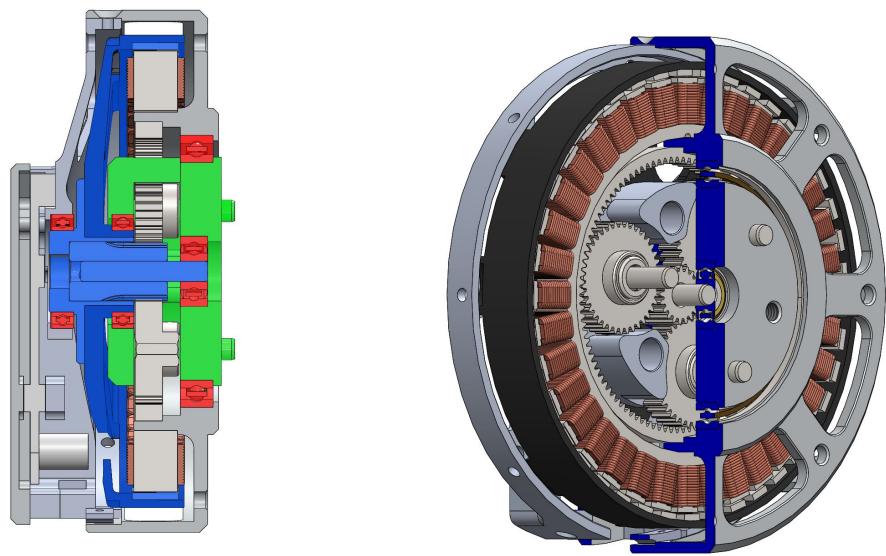


Figure 2-5: Cross-sectional views of the actuator. On the left, bearings are highlighted in red, rotor in blue, and planet carrier in green.



Figure 2-6: Machined components of the housing and planetary gearbox

2.2.3 Design for Impact: Estimating Transmission Loading During Collisions.

Robustness to impacts is a critical feature for actuators in legged robots. Non-zero impact velocity contacts are inevitable, even if the control strategy works hard to avoid them. Any uncertainty in the terrain, any error in robot state estimation, and any disturbance can cause impacts with significant velocity, and legged machines must be robust to this.

As previously discussed, strategies for mechanically mitigating these impacts exist, like adding intentional series elastic elements to either the legs or actuators. Alternatively, as with the MIT Cheetah robots, the inherent compliance in the robot's feet and leg links is sufficient to prevent the actuators from breaking. For the MIT Cheetah, the effects due to these compliances are sufficiently high-frequency to be neglected for locomotion control. To understand some of the limitations of actuators which must undergo impacts, some simple analysis is done here to estimate transmission loading during collisions, and aid in the design of both actuator transmissions and robot legs.

The basic model which will be examined is as follows: A motor with rotor inertia J_1 is connected through a geartrain or ratio n to a rigid link, of inertia J_2 . The link, moving at angular velocity ω collides with the world inelastically, coming to a complete and instantaneous stop. Somewhere in the system is a compliant element of stiffness K : between the motor and the input gear (Case 1), between the output gear and the link like most SEA's, (Case 2), or between the end of the link and the world, like a compliant foot (Case 3).

Peak force in the gearing will occur when all the kinetic energy in the rotor (or rotor and link, in Case 3), is transferred to potential energy stored in the displacement of the spring. At this point, the springs will have displaced by angle θ . If there is no damping in the system, so energy is conserved:

$$\text{Case 1: } \frac{1}{2}J_1\omega^2n^2 = \frac{1}{2}K\theta_1^2 \quad (2.1)$$

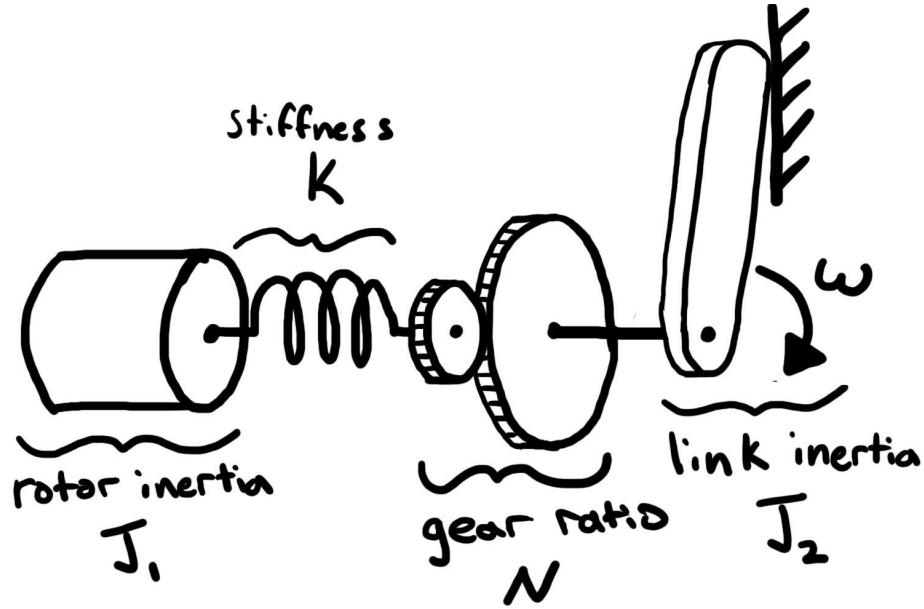


Figure 2-7: Case 1: Actuator input compliance

$$\text{Case 2: } \frac{1}{2}J_1\omega^2n^2 = \frac{1}{2}K\theta_2^2 \quad (2.2)$$

$$\text{Case 3: } \frac{1}{2}\omega^2(J_1n^2 + J_2) = \frac{1}{2}K\theta_3^2 \quad (2.3)$$

$$\theta_1 = \omega n \sqrt{\frac{J_1}{K}} \quad (2.4)$$

$$\theta_2 = \omega n \sqrt{\frac{J_1}{K}} \quad (2.5)$$

$$\theta_3 = \omega \sqrt{\frac{J_1n^2 + J_2}{K}} \quad (2.6)$$

Peak torque on the input gear is given by:

$$\tau_1 = K\theta_1 = \omega n \sqrt{J_1 K} \quad (2.7)$$

$$\tau_2 = \frac{K\theta_2}{n} = \omega \sqrt{J_1 K} \quad (2.8)$$

$$\tau_3 = \frac{K\theta_3}{n} \frac{J_1n^2}{J_1n^2 + J_2} = \omega J_1 n \sqrt{\frac{K}{J_1n^2 + J_2}} \quad (2.9)$$

We see already that impact forces in all cases scale with impact velocity ω , with

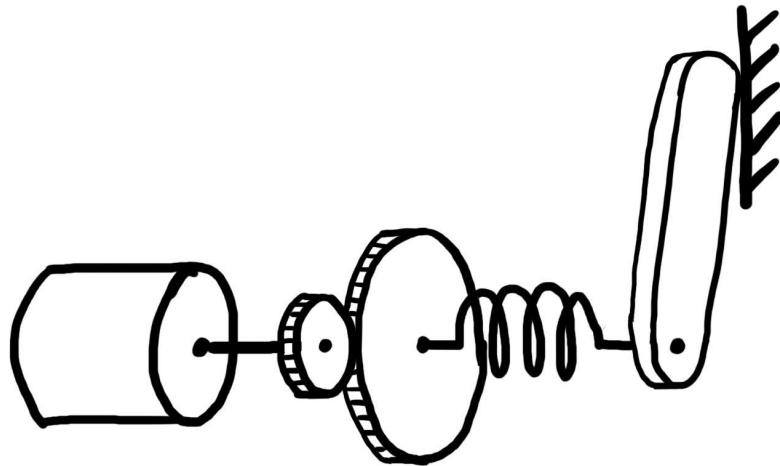


Figure 2-8: Case 2: Actuator output compliance

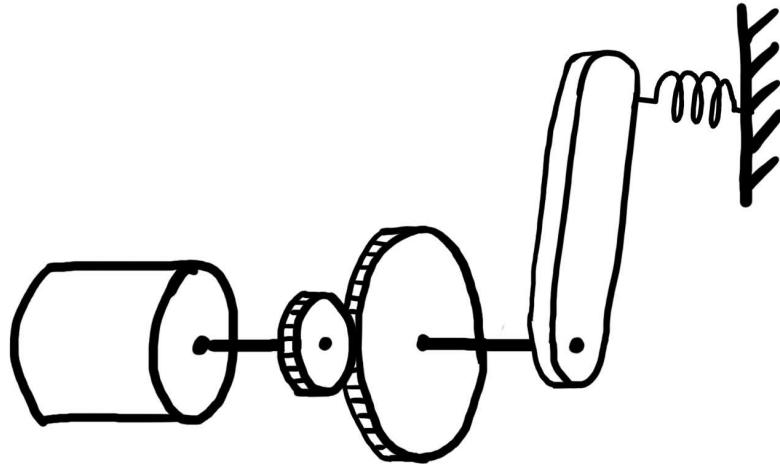


Figure 2-9: Case 3: End effector compliance

the square root of rotor inertia J_1 , and with the square root of stiffness K .

Open-loop force control bandwidth, for all these cases, will be limited by the natural frequency of the rotor and the series compliance, when the end of the link is fixed. Fixing the natural frequency to be ω_n , and solving for the required spring constants K_1 , K_2 , K_3 for Case 1, Case 2, Case 3 respectively:

$$\omega_n = \sqrt{\frac{K_1}{J_1}}, \quad K_1 = J_1 \omega_n^2 \quad (2.10)$$

$$\omega_n = \sqrt{\frac{K_2}{J_1 n^2}}, \quad K_2 = J_1 n^2 \omega_n^2 \quad (2.11)$$

$$\omega_n = \sqrt{\frac{K_3}{J_1 n^2 + J_2}}, K_3 = \omega_n^2 (J_1 n^2 + J_2) \quad (2.12)$$

Plugging these K s into the peak torque equations:

$$\tau_1 = \omega n J_1 \omega_n \quad (2.13)$$

$$\tau_2 = \omega n J_1 \omega_n \quad (2.14)$$

$$\tau_3 = \omega n J_1 \omega_n \quad (2.15)$$

The peak torques are all identical, if the stiffness are chosen such that the natural frequencies (and thus the open-loop torque bandwidth) are equal. This may be an intuitive result, but it is a useful one. With some idea of the impact velocities the robot should be able to withstand and necessary force control bandwidth, the robot legs can be designed, and actuator components sized, so that the actuator can survive these impacts. By intentionally including compliance within the actuator (i.e. and SEA), the actuator can be made "perfectly" passively impact robust, if the components are sized and stiffness chosen such that the maximum tolerable impact force is at a higher velocity than the maximum output velocity of the actuator. Even in that case, however, there will be some impact velocity which the actuator will not be able to withstand, which could be caused by something external to the robot.

By inserting the maximum allowable torque for the limiting component of the actuator into 2.13 (in this case, the sun gear in the planetary gearbox), and deciding a maximum allowable collision speed, we can determine how stiff the leg of the robot must be to satisfy the maximum torque, and what open-loop bandwidth is possible. For this actuator, the allowable torque on the sun gear is 11 N m. With a gear ratio of $n = 6$ and rotor inertia $J_1 = 0.000\,072 \text{ kg m}^2$, and a maximum actuator velocity of 40 rad s^{-1} , the maximum allowable natural frequency for a full-speed collision is 101 Hz. Although that torque bandwidth is actually quite reasonable, and certainly sufficient for legged locomotion, even higher bandwidth can be achieved if the maximum collision velocity is limited: in all likelihood, a robot using these actuators will not experience a maximum-speed collision with a rigid object. These numbers give

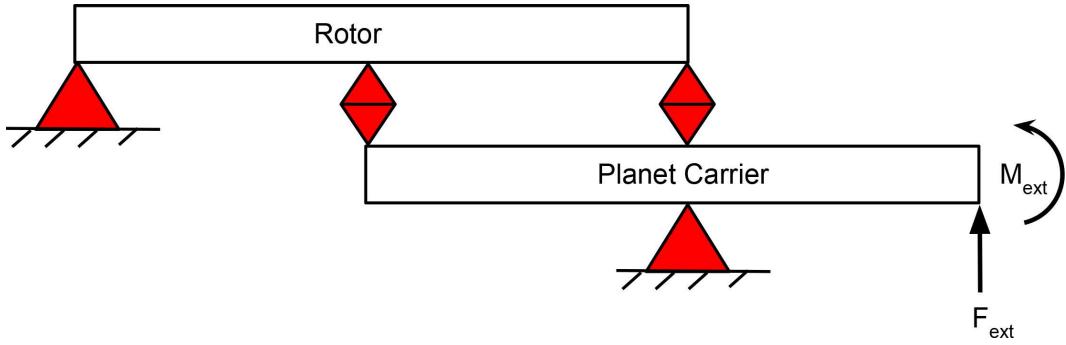
reasonable confidence that the actuator will handle impacts well without any compliance built into the actuator, as the stiffness of the links, other transmission elements, and foot of the robot will likely bring the natural frequency below these values.

Worth pointing out, the peak force tolerable by the transmission is significantly greater than the peak force which the motor can produce - by roughly a factor of 4. This means that for hard impacts, using the motor to actively "run away" from impacts to reduce the load on the transmission would have little effect, only reducing impact forces by around $\frac{1}{4}$. If the transmission were sized only to match the capabilities of the motor, then the achievable bandwidth for a given maximum impact velocity would be *far* lower. This is the operating regime most SEAs operate in.

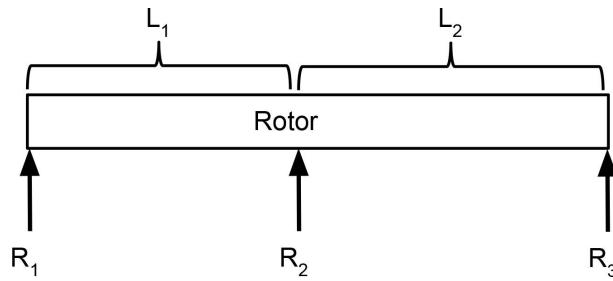
2.2.4 Bearing Loads

The actuator was designed so that the output of the module could directly support relatively large moment loads, like a leg link attached to the output. Since multiple of these actuators will be used in series, a good target for moment support is at least the output torque of the actuator. Placing a single crossed roller bearing at the output was a very attractive option from a design standpoint, as they can support large radial, axial, and moment loads by themselves. However a suitable crossed roller bearing would have by itself increased the parts cost of the actuator by over 50%. Instead, low cost thin-section metric bearings have been used throughout the actuator. As a result, each bearing used in the actuator costs only a few dollars, as opposed to over \$100.

To estimate the bearing loads, and the allowable moment on the output of the actuator, we assume that individual bearings have no moment stiffness, so all moment loads on the output are reacted by radial loads on the bearings. This should be a reasonable assumption for small angular deflections of the bearing inner races, as under no moment, the contact angle in the bearing is 0° , giving them no moment stiffness. Under this assumption, solving the bearing loads becomes a simple statics problem, and the actuator can be simplified to the following schematic, with bearings in red:



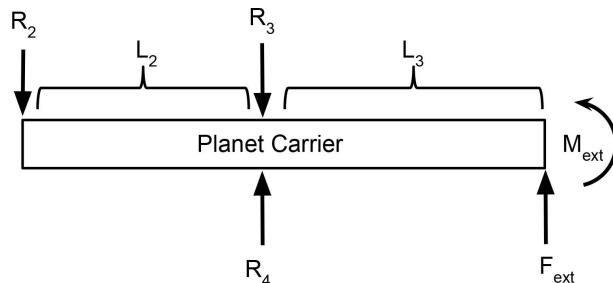
First, looking at the force and moment balance on the rotor:



$$\sum F_z = 0 = R_1 + R_2 + R_3 \quad (2.16)$$

$$\sum M = 0 = R_3 L_2 - R_1 L_1 \quad (2.17)$$

And for the planet carrier:



$$\sum F_z = 0 = F_{ext} - R_2 - R_3 + R_4 \quad (2.18)$$

$$\sum M = 0 = M_{ext} + F_{ext} L_3 + R_2 L_2 \quad (2.19)$$

The bearing loads $R_1 - R_4$ can be solved for by placing the equations above in

the form $0 = Ax + B$, where B comes from the external loads and A comes from geometry, and solving for x , where x is the column vector of $R_1 - R_4$.

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ -l_1 & 0 & l_2 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & l_2 & 0 & 0 \end{bmatrix} \quad (2.20)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ F_{ext} \\ F_{ext}l_3 + M_{ext} \end{bmatrix} \quad (2.21)$$

For the actuator, $l_1 = 11$ mm and $l_2 = 13.5$ mm. The maximum dynamic loads of the bearings, R_{max} , in Newtons, are:

$$|R_{max}| = [R_1 \ R_2 \ R_3 \ R_4]^T = [940 \ 940 \ 1100 \ 2500]^T$$

Where the bearings are 6702, 6702, 624H, and 6708 series, respectively. For a pure moment load of 17 N-m, the maximum output torque of the actuator, this puts bearing loads, in Newtons, at:

$$|R| = [690 \ 1260 \ 565 \ 694]^T$$

Clearly bearing 2, between the rotor and the planet carrier, is the moment-limiting component in this design. With 17 N m of bending moment on the output, the bearing is loaded 30% over its dynamic load rating. Increasing the bearing by one size, to a 6802 series bearing with the same 15 mm I.D. doubles the dynamic load rating of the bearing. This would put all the bearings within their load ratings for this bending moment on the output of the actuator. This change will be made in future revisions of this design. So far, this failure mode has not been observed, likely since peak torques are rarely necessary and are never sustained for more than a fraction of a second.

For another typical configuration in a robot, such as the leg described in Chapter 3 producing 150 Newtons of vertical force (1.5 bodyweights of the quadruped described), a force of 150 N acts at a distance $l_3 = 50$ mm from the output of the actuator, and the resulting loads are:

$$|R| = \begin{bmatrix} 306 & 556 & 250 & 450 \end{bmatrix}^T$$

Which are well within safe bounds for all the bearings.

2.3 Motor Control Hardware

A single PCB contains the hardware for motor control, position sensing, and communication (Figure 4). This keeps the number of wires needed to assemble a robot around these actuator to a minimum, and keeps the electrically noisy inverter and motor phase leads enclosed and away from other electronics. The power electronics stage of the motor driver is designed for 24V nominal input voltage and 40 amp peak phase current, although thermal limitations of the motor limit continuous operating current to below that value. Components that see the input voltage are all rated to at least 40V, the limiting components being the MOSFETS, so the actuator has headroom to tolerate unexpected spikes in input voltage. The controller receives desired torque, position, velocity, and position and velocity gains over CAN bus at rates of up to $\frac{4\text{kHz}}{\# \text{ of actuators}}$, and responds with measured position, speed, and estimated torque based on current measurements. For many-DOF systems needing high communication bandwidth, multiple CAN networks can be used to keep communication bandwidth high - in the quadruped described in Chapter 3, 4 CAN networks were used; one per 3-DOF leg.

2.3.1 3-Phase Inverter

At the heart of any 3-phase motor controller is a 3-phase inverter. The inverter consists of three transistor half-bridges, which apply voltage to the three motor terminals.

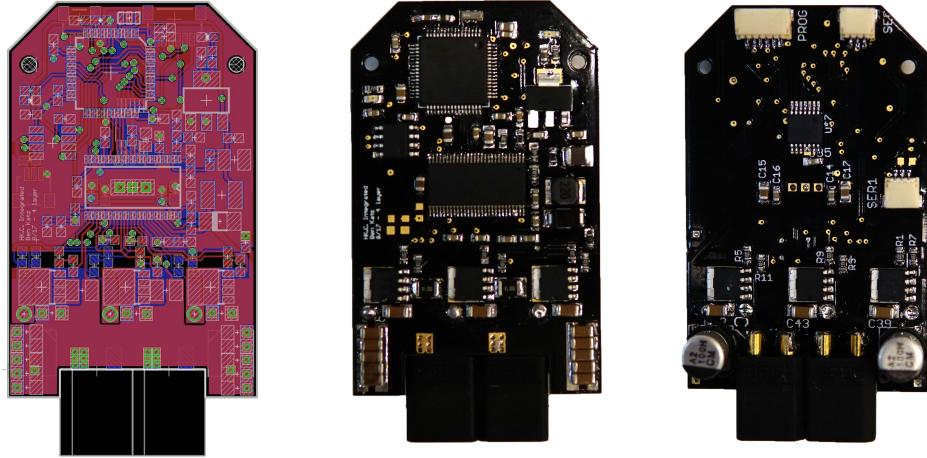


Figure 2-10: PCB layout, front, and back of the inverter.

The MOSFET inverter described here was designed to handle slightly higher current levels to the electric motor used, so that the motor would be the limiting factor for performance, rather than the electronics.

Because the motor used is very low inductance, around $30\ \mu\text{H}$, a fairly high switching frequency and control loop frequency of $40\ \text{kHz}$ was used, to allow fast closed-loop current control, and minimal current ripple caused by PWM.

For each half-bridge, the power dissipation at a constant torque can be estimated as a combination of conduction and switching losses:

$$P = \left(\frac{I_{peak}}{\sqrt{2}} \right)^2 R_{DS(on)} + \frac{2}{\pi} V I_{peak} F_{switch}(t_{rise} + t_{fall}) \quad (2.22)$$

Where I_{peak} is the peak phase current for the given torque, V is the DC bus voltage, $R_{DS(on)}$ is the on-state resistance of the MOSFET, t_{rise} is the rise-time of the half bridge output voltage, t_{fall} is fall time, and F_{switch} is the switching frequency. The factors of $\frac{1}{\sqrt{2}}$ and $\frac{2}{\pi}$ come from the fact that a constant torque corresponds to sinusoidal phase currents.

The FETs are top-side heatsinked to the aluminum housing of the actuator. While for many MOSFET packages, top-sinking performance is poor due to the high thermal resistance between the die and the top of the case, since the transistor package is

made of plastic, the package used (LFPAK) has reasonable thermal performance in this configuration [21]. A 0.1 mm layer of $6 \text{ W m}^{-1} \text{ K}$ thermal pad separates the top of each FET from the aluminum housing, to take up slight variation in the heights of the transistors, and improve thermal contact between the transistors and the housing.

Given $R_{DS(on)}$ versus temperature estimated from the datasheet of the MOSFET used, and t_{rise} of 42 ns, and t_{fall} of 28 ns also estimated from datasheet values, and a thermal resistance from MOSFET junction to ambient of 25 K W^{-1} estimated from [21], and an ambient temperature of 25°C , the predicted steady-state MOSFET junction temperature, T_j , vs I_{peak} for the inverter is shown in Figure 2-11

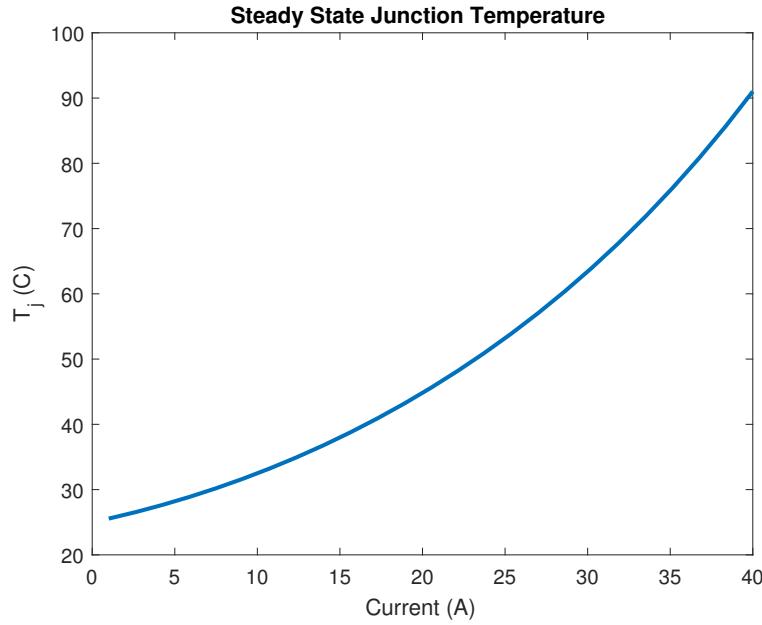


Figure 2-11: Estimated steady-state MOSFET temperature vs peak phase current.

Given these estimates, the inverter should be capable of substantially more steady-state current than the windings of the electric motor used, and this is validated by the thermal testing discussed at the end of the chapter.

2.3.2 Logic Power, Gate Drive, Current Sensing, and Microcontroller

Motor control, sensor measurement, and communication are handled by an STM32F446 microcontroller. The microcontroller runs a field oriented control algorithm to control motor torque at a loop rate of 40 kHz, synchronous with the FET switching. A PD position loop is run simultaneously, if desired. Current through two of the three phases are measured using $1\text{ m}\Omega$ low-side current shunts. A TI DRV8302 IC provides 3-phase gate drive, amplifier for the current shunts, and buck converter controller for powering the logic. The amplifiers on the DRV8302 are configured for a gain of 40, giving a current measurement range of slightly over 40 A

TI has recently released a new IC with the same functionality, but smaller footprint and lower cost, the DRV8323, which will likely be used in future revisions of the inverter.

2.3.3 Position Sensing

The motor controller has a hall-effect digital encoder directly integrated onto the PCB. The particular sensor used is the AS5047P, made by AMS. Nominally this is a 14-bit position sensor, although in reality individual samples have roughly 2 bits of noise on them. Every loop cycle, the microcontroller reads the output of the position sensor over SPI. This position sensor is used for both the electrical angle of the rotor, for commutation, and for output position measurement. By keeping track of when the output of the position sensor rolls over each rotation, mechanical position over multiple rotations of the rotor can be tracked. Since the position sensor is absolute at the rotor, output position is known at turn-on to within one rotation of the rotor, or 60° . This allows the actuator to be very roughly visually zeroed at turn-on. Motor velocity is estimated with a 16-sample moving average of the finite difference between position samples, equivalently

$$\dot{\theta} = 16\Delta t(\theta_k - \theta_{k-16})$$

Links to the PCB design files in EAGLE, bill of materials, and firmware can be found in the Appendix A.

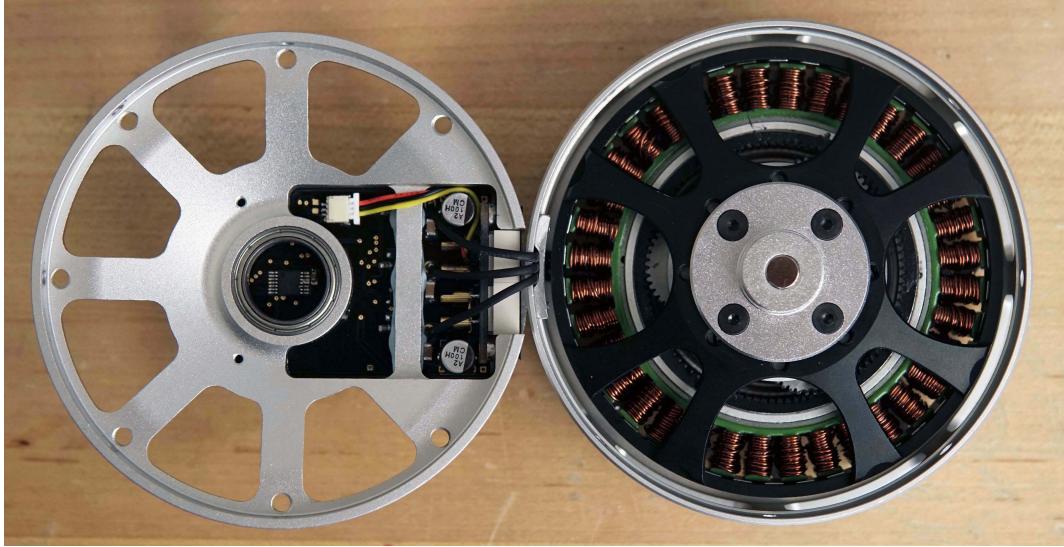


Figure 2-12: View of the actuator with the housing open. The encoder IC which measures rotor position can be seen at the center of the bearing on the left half of the housing. The diametrically magnetized magnet it uses is at the center of the rotor on the right.

2.4 Control

2.4.1 Current and Torque Control

Motor current control is implemented with field oriented control (FOC). See [12] for a detailed discussion of FOC. In brief, using FOC, 3-phase stator currents are transformed to a coordinate system rotating with the rotor by the DQ0 transform. The motor's behavior is roughly linear in the rotor reference frame, so high bandwidth current control can be easily achieved using discrete-time linear control techniques.

Voltages in the rotor frame are then transformed back to the stationary stator reference frame by the inverse DQ0 transform, and applied to the motor terminals by the inverter. The hardware was able to accomplish current (and thus electromagnetic torque) control at over 4.5 kHz closed-loop bandwidth, thanks to high sample rate (40 kHz), and fast electrical dynamics of the motor Figure 2-14.



Figure 2-13: View of the connectors to the actuator. Two XT-30 connectors are used for DC input power, and two 3-Pin Molex SPOX connectors are used for CAN. Each pair of connectors is in parallel, so actuator modules can be easily daisy-chained.

To reach closed-loop bandwidth of faster than 1/10th the sample frequency, an automatically tuned PI controller was used, which sets the gains of the current controller based on electrical parameters measured by the motor controller itself. The motor electrical dynamics (for one pole-pair) are described by the following voltage equations in the D/Q frame [13]:

$$V_q = RI_q + L_q \frac{di_q}{dt} + \omega \lambda_r + \omega L_d i_d \quad (2.23)$$

$$V_d = RI_d + L_d \frac{di_d}{dt} - \omega L_q i_q \quad (2.24)$$

Where R is phase resistance, L_d is direct axis inductance, L_q is quadrature axis inductance, λ_r is rotor flux linkage, ω is electrical angular velocity, and i_d and i_q are direct and quadrature axis currents. Because ω is governed by the motor's mechanical dynamics, it will generally change much more slowly than currents i_d and i_q . So assuming sufficient current control bandwidth can be achieved, the "Back-EMF" term, $\omega \lambda_r$, can be neglected for controller design, since it only affects the frequency

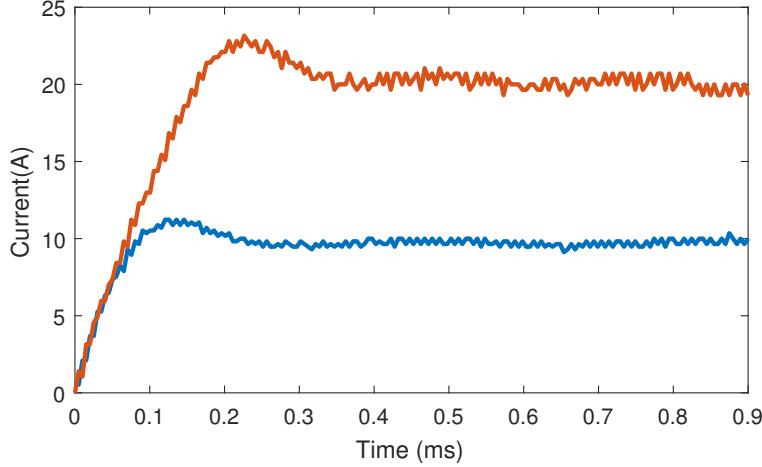


Figure 2-14: Phase current step response for 10 and 20 Amp steps, measured by external current probe. Current rise time is $75 \mu\text{s}$ for 10A, and $110 \mu\text{s}$ for 20A due to motor inductance and voltage limitation.

response well below the loop crossover frequency. The remaining dynamics are essentially two coupled first-order systems. The coupling has been dealt with here in a typical fashion [14] through feedforward terms in the controllers, although because of the low inductance of the particular motor used, the coupling terms tend to be small. Two independent loops control i_d and i_q .

For surface permanent magnet motors, like the one used, L_d and L_q are nearly identical, so the same sets of gains can be used for both controllers.

To measure R and L to get the discrete time dynamics of the motor and design the current controller, the inverter applies a small voltage step to V_d , and the resulting current step is recorded. Inductance is calculated from the initial slope of the current step response, and resistance from the steady state current.

The current controller is a discrete-time PI controller of the form

$$\frac{U(z)}{E(z)} = k \left(1 + \frac{ki}{z-1} \right)$$

, where U is control effort, and E is error. This has a pole at $z = 1$, a zero at $z = 1 - ki$, and a high-frequency gain of k . In this form, changing k only changes the loop gain, and changing ki only changes the zero location.

The zero of the integrator is placed near the pole of the RL circuit, so the forward

path looks like an integrator, using the following equation:

$$ki = 1 - e^{\frac{-RT_s}{L}} \quad (2.25)$$

The loop gain can then be approximately calculated for the desired crossover frequency, by the following equation. See Appendix A for derivation of these equations.

$$k = \frac{R\omega_c}{ki} = \frac{R\omega_c}{1 - e^{\frac{-RT_s}{L}}} \quad (2.26)$$

Crossover frequency is a user-configurable parameter on each motor controller. Very high current bandwidths result in slight current ripple which produces an audible "hiss", caused by a combination of noise on the current sensors and limited PWM resolution. For most the experiments conducted using the actuator, current bandwidth was set to 1 kHz.

Worth noting, the coupling term in V_q equation (Eqn.2.23) is what allow for "field weakening", or operating the motor above its base speed. By applying a negative current to the D-axis, a negative voltage proportional to ω , L_d , and i_d shows up on the Q-axis. This voltage has the effect of "cancelling out" some of the Back-EMF voltage, i.e. $\omega\lambda_r$. This effectively reduces the torque constant of the motor, allowing it to reach higher operating speeds and a wider power band for a given maximum DC-link voltage, at the cost of producing less torque per total current ($\sqrt{i_q^2 + i_d^2}$). Surface permanent magnet motors like hobby brushless motors inherently have low inductance, so their field weakening capability tends to be poor. L_d is small, so it takes a large current i_d for $L_d i_d$ to be comparable in magnitude to λ_r . That being said, correct use of field weakening, even on a surface PM motor, strictly improves the maximum performance of the motor. The power and torque plots in Figure 2-15 show the capabilities of the actuator used without any field weakening, with a maximum of field weakening current 10 A on the D-axis. For the motor used, an additional 20% speed and 7% higher peak power can be achieved.



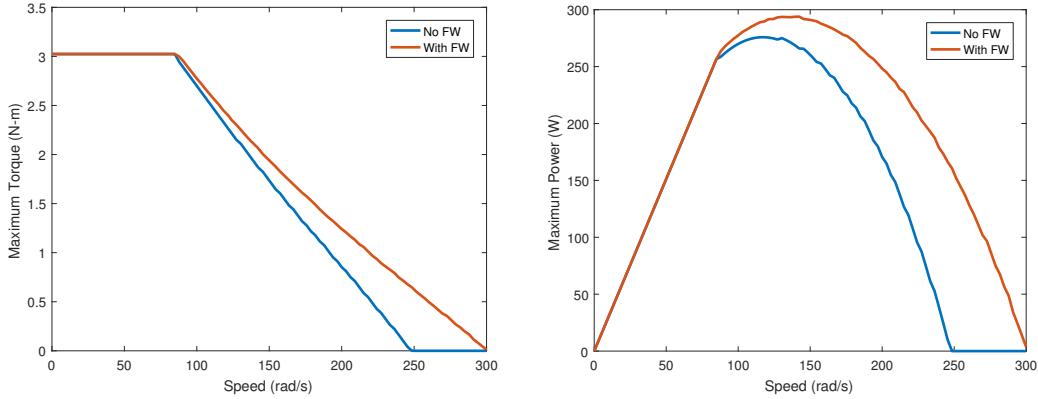


Figure 2-15: Maximum torque vs speed (Left) and power vs speed (Right), with and without field weakening.

2.4.2 Position Sensor Calibration and Linearization

An encoder calibration and linearization procedure was developed to automatically handle several issues which arise during the assembly of the actuator:

- It is not possible to accurately orient the diametrically magnetized position sensor magnet with respect to the magnets on the rotor. Ideally, the zero position of the encoder would align with the d-axis of the rotor.
- The ordering of the 3 motor phases is not guaranteed to be identical between stators, although we have observed them to be consistent. The phase ordering should be such that a positive q axis current results in a torque which produces a positive rotation on the encoder.
- The encoder IC is located only by the solder pads on the PCB. As a result, the encoder can be placed up to several hundred microns off the axis of rotation of the rotor. This introduces nonlinearity in the output of the encoder. Any mechanical angle error is amplified 21 times (the number of pole-pairs of the motor) for electrical angle, which can have significant effects for motor commutation.

The calibration procedure takes the following steps:

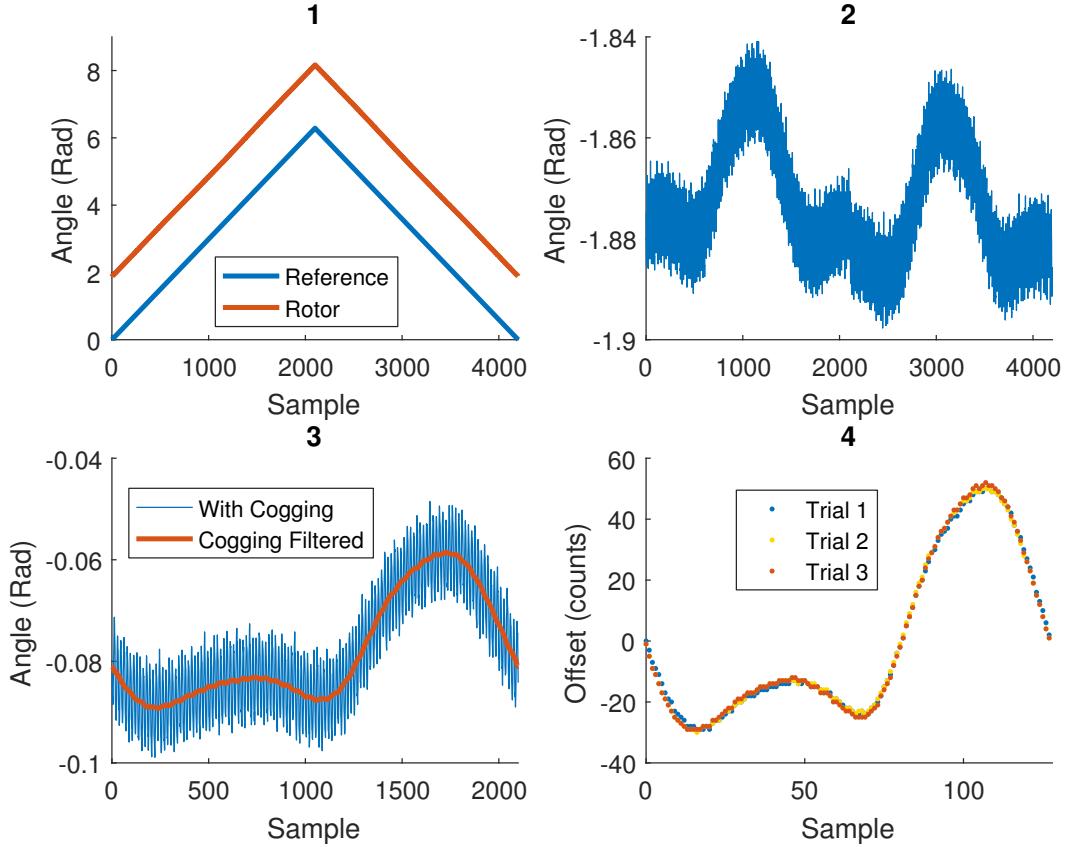


Figure 2-16: 1: Rotor angle and voltage vector angle during calibration. 2: Rotor angle minus voltage vector angle. 3: Averaged forwards and backwards error, with and without cogging torque FIR filter. 4: Calibration lookup table

1. To determine the ordering of the motor phases, the stator voltage vector is slowly rotated in the positive direction. If the output of the position sensor increases, the phase ordering is correct. If the output of the position sensor decreases, then two motor phases are swapped in firmware.
2. To determine position sensor offset, the stator voltage vector is slowly swept through a full mechanical rotation of the motor, both forwards and backwards, and the resulting rotor position is recorded. See Figure 2-16-1. During this sweep, the rotor d-axis closely tracks the stator current vector, which is the lowest-energy position. Because of cogging torque and friction, there is some tracking error between the angle of the current vector and the angle of the rotor. The DC offset of the position-sensing magnet is the average angle difference

between the stator voltage vector and the rotor angle.

3. Nonlinearities are measured by taking the difference between the series of rotor angles and voltage angles, Figure 2-16-2, subtracting off the DC offset, and averaging the forward and backwards rotation directions to eliminate friction effects 2-16-3. Several spacial frequencies of error can be observed in the difference: relatively high-frequency, from the motor's cogging torque, and low-frequency, caused by position sensor nonlinearity.
4. To compensate position sensor nonlinearity, first the cogging torque effects must be filtered out. This was accomplished by using a moving average (i.e. equal weighting) FIR filter with a window causing it to have zero gain at cogging frequency. Since the motor is rotationally symmetric, the cogging frequency must be periodic within an electrical cycle of the motor. If the FIR filter has a window width equal to one electrical cycle, then it will have zero gain at electrical frequency and all integer multiples of electrical frequency, so cogging effects should be perfectly filtered out. Applying this filter circularly over the signal produces an error signal with only position sensor nonlinearity remaining, which can be used as a lookup table to linearize the output of the position sensor, as shown in Fig. 2-16-4.

This technique proved effective even with intentional position sensor placement error of 0.75mm off the axis of rotation. Without correction, this caused such severe electrical position error that the sign of the motor torque actually reversed. After calibration, the torque ripple was not measurable beneath the cogging torque. Run multiple times, measured offset was found to be repeatable to within .0005 radians at the rotor.

2.4.3 Actuator configuration and communication

Commands and sensor information are sent between each actuator and a higher level controller (which could be a computer, another microcontroller, FPGA, etc.) over

CAN bus. CAN was chosen for its electrical robustness, daisy-chaining capability, and ease of implantation on embedded devices like the microcontroller on each inverter. Although CAN is not a particularly high-speed interface, with a maximum data rate of 1 MBps, the protocol has many built-in features which make networking multiple actuators easy and reliable.

Each motor receives 5 commands: position setpoint; velocity setpoint; position gain; velocity gain; and torque, and replies with 3 measurements: angle; angular velocity; and estimated torque. Estimated torque is calculated as $Kt \cdot i_q$, and may differ from the torque command if the motor is spinning at high speed and becomes voltage-limited. The built-in PD positioning functionality allows the PD loop to be closed at much higher bandwidth than would be capable by the high-level controller connected to all the motors, as it is run synchronously with the current control loop at 40 kHz. This feature may be useful for tasks like high-speed leg trajectory tracking during the swing phase for a legged robot. The commands and measurements are scaled so they can be efficiently packed into CAN frames (Table 2.3), with 8 bytes of commands and 6 bytes of measurements. With the additional 44 bits of CAN overhead per CAN Standard frame, this means a total 200 bits per set of commands and measurements per actuator on the CAN Bus. With a bus speed of 1 Mbps, this theoretically allows a communication rate of $\frac{5\text{kHz}}{\# \text{ of actuators}}$, although in reality achieving 100% bus utilization is challenging. In practice, though, we have run 3-DOF networks at 1.2 kHz without trouble.

Table 2.3: CAN Data

| Command | | Measurement | |
|-------------------|-------------------|------------------|-------------------|
| Parameter | Resolution (Bits) | Parameter | Resolution (Bits) |
| Position Setpoint | 16 | Position | 16 |
| Velocity Setpoint | 12 | Velocity | 12 |
| Position Gain | 12 | Estimated Torque | 12 |
| Velocity Gain | 12 | CAN ID | 8 |
| Torque | 12 | | |

The controllers have a serial console interface through which they can be configured and tested. Through this interface, parameters like the CAN ID the controller receives commands to, the CAN ID the controller sends commands to, the current control bandwidth, torque limit, and CAN Timeout period. This last feature is a safety measure so that if no new data is received over the CAN bus for a user-configurable period of time, the current setpoints are zeroed so the motor stops producing torque. The serial interface is also used for running the encoder calibration procedure previously described, and setting the origin of the actuator output.

2.5 Actuator Testing and Characterization

Most commercial robot actuators and motors have frustratingly little performance data provided with them. While sometimes peak power/torque vs speed plots are available, often not even that information provided. Finding detailed efficiency and power input/output data over the entire operating space of the actuator is more or less impossible. Similarly, actuators including transmissions rarely discuss the friction and losses associated with them. Without good actuator characterization, robot designers are left to do their own modeling to predict actuator and robot performance. By providing detailed actuator characteristics here, hopefully robot designers using these actuators will have more confidence in their robot designs and simulations.

In addition to the plots shown here, raw characterization data, as well as scripts for analyzing time-series along actuator trajectories are provided in Appendix A.

2.5.1 Steady-State Performance

Steady-state performance over most of the operating space of the actuator module was characterized on a custom 4-quadrant motor dynamometer (Figure 2-17) capable of measuring shaft torque, shaft speed, and input voltage and current to the actuator. The motor and controller without 6:1 planetary gearbox was tested over its full operating range, however the torque limit of the dynamometer (11 N-m) limited the testing of the entire actuator with gear reduction. Measurements were made with

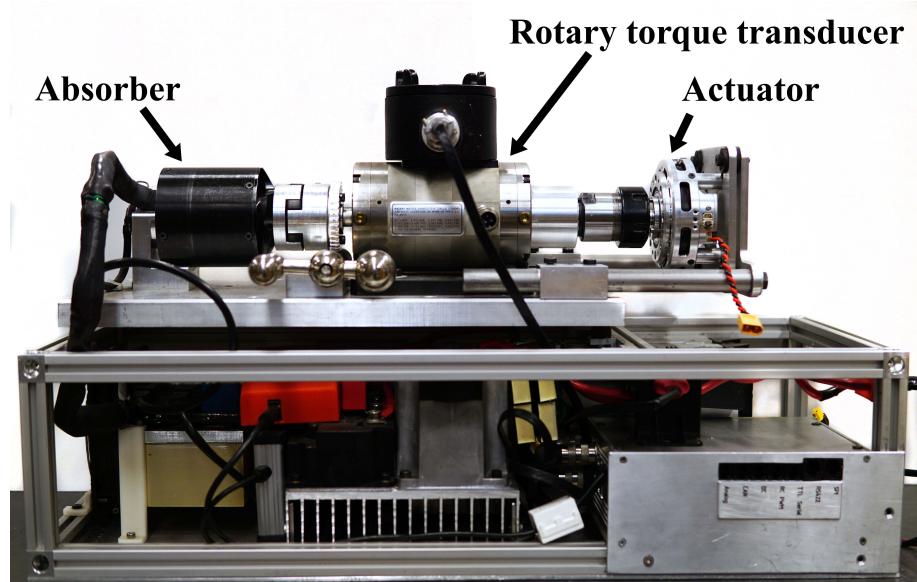


Figure 2-17: Custom 4-quadrant motor dynamometer used to measure steady-state performance and torque accuracy.

22V DC input to the actuator. Detailed steady-state performance of the motor is described by the following performance maps.

The efficiency map (2-18) shows the efficiency of converting electrical power to mechanical power, i.e. $\frac{\text{torque} \cdot \text{speed}}{\text{voltage} \cdot \text{current}}$ at every 1st quadrant (positive speed, positive torque) operating point.

The loss map (2-19) indicates at each torque and speed operating point, how much power is dissipated in the motor and control electronics, i.e. $\text{torque} \cdot \text{speed} - \text{voltage} \cdot \text{current}$. This can be used to ensure the desired operating trajectories of the actuator during use do not exceed the thermal limits of the actuator.

The input power map (2-20) indicates how much electrical power, either positive or negative, is required by the actuator at each operating point, i.e. $\text{voltage} \cdot \text{current}$. The energy consumption of a particular trajectory can be determined by integrating the input power map over the operating points in the trajectory. This data can be used to inform battery sizing, robot range, and other machine parameters.

Because of motor dynamics, the torque/speed operating areas are different for positive and negative work. Doing positive work, the actuator can reach full torque

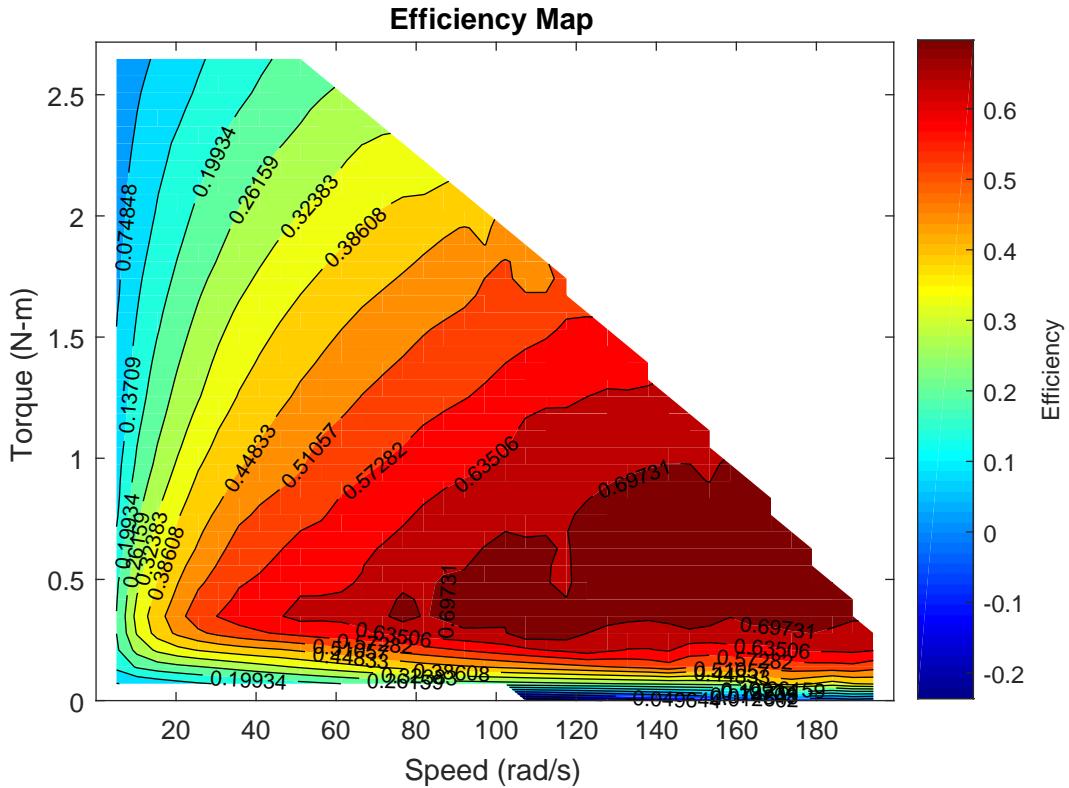


Figure 2-18: Efficiency Map

up to 60 rad/s at the motor (10 at the output), after which torque capability decreases due to voltage limitation. While doing negative work, full torque can be reached over the entire speed range. Maximum positive-work power output is 250 W, while maximum negative work output power is -680 W.

2.5.2 Torque Accuracy

The actuator has no output torque sensor, and performs no closed-loop output torque control. As a result, rotor inertia, gear and bearing friction, and cogging torque introduce error in the output torque. Although high torque accuracy was not a design goal for the actuator, torque accuracy has been characterized for completeness.

Gear and bearing friction was observed to be dependent on torque and rotation direction, with negligible speed dependence within the operating range of the actuator. The actuator has 0.09 N m static friction, and 0.04 N m/N m of torque-dependent

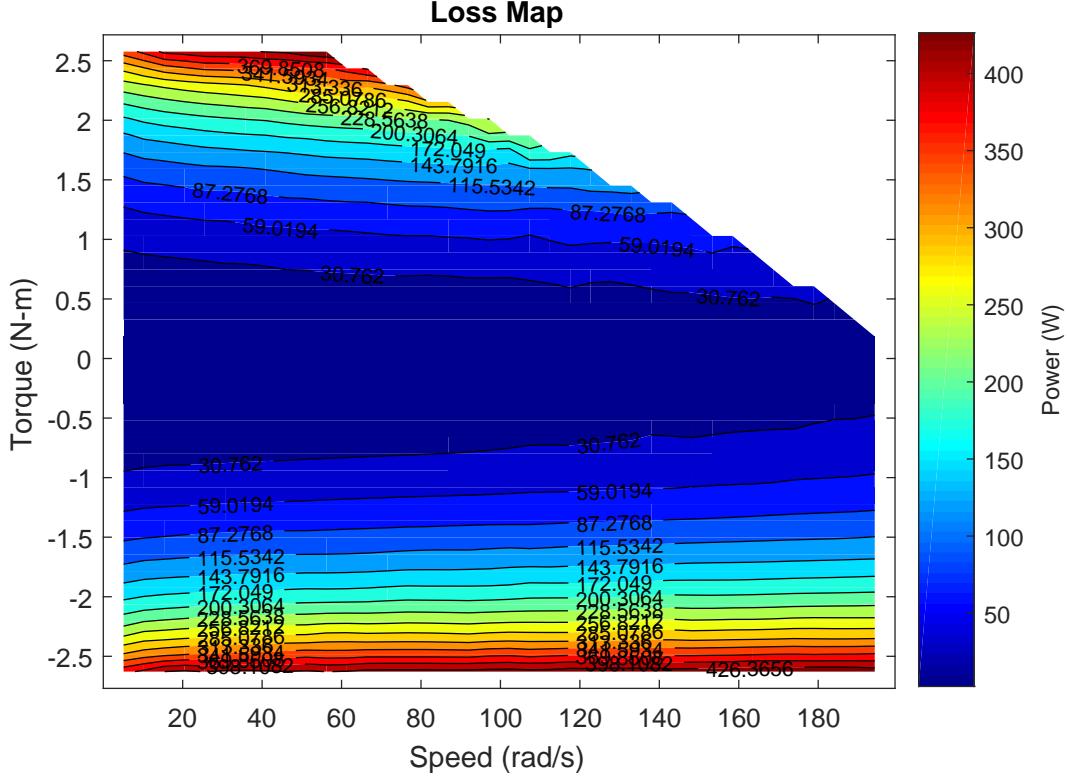


Figure 2-19: Loss Map

friction. The gear and bearing friction is well modeled by equation 2.27.

$$\tau_{friction} = -0.09 sgn(\omega) - 0.04 \cdot \tau_{motor} sgn(\omega) \quad (2.27)$$

Where ω is the angular velocity of the output, $\tau_{motor} = \tau_{rotor} \cdot gear\ ratio$, and output torque is $\tau_{output} = \tau_{motor} + \tau_{friction}$. This corresponds to a transmission efficiency of 90-95% over most of the operating space of the actuator. Figure 2-21 shows measured torque output vs. torque command for output rotation speeds of ± 12 rad/s.

The motor has 0.045 N m peak, 0.0228 N m RMS cogging torque at the rotor, corresponding to an RMS ripple of 0.137 N m at the output of the actuator. The cogging torque is primarily composed of harmonics at 1 and 12 times electrical frequency (Figure 2-22). At this point no attempt has been made to compensate gear friction or cogging torque, although a variety of techniques exist to mitigate these effects[15][16] should greater instantaneous torque accuracy prove useful. For locomotion in parti-

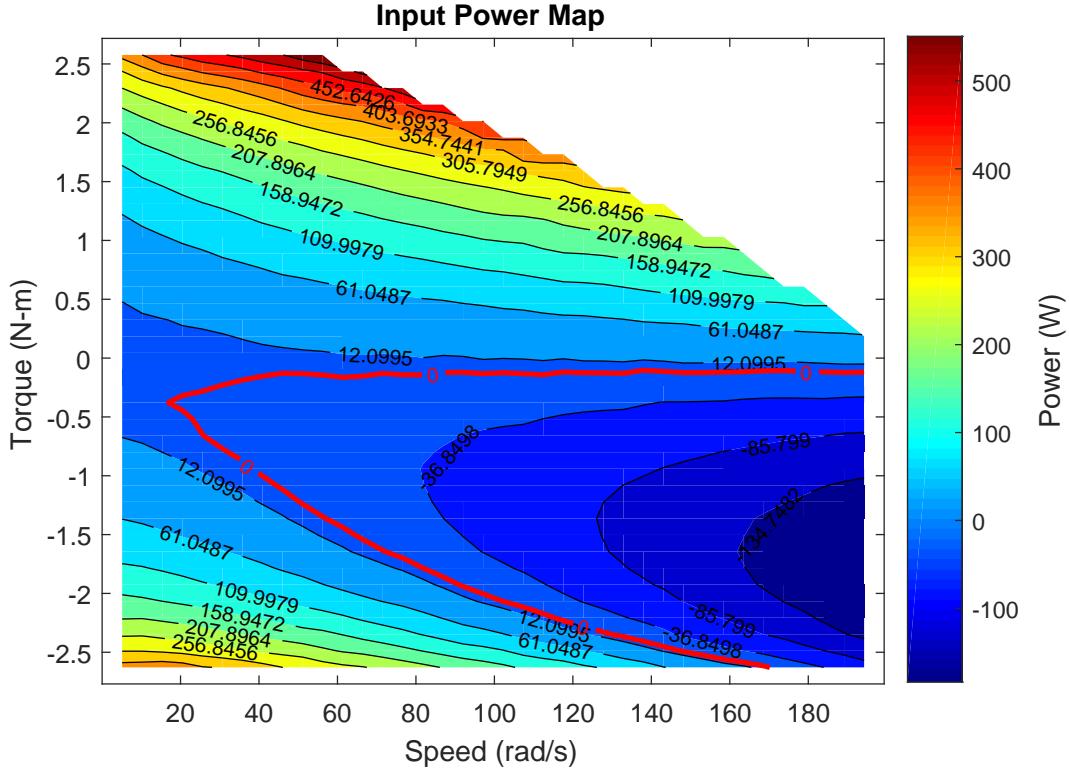


Figure 2-20: Input Power Map. The red contour indicates the zero-power operating points. Outside this region, the actuator sinks electrical power, and inside this region the actuator sources power.

cular, cogging torque should have little effect on high-level robot performance, as it averages to zero and thus does not substantially affect average force and power delivery over a trajectory. Applications like haptics, on the other hand, demonstrated using these actuators in Chapter 4, would greatly benefit from cogging cancellation, in terms of the qualitative "feel" of the haptic feedback system.

2.5.3 Thermal Analysis

To determine steady-state and transient torque limits of both the motor and inverter, winding and inverter temperatures were recorded with a thermal camera for several current levels. This data indicates a thermal resistance for the windings of 1.23 K/W, and a thermal mass of 32 J/K. To keep the winding temperatures below 100 °C, average power dissipation should be kept below 60 W with an ambient temperature of 25 °C. This corresponds to 15.4 peak phase amps, including the 30%

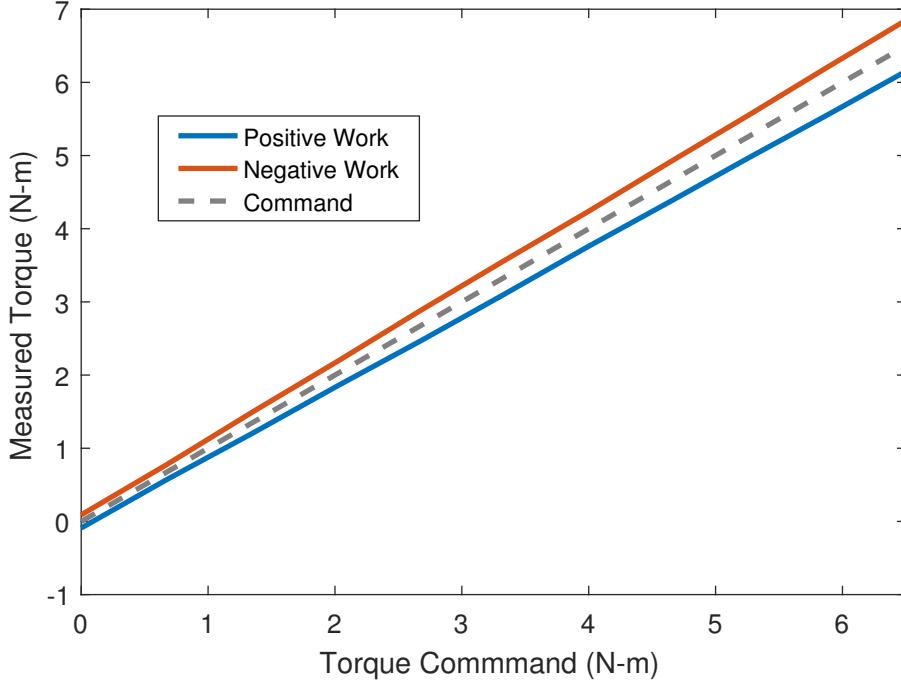


Figure 2-21: Measured output torque vs torque command, averaged over several rotations, for both positive and negative work.

increase in winding resistance expected for a such a temperature rise, corresponding to 6.9 N m at the output of the actuator.

Adding a small fan to provide airflow to the windings reduced the steady state winding temperature to only 17 °C above ambient with 50 watts of power dissipation. This corresponds to a thermal resistance of 0.34 K/W, or nearly a four-fold improvement over the case with no airflow. If higher continuous torque were required, similar cooling should roughly double the continuous torque capability of the actuator.

Throughout testing, inverter temperatures were observed to be significantly lower than winding temperatures, reaching a maximum of 45 °C. While the actual MOSFET junction temperatures are likely higher, the windings appear to be by far the thermally limiting component in the actuator.

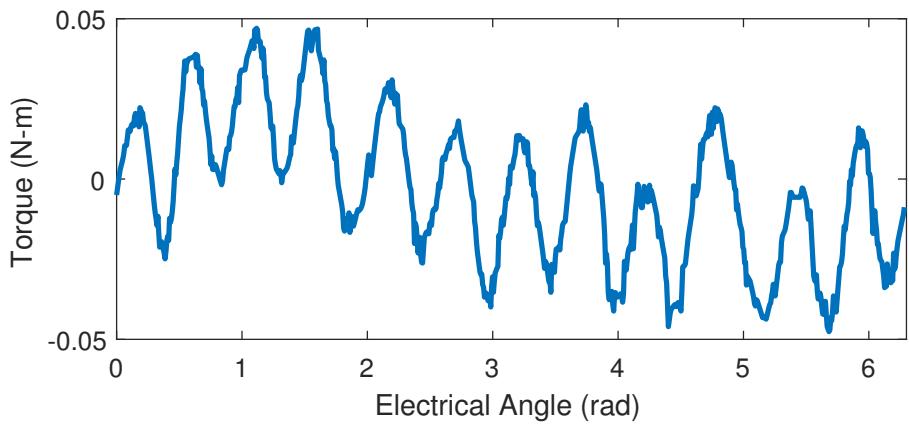


Figure 2-22: Measured rotor cogging torque v s electrical angle.

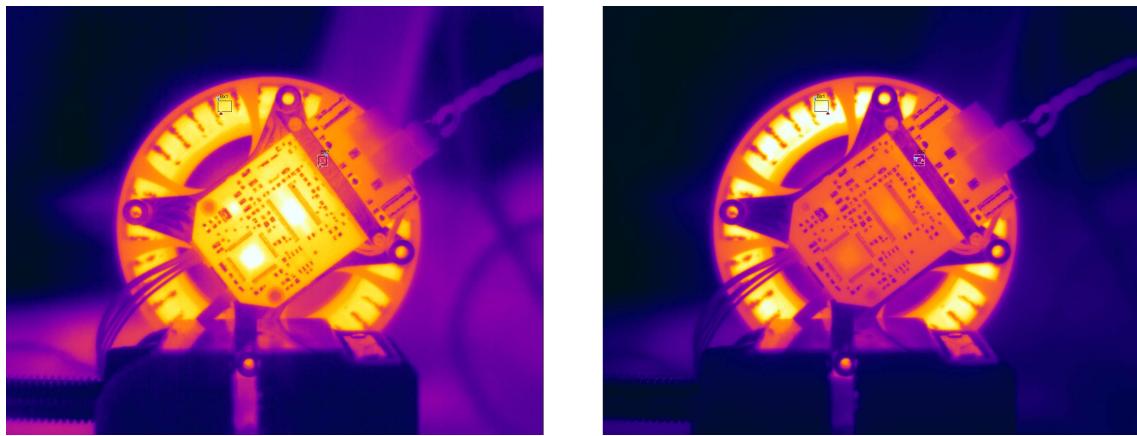


Figure 2-23: Thermal camera images at 5A (left) and 12.5A (right) phase currents. Colors do not correspond between images, as the temperature-color mapping automatically scales.

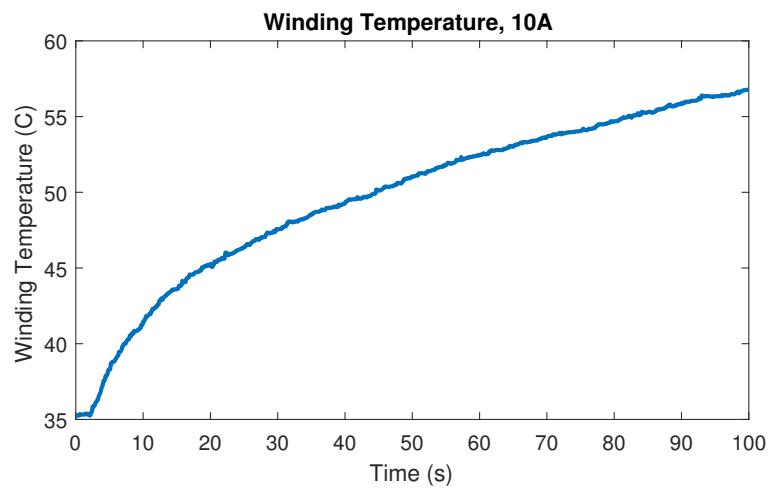


Figure 2-24: Winding temperature during a 10A step.

Chapter 3

Quadruped Platform

One of the intended applications of the actuator module was to build a small, low-cost quadruped platform with similar dynamic capabilities to the MIT Cheetah 3 Robot. The quadruped is roughly 60% the size of the MIT Cheetah. Weighing 9 kg including battery, it can produce a vertical force-to-weight ratio of 1.6 bodyweights *per leg* with the legs fully collapsed, and a horizontal foot-to-body linear speed of 15.6 m s^{-1} with the legs at full extension. While we have only performed a few experiments on the machine at this point, so far its dynamic capabilities look very promising.

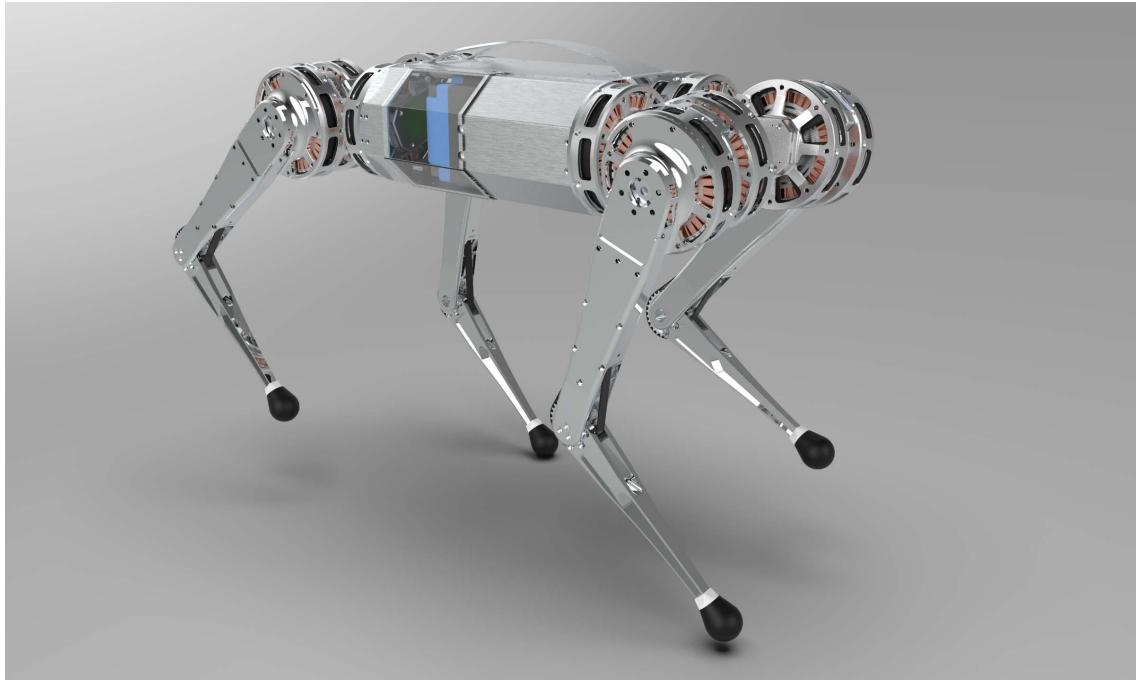


Figure 3-1: CAD Rendering of the quadruped

3.1 Mechanical Design

3.1.1 Legs

Leg design has a huge impact on the final performance bounds of a legged machine. The two main objectives driving this leg design were to:

- Minimize leg mass and inertia
- Maximize leg range of motion and workspace

while satisfying the necessary strength and stiffness requirements of the links.

A low-mass, low-inertia leg with high bandwidth actuators allows for many control simplifications. For the majority of controllers run on the MIT Cheetah robot, we ignore the mass and inertia of the legs for the purpose of control, thanks to these features. This lets us, for example, assume that ground reaction forces $F_{foot} = (J^t)^{-1}\tau$, where J is the leg Jacobian matrix and τ are the joint torques - i.e. ignoring the leg dynamics for force control. Similarly, the affect of the swing-legs on the body of

the robot can also usually be ignored, since the body inertia is so much larger than the leg inertia. These simplifications have proved remarkably effective for control of the MIT Cheetah robots: The locomotion controllers which have been implemented on Cheetah 3 (Convex MPC [11]), Policy-Regularized MPC [10], Balance QP [9]) all use an internal model which neglects leg dynamics, and have demonstrated incredibly dynamic behavior, like blind stair-climbing, 3 m s^{-1} galloping, and impressive robustness to kicks, pushes, and other disturbances. Although pushing to the very limits of robot performance often does require considering the "full" robot dynamics, such as the flipping controller discussed towards the end of this chapter, many useful behaviors work using much simpler models, thanks to this approach to the mechanical design and actuation of the robot.

The wide range of motion of the legs designed greatly increases the utility of the legs, both for locomotion and other tasks. Because of the leg configuration, the quadruped has no preferential direction - it can operate identically forwards, backwards, or even upside-down. The ability to flip the orientation of the knees will allow the robot to climb up or down stairs without the lower links colliding with the edges of the stairs. Since the hip joints are able to rotate $\pm 270^\circ$ from horizontal, the legs can be raised above the body of the robot and potentially used for simple manipulation or climbing over very tall obstacles. The over $\pm 90^\circ$ range of motion of the abduction/adduction joint means the robot can walk with its body rolled 90° , allowing it to fit through gaps as narrow as around 120 cm. See Figure 3-3.

The leg is serially actuated, but the actuator which drives the second link is placed coaxial to the hip actuator, rather than at the knee joint. This minimizes the inertia added to the first link by the knee actuator. The second link is actuated by a belt transmission which passes through the hollow upper link of the leg, and provides an additional 1.55:1 reduction between the actuator and lower link, to increase knee torque. The belts used are Gates Poly Chain belt with aramid tensile members. These belts are designed for high force at relatively low speeds, and high shock loading capability. A pair of belt tensioning arms on the inside of the leg allow adjustment of the belt tension with two set screws. The tensioning arms have a pair of small

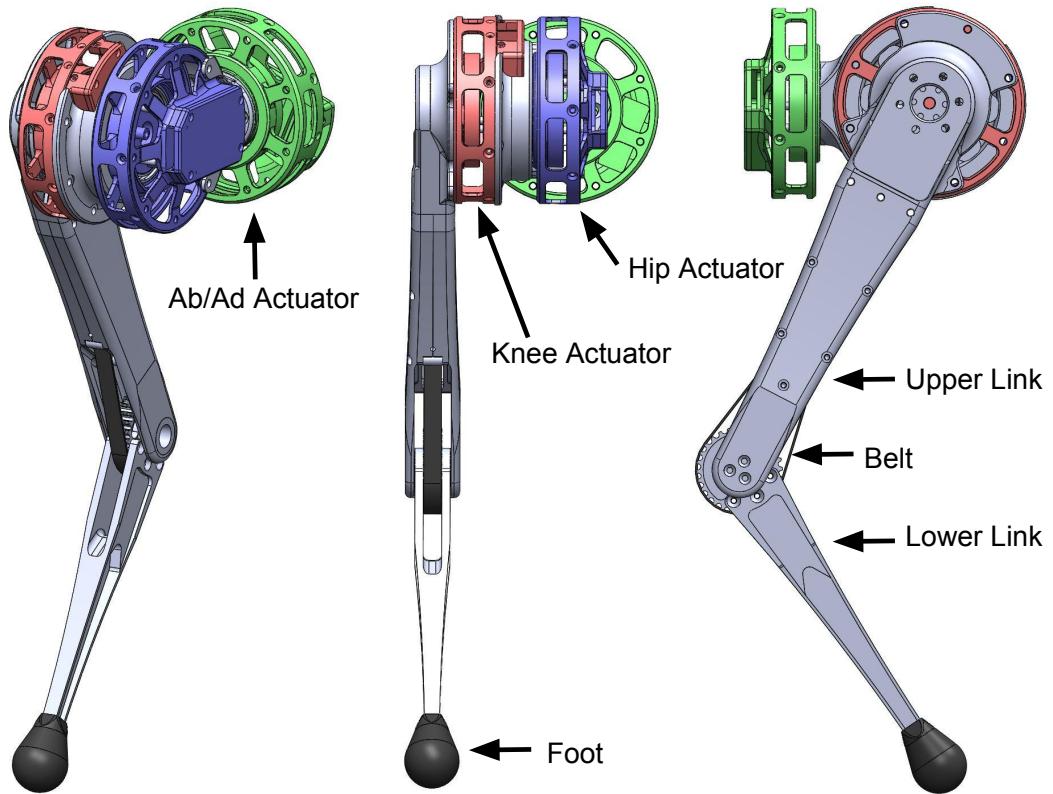


Figure 3-2: CAD Diagram of one leg

bearings on the end which roll against the back surface of the belt, and push the belt inwards to increase the length of the belt path.

The use of a belt transmission between the knee motor and the knee joint is a major design compromise, in terms of force control bandwidth. Although the belt used is probably the stiffest available in this size, it is still far less stiff than a metal linkage would be, because of the compliance in both the belt tensile elements and the belt teeth. As a result, the **rotor-belt resonant frequency** is only 30 Hz. This effectively limits the open-loop torque control bandwidth of the knee to that resonant frequency.

The Cheetah 3 robot uses a roller chain transmission between the motor and knee, and as a result has a similar low-frequency rotor-chain resonance at around 30 Hz, which we do not model in our dynamics simulations. In running experiments on the Cheetah 3 hardware, we have not observed this unmodeled compliance to be high-



Figure 3-3: A variety of configurations possible with the wide range of motion of the legs. Top Left: Turn-on configuration. In this configuration the legs are correctly zeroed for turn-on. Top Right: The robot can be folded up for transportation. Bottom Left: The ab/ad joints have enough range of motion for the robot to walk with the body rolled $\pm 90^\circ$. Bottom Right: The legs can be lifted above the body, for climbing very tall obstacles or manipulation.

level performance bottleneck. Should the compliance caused by the belt transmission turn out to be a limiting factor for robot performance, it could be replaced with a linkage, at the cost of the additional gear-ratio provided by the pulleys, and range of motion, or a wider belt and/or larger diameter pulleys of the same type, with only a slight robot weight and width penalty.

The legs links were designed for very light weight and high stiffness in the directions in which they will experience loads. The lower link, which only experiences bending loads, due to the spherical foot, is roughly an I-beam in cross section, with very thin walls. The leg tapers significantly down its length, as the bending moment on the link due to foot-forces decrease closer to the foot. The knee pulleys were purchased and post machined. The lower link pivots on two thin-section bearings pressed into the knee pulley. A spacer sits between the inner races of the knee bearings and

is sized such that when the leg is assembled, the knee bearings are pre-loaded in a back-to-back configuration. The entire lower link, without the foot, has a mass of only 55 g.



Figure 3-4: Completed lower link with bearings(Left), and cross section (Right)

The upper link experiences both bending loads and torsion, and needs an internal path for the belt which actuates the lower link. The upper link is a two part shell design, with two thin-walled halves which fasten together along the edges, forming a continuous tube. Bosses around all the fasteners align the two halves and take the shear load at the seam between the two legs when the leg is in torsion or lateral bending, ensuring the fasteners between the two halves are not under shear, and that the seam cannot slip. Most of the upper link has a wall thickness of only 1.5 mm, so the two halves of the upper link, plus knee axle, together have a mass of 92 g.

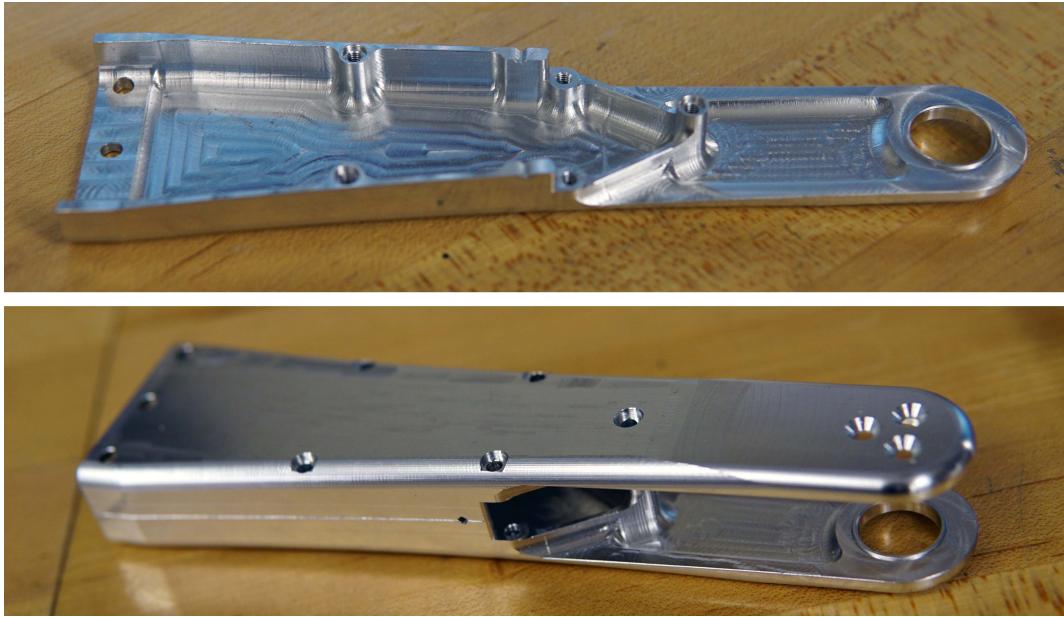


Figure 3-5: One half of the upper link (Top), and the two assembled halves (Bottom)

At the knee joint, the knee axle press-fits into one half of the upper link. This axle passes through the inner races of the knee bearings, and supports the lower link. Although these could have been machined as one part, it would have nearly doubled the thickness of the stock required, so the parts were made separately and press-fit and bonded together with retaining compound. The knee axle prevents the two halves of the upper leg from twisting at the knee joint, substantially increasing the torsional stiffness of the upper link.

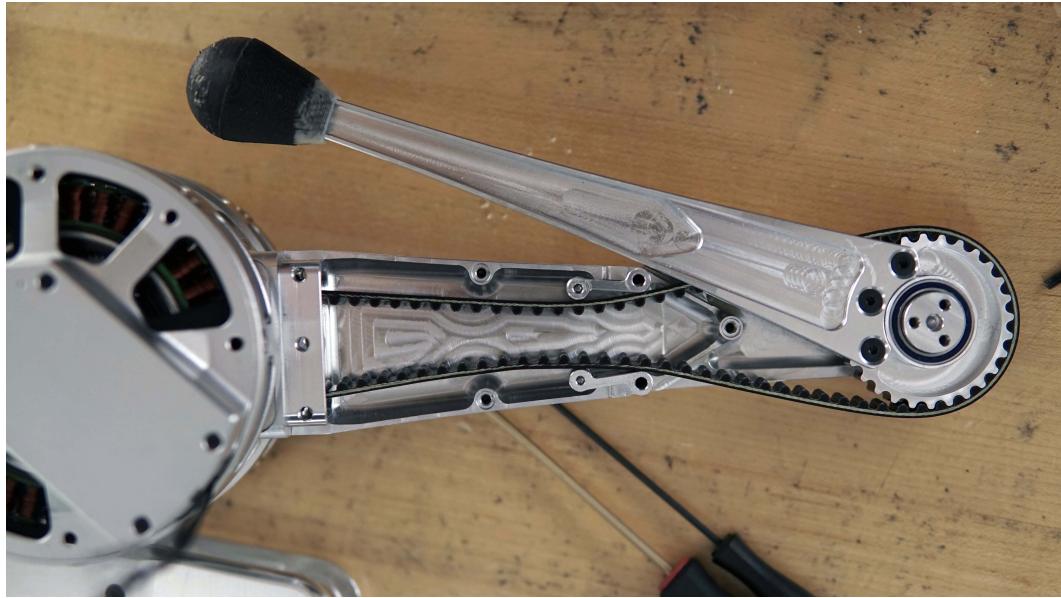


Figure 3-6: Belt path through the upper link

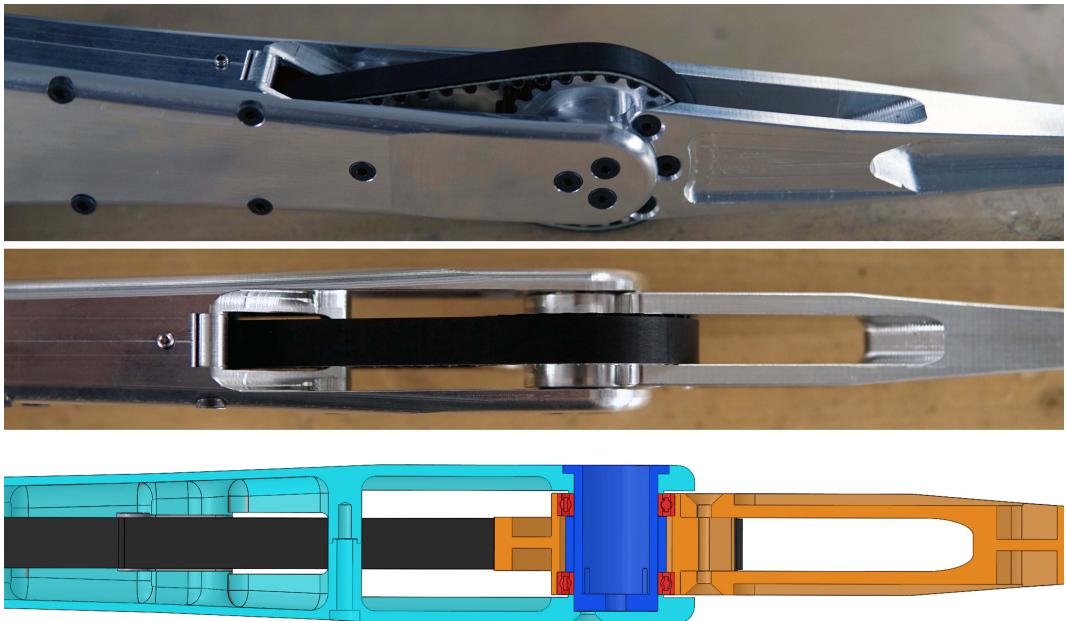


Figure 3-7: Assembled knee joint. Cross section (Bottom) with the knee axle and bearing spacer in dark blue, bearings in red, upper link in light blue, lower link in orange, and belt in black

A short link connects the abduction motors to the hip motor, using the housing of the hip actuator itself as structure for the leg. This allowed the connecting link to be extremely light, at only 30 g.

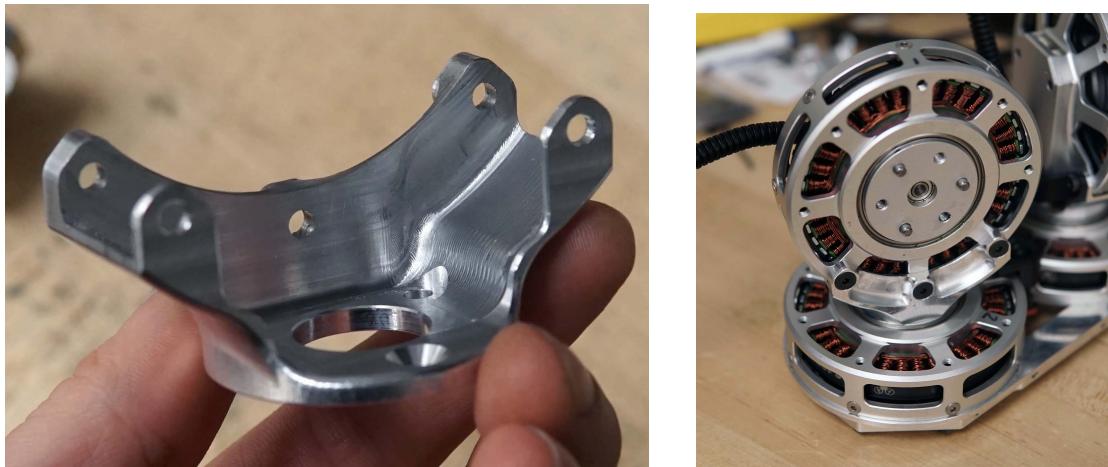


Figure 3-8: Abduction to hip link (Left), attached to both actuators (Right)

Similarly, the hip connects to the upper link of the leg through the knee actuator, using the housing of the knee actuator as leg structure. Originally a configuration

more like Cheetah 3 was used, with the leg in-between the hip and knee actuators, but this design proved significantly more compact, and makes the robot's crash zone the outside of the leg links, rather than the knee actuator.



Figure 3-9: Hip to leg link

All the leg parts were CNC machined from 6061 T6 aluminum billet in house, on the lab's HAAS Super Mini Mill. Should we decide to produce a large quantity of these machines, the leg design should be easily adaptable for injection molding in a fiberglass or carbon fiber reinforced plastic, e.g. PA6-GF30 or PA6-CF30.

3.1.2 Body

With the motor control electronics built into the actuators, fitting the necessary components in the body of the robot was surprisingly easy. These were: A 24V Lithium Ion cordless drill battery; and UP Board embedded computer; a Vectornav VN100 Rugged IMU; a power supply for the computer and electronics; computer to actuator communication electronics; and plenty of space for wiring. The internal components were arranged so that there is sufficient space within the body to fit one or two more small form factor computers like the UP board or an NVIDIA Jetson, which will likely be necessary for adding vision and other computation-heavy sensing or planning to the robot in the future.

The robot's body was fabricated from 1.5 mm aluminum sheet, waterjet cut and

bent to shape. The sheet metal forms a light-weight tube between the front and rear pairs of legs, which is extremely stiff in bending and torsion. Two internal plastic ribs provide fastening points for the sheet metal and internal components like the IMU and battery, and help prevent the sheet metal from denting from side-impacts the robot might experience. A cutout in the side of the body with an easy-to-remove cover gives easy-access to the battery, a curved sheet metal handle on top makes the robot easy to pick up and carry around.



Figure 3-10: Body components made from waterjet-cut and bent 1.5 mm 6061 sheet

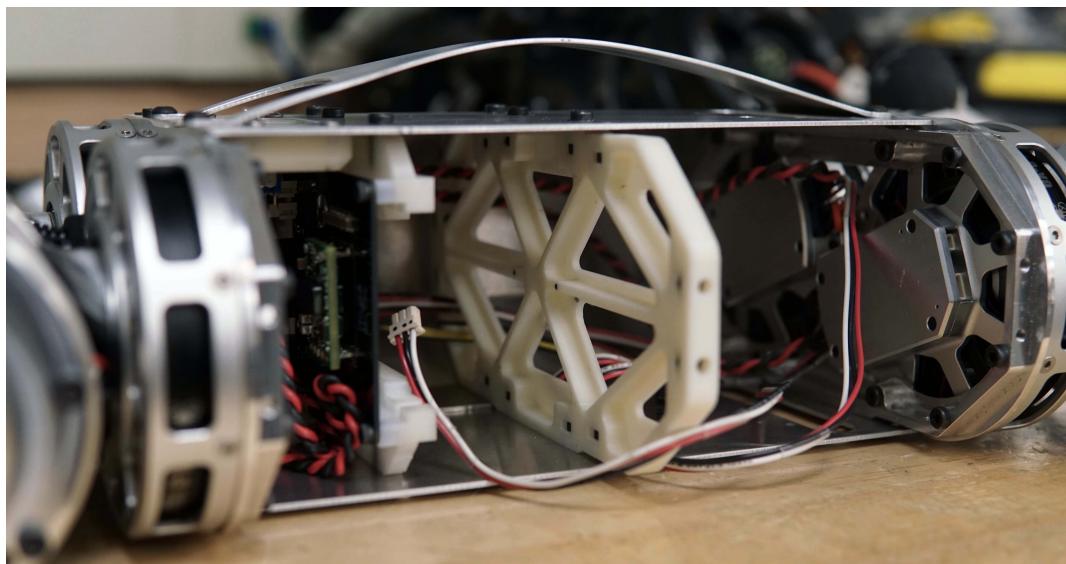


Figure 3-11: Inside of the assembled body, showing the internal frame-stiffeners, battery mount, and handle on top.

3.1.3 Feet

Some guiding principles for design of the feet were:

- Grip: High coefficient of friction against most surfaces
- Wide range of contact angles: Since the legs have a very large range of motion, the feet should have traction on the ground at a similarly wide range of leg angles.
- Damping: Although a well-damped foot means energy is dissipated in each footprint, this seems preferable to a "bouncy" foot, which might bounce in-and-out of contact.
- Durability: Very soft foot materials, like Smooth-On VytaFlex 20A urethane, have not lasted long on Cheetah 3. Currently on Cheetah 3 we use VytaFlex 60, which has proved reasonably durable, but has very low damping.

It was observed that squash balls fit many of these criteria, and are of approximately the correct size for the robot, at around 40 mm in diameter. The feet were made from modified squash balls: A 3D-printed insert fits over the end of the lower link of the leg, and through a hole cut in the the squash ball. The gaps between the insert and the inside of the ball are filled with Smooth-On FlexFoam-It 23, a castable expanding foam, with similar properties to the EVA foam used in many shoe midsoles. Together these make for a soft, well damped, grippy, and hopefully durable foot.



Figure 3-12: Foot, and cross section(Left). The plastic insert has holes which are filled by the casting foam.

3.1.4 Wiring

Wiring and cable management is an oft-overlooked and un-glamorous part of every complex electro-mechanical assembly. Because of the range of motion of this robot, the wiring to the knee motors was particularly important, so that the cables are not placed under stress and cannot accidentally become entangled over any of the 540° of rotation of the hip motor.

All external leg wires are housed within corrugated plastic sleeves, which are rigidly fixed at either end by 3D printed attachments. These ensure that none of the bending the cables must undergo happens nearby the connectors, where they would eventually fatigue and fail.



Figure 3-13: Wiring harness for one leg. Flexible wires are housed within protective corrugated sheaths, which are fixed by 3D printed guides at each end, strain-relieving the connectors.



Figure 3-14: Installed wiring harness for one leg.

3.2 Electronics and System Architecture

3.2.1 Embedded Linux Computer

For high-level locomotion control, an UP Board single board computer was used. The UP board [31] is a small, low-power embedded x86 computer with a 4-core Intel Atom x5-Z8350 processor, 4 gb RAM, and roughly 5 W peak power consumption. As it has far less computing power than we have available on Cheetah 3 (Sandy Bridge

Core i7, 8 Gb ram), some code restructuring and performance improvements will be required to be able to run the same controllers on it. The robot physically has space for up to 2 more computers of similar size, which will likely be necessary for adding computer vision and other computation-heavy high level control in the future.

The UP Board runs the Linux kernel with a PREEMPT-RT patch to achieve pseudo-realtime code execution. The various pieces of software running on the computer communicate over Lightweight Communication and Marshalling (LCM) [18]. LCM also allows for fast data logging, in this case 1 kHz, of every command, sensor measurement, internal control variable, etc, which is invaluable for the controller development and debugging process.

3.2.2 SPIne

The UP Board has no built-in CAN interfaces, so to communicate with the leg actuators intermediate electronics were required. The communication interface, called the SPIne, sits between the computer (the "brain") and the limbs, and communicates to the computer over a SPI (Serial Peripheral Interface) bus. Two STM32F446 microcontrollers are on the SPI bus, and each of those microcontrollers generates 2 CAN networks. As this is the same microcontroller used on the actuators for motor control, much of the communication code could be reused. The logic level CAN TXD and RXD signals from the microcontroller pins are passed through RF digital isolators before the CAN transceivers, so the communication is electrically isolated. A 5V isolated DC-DC converter for each CAN bus provides power and ground reference to the CAN transceivers. The SPIne also includes an XBEE radio, for wireless control of the robot. The PCB design files in EAGLE and firmware for the SPIne can be found in Appendix A.

3.2.3 Battery and Power Supplies

The robot is powered by an on-board Kobalt 24V Max 5 Ah Lithium Ion battery, sold for use in cordless power tools. The off-the-shelf power tool battery was chosen

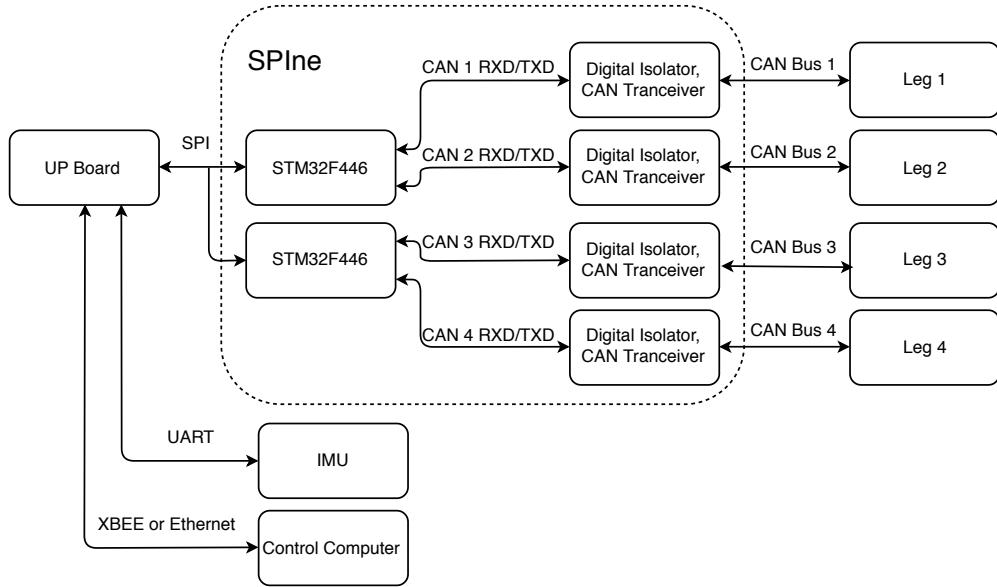


Figure 3-15: Diagram of the robot's communication architecture

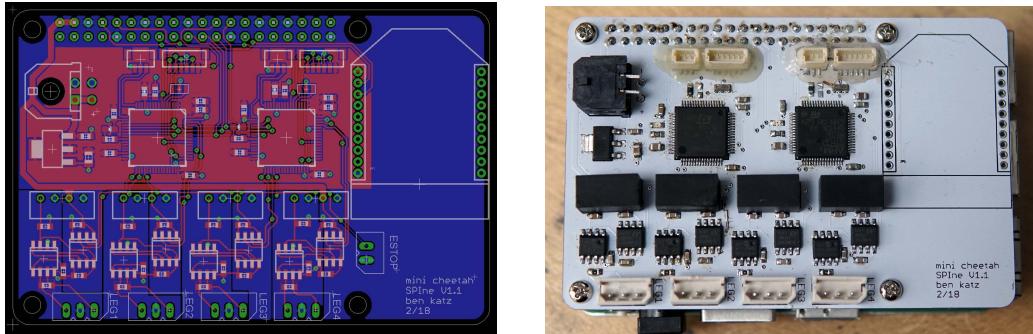


Figure 3-16: PCB Layout of the SPIne, on the left, and assembled board on the right.

for ease of use and ruggedness. It already has a built in battery management system (BMS), under voltage protection, and easy to use charging system. Its mechanical interface allows batteries to be swapped out in seconds when they need charging.

The specific battery used was chosen for its slightly higher voltage and lower cost than typical consumer cordless drill batteries. Despitg this, the battery uses very high quality cells internally: Samsung INR18650-25R cells, which have a capacity of 2.5 A h, nominal voltage of 3.6 V, maximum discharge current of 100 A, and continuous discharge current of 20 A. The battery pack is a 6S2P configuration of cells, giving the pack a capacity of 108 W h, and continuous output power of 960 W, which is more

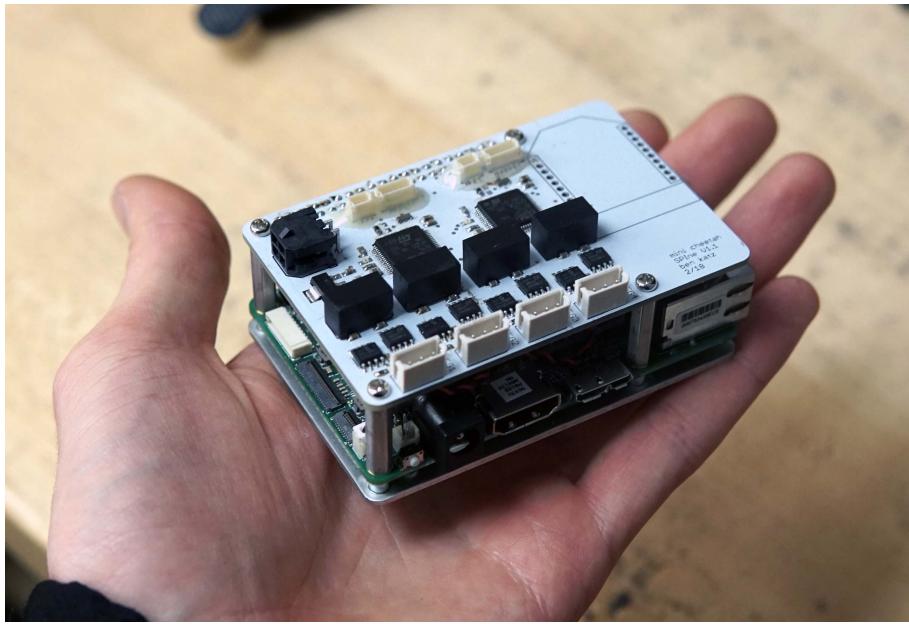


Figure 3-17: SPIne and UP board assembled together

than sufficient for this machine.

To interface with the battery, a custom power supply was built which mates to the battery terminals. The power supply uses a 5 V, 30 W isolated DC-DC converter to generate an isolated power supply for the computer and logic. This power supply has sufficient overhead to power significantly more than the UP Board used, so adding more computers in the future should require no changes on the power supply side. To switch power to the motors and logic on and off, low-side pass transistors were used, which can be switched on and off by low-current mechanical switches. Turning on motor power requires some care, as the 12 motor controllers together have substantial input capacitance. The total energy storage of the input capacitance is around 1.5 J at 24 V, so the switch which turns on the motor power must be able to dissipate this amount of energy, as charging a capacitor through a resistor is fundamentally 50% efficient. Also, instantaneously turning on motor power would cause a huge inrush of current as the capacitors start charging through a very low-resistance connection. To mitigate these effects, a precharge circuit was designed for the actuator power switch. The precharge circuit first slowly charges the capacitors through a power resistor, over a time period of 80 ms. Then another transistor is automatically switched on,

with no series resistor, which passes high current to the actuators. See Figure 3-18 for a schematic diagram of the circuit.

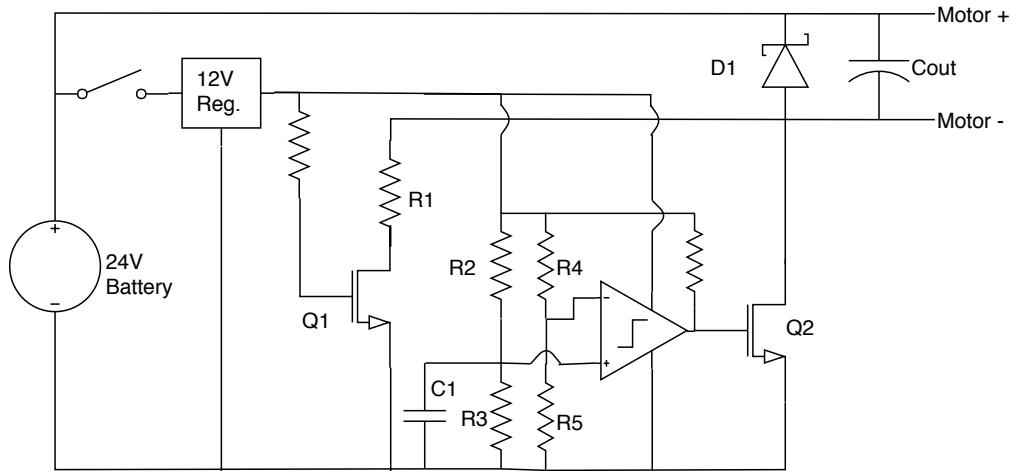


Figure 3-18: Precharge circuit for turning on battery power. When the power switch is closed, a 12V supply is turned on which charges the gate of MOSFET Q1, turning it on. This allows the output capacitance C_{out} to be charged through power resistor R1, limiting the charging current to a maximum of V/R_1 . After a fixed delay set by the time constant of the RC filter made by R2, R3, and C1, the output of the comparator switches on, turning on MOSFET Q2, which bypasses the power resistor. Flyback diode D1 allows current to continue flowing when the switches have been turned off, so that inductance on the output does not cause voltage spikes which would damage the pass transistors or motor drivers.

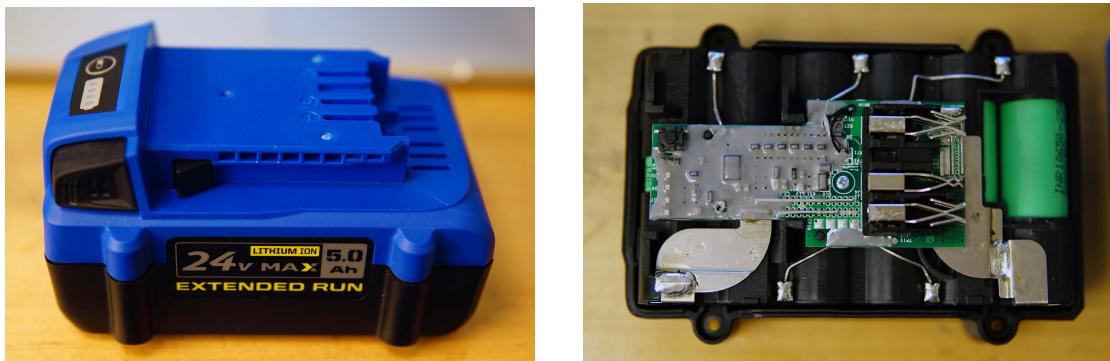


Figure 3-19: The Kobalt 24V 5 Ah battery. Housing removed on the right, showing the built-in BMS, output terminals, and 18650 cells.

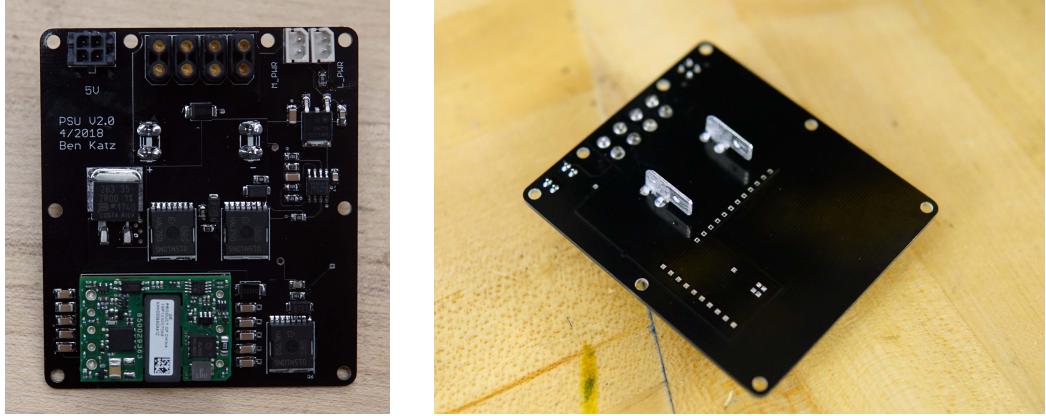


Figure 3-20: Robot power supply board, which generates an isolated 5V logic supply, distributes power to the legs and computer, and has pass transistors for turning on and off motor power and logic power. The underside of the power supply (right) has board mount FASTON terminals which mate with the spades built into the battery.

3.3 Control Experiments

Although the entire robot has so far only been operational for a short time, we have performed a few control experiments on the robot, demonstrating impressive dynamic capabilities.

3.3.1 Flipping

The first controller implemented on the quadruped was a back-flipping controller. To our knowledge, this is the first example of a quadruped robot doing a 360° flip, although the 2D and 3D Biped robots from the MIT Leg Lab and the biped Atlas from Boston Dynamics have done a somersault [17] and a backflip off a raised platform [33], respectively.

The backflip was generated with a nonlinear optimization on a 5-link sagittal plane model of the robot. The trajectory optimization generated a set of joint torque profiles and joint states for the flip. The open source library CassADi [28] was used to set up the trajectory optimization, which was solved using IPOPT [29]. To execute the flip, the joint torques were used as a feed-forward, and the joint states used as the input to joint PD controllers. Videos of the robot flipping can be found in Appendix

B.

The flip was generated by minimizing the cost function

$$\sum^n Q_{error}^T Q_{error} \quad (3.1)$$

Where Q_{error} is the difference between the desired final joint configuration of the robot Q_{end} , and the joint configuration at each optimization timestep Q_n .

subject to the constraints:

- Initial and final positions and velocities
- Zero force on the feet which are not in contact with the ground
- Minimum normal force f_{min} when the feet are on the ground, to prevent the feet from slipping
- Feet on the ground stay at their initial position and do not slip
- Knees do not penetrate the ground
- Ground reaction forces at the feet lie within friction cones
- Maximum joint torques
- Speed-dependent torque limits from actuator dynamics
- Euler integration of the robot dynamics

This cost function tends to keep the legs from swinging around excessively while the robot is in the air, while the final angle and body height constraints ensure the robot has completed the flip above the ground by the end of the time interval. The contact states of the robot (4 legs on the ground, 2 legs on the ground, 0 legs on the ground) were pre-determined, and the optimization set up using direct transcription. A link to the optimization code used to generate the flip can be found in Appendix A.

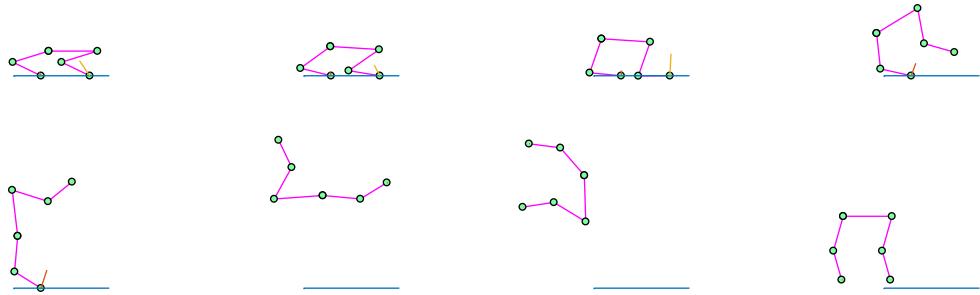


Figure 3-21: Sequence of frames from the optimization on a 2D sagittal plane model of the robot.

The landing controller very simple - we were able to successfully land the flips by using joint-PD control to a wide stance, with moderate stiffness and high damping, making the configuration of the robot inherently very stable.

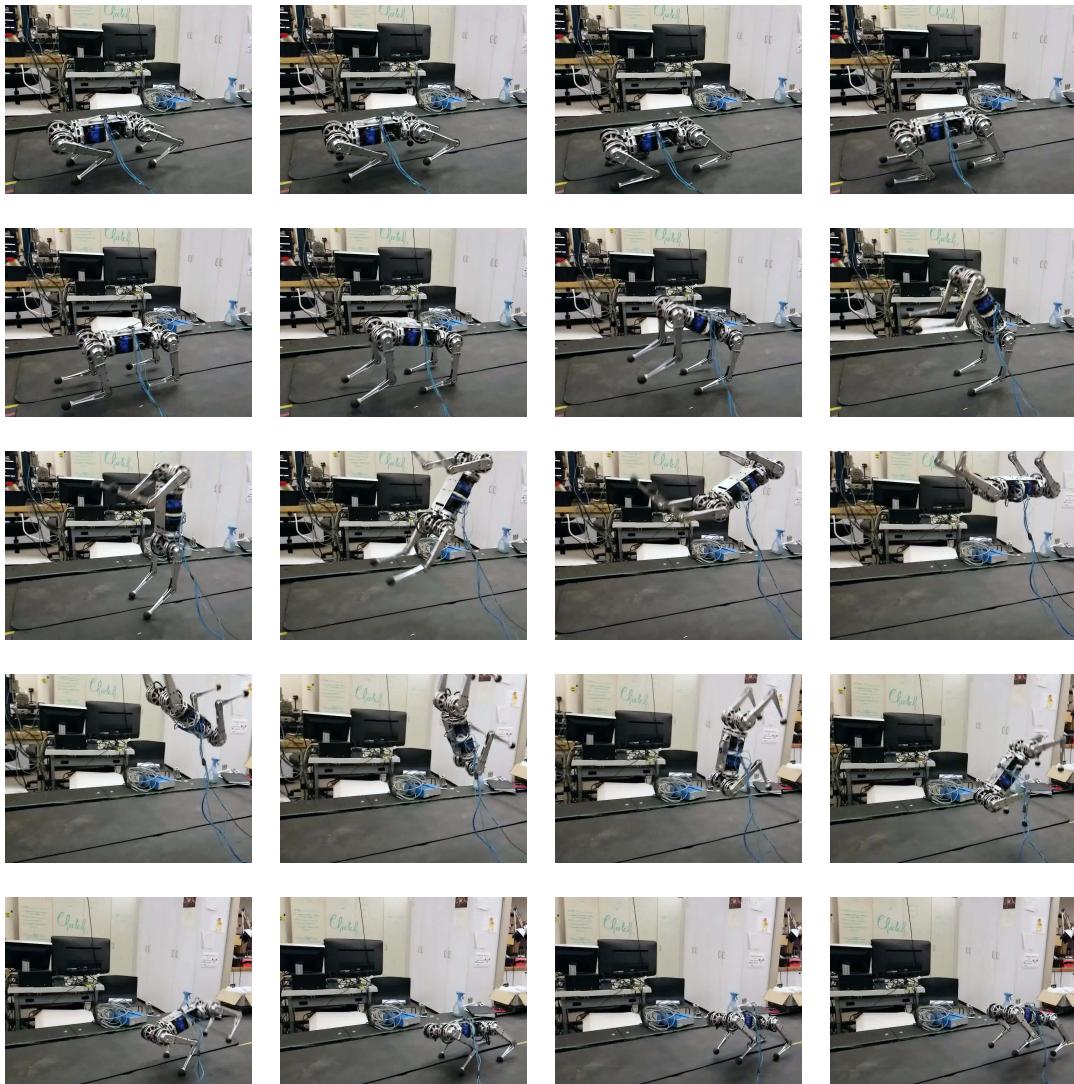


Figure 3-22: Time series of the quadruped doing a back-flip

During the flip, the robot's COM reached a height of around 0.65 m. Joint torques and output power required for the flip are shown in figure 3-23. Mechanical output power of the robot hit a peak of 690 W, just before the rear legs take off the ground.

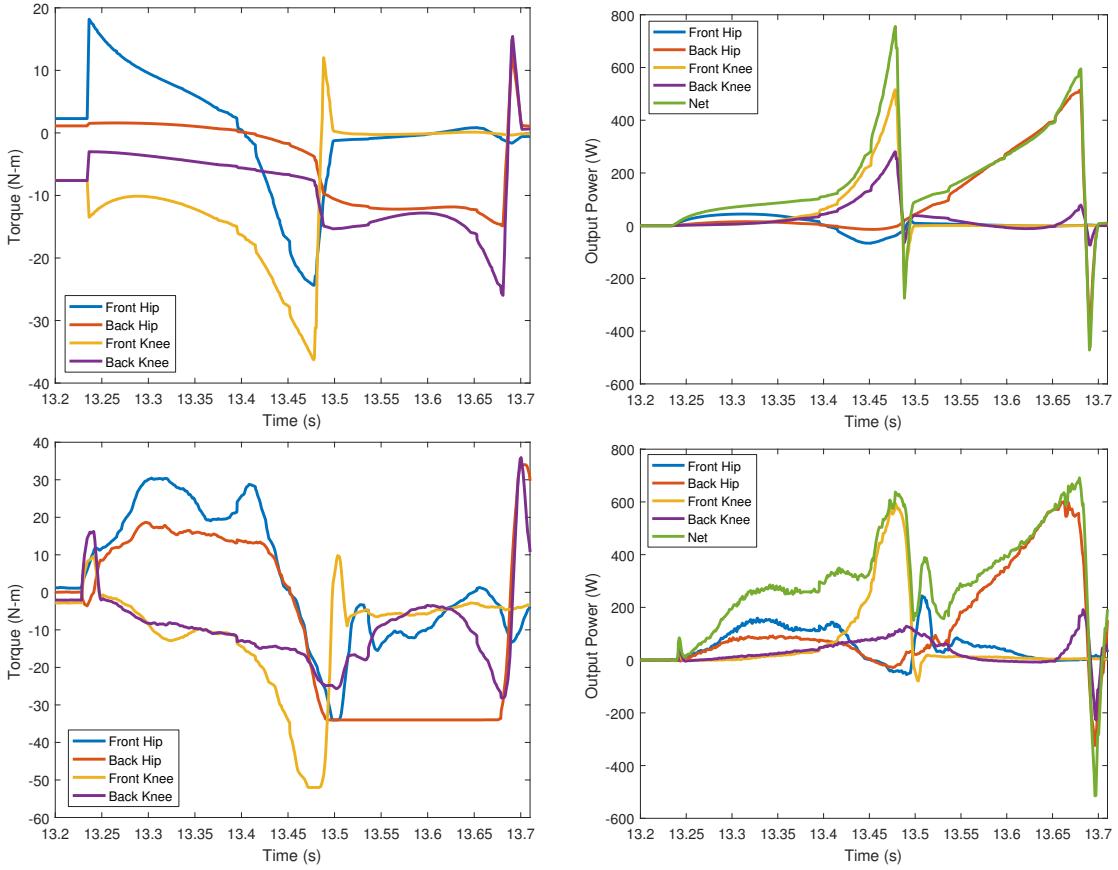


Figure 3-23: Joint torques (Left) and joint output power (Right) during the takeoff of the backflip. On top is the output from the 2D optimization, and on the bottom, the data collected from the actual robot during the flip. Values are for each pair of legs, so individual joint torques and powers are half those plotted.

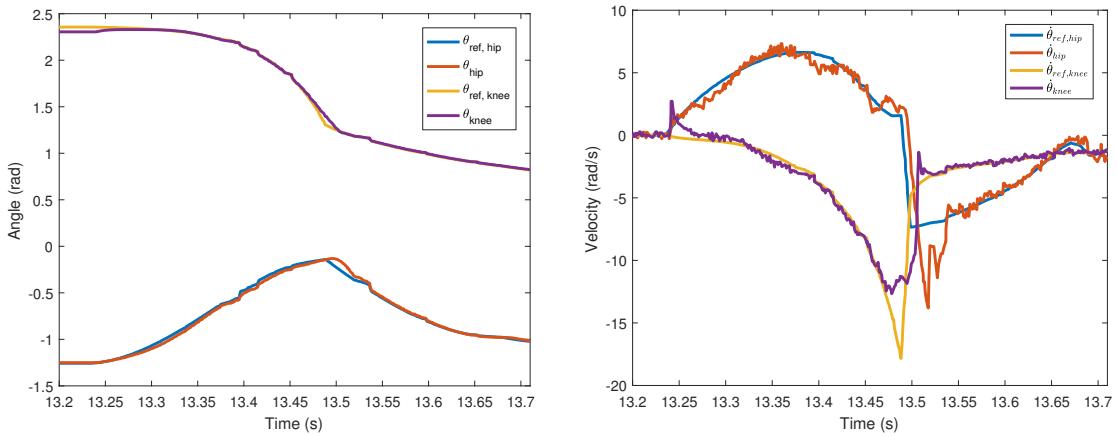


Figure 3-24: Joint position (Left) and joint velocity (Right) tracking during the takeoff of the backflip, for the front left leg.

Using joint PD control to track the flip worked on flat terrain, but is not a particu-

larly robust feedback controller for these behaviors. For example, if the robot started out on a slope, or anything but flat ground, this controller would almost certainly not work. A better alternative would likely be to use a controller similar to the balance controller described in [9] or [27], controlling the position, velocity orientation, of the body of the robot, rather than individual joint angles. At this point, however, we have not yet had the opportunity to try out any more interesting controllers.

Chapter 4

Other Actuator Implementations

4.1 Bilateral Teleoperation and High-Force Haptics

The actuation requirements for legged locomotion share many similarities with the actuation requirements for haptic feedback interfaces and bilateral teleoperation: high bandwidth torque control, high backdriveability, low reflected inertia, stiff mechanical structure - with a key difference being specific torque, which is necessary for legged locomotion but not for the other applications (which do not need to support their own weight). For teleoperation and haptics, these design principles are exemplified by the MIT Manus Arm, [26], the Phantom haptic interface [6], and the WAM arm for teleoperation [25].

4.1.1 Hardware

The teleoperation system was built from two of the legs discussed in the previous chapter. The teleoperation setup was a testbed for the leg design, rather than a purpose-built haptic interface, so there are certainly components of its design which could improve its haptic feedback performance. For example, while the belt-driven second link serves to increase the range of motion of the leg, this feature is probably not necessary for haptics applications, and could easily be replaced with a stiffer,

lower friction linkage. The two legs are connected over their CAN interfaces to a central microcontroller, which controls the two legs.



Figure 4-1: Teleoperation and haptics setup consisting of two 3-DOF arms, with feet replaced by smooth spheres.

4.1.2 Bilateral Teleoperation

The hardware design lends itself to high-force bilateral teleoperation, similar to [25]. The two arms are virtually coupled to each other, so either can act as either the master (the haptic interface) or the slave (the manipulator) in the system. By not using a force sensor at the end effector for force control and instead relying on proprioception, the two arms can be virtually coupled with stability during high-force impacts.

The two arms can be coupled in joint-space by running PD position control loops on each joint, where the setpoints are the current states of the corresponding joints on opposite arms. An additional local joint damping ("passivity" [23]) term is added to ensure stability. For each joint i on arms 1 and 2, the control law is:

$$\tau_{1,i} = Kp(\theta_{2,i} - \theta_{1,i}) + Kd(\dot{\theta}_{2,i} - \dot{\theta}_{1,i}) - K(\dot{\theta}_{1,i}) \quad (4.1)$$

$$\tau_{2,i} = Kp(\theta_{1,i} - \theta_{2,i}) + Kd(\dot{\theta}_{1,i} - \dot{\theta}_{2,i}) - K(\dot{\theta}_{2,i}) \quad (4.2)$$

Note that if the gains used are identical on arms 1 and 2, then the motor torques at each joint are nearly identical between the two arms, differing only by the local damping applied. This means that, if joint accelerations and friction are small, then the end-effector force felt by the operator is very close to the end-effector force on the slave arm. Dynamically, the operator feels the friction and inertia of the master side, as well as the friction and inertia of the slave side, virtually coupled through the controller.

Achievable coupling stiffness was largely governed by the communication rate between the two arms. With a 1kHz update rate, a joint-space stiffness Kp of 100 N m rad^{-1} was achievable. Increasing the loop rate by reducing the amount of data sent over the CAN bus to only the strictly necessary signals, it should be possible to achieve significantly higher coupling stiffness between the two arms, improving the force-feedback quality.

See Appendix B for a video demonstration of bilateral teleoperation setup.

4.1.3 Virtual Environments

By employing cartesian impedance control [22], virtual environments can be created, which the operator can feel using the haptic interface. Although the stiffness achievable with the system presented here is similar to the Phantom's 3.5 N mm^{-1} , the maximum force far exceeds that of commercially available haptic systems, at over 200N over much of the workspace.

One of the simplest examples, a virtual frictionless wall in the Y-Z plane can be implemented by the following control law:

$$F_x = \begin{cases} K_{wall}(x_{wall} - x_{arm}) - Kd_{wall}\dot{x}_{arm}, & \text{if } x_{arm} > x_{wall} \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

$$F_y = 0 \quad (4.4)$$

$$F_z = 0 \quad (4.5)$$

$$\tau = \begin{bmatrix} \tau_{q1} \\ \tau_{q2} \\ \tau_{q3} \end{bmatrix} = J^T \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (4.6)$$

Where K_{wall} is the stiffness of the virtual wall, Kd_{wall} is the virtual damping of the wall, x_{wall} is the position of the virtual wall, and J is the arm Jacobian. Once the end effector has crossed the boundary of the virtual wall, the controller applies a restoring force to push it out of the wall. Friction can be easily added to the wall as well, by adding a force in the Y-Z plane proportional to the normal force, and opposite the direction of motion along the wall.

An interesting phenomenon observed while implementing the virtual wall is that the achievable stiffness is highly dependent on the configuration of the arm. Intuitively, this makes sense - the wall acts as a PD control law on the end effector of the arm, but as the configuration of the arm changes, so does the apparent mass of the end effector. In directions where the apparent mass is high, much higher stiffnesses can be achieved before the system goes unstable. But when the apparent end-effector mass decreases, the crossover frequency of the PD loop increases, and eventually goes unstable in the classical sense, from loop delay and other effects. From another perspective, mapping the cartesian stiffnesses into joint space, as the arm Jacobian becomes less favorable for force production at the end effector, the effective joint-space stiffnesses gain increase, eventually causing the joints to go unstable.

To combat this effect, a simple gain-scaling controller was implemented to try to limit the maximum cartesian natural frequency of the arm, to prevent instability.

Assuming that the joint-space inertia is configuration independent (which is actually reasonably accurate for this design), the joint-space mass matrix is given by:

$$M_q \approx \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \quad (4.7)$$

The inverse of the end-effector mass matrix can be easily calculated as:

$$M_x^{-1} = JM_q^{-1}J^T \quad (4.8)$$

Using only the diagonals from the inverse of the mass matrix

$$M_x^{-1} \approx \begin{pmatrix} \frac{1}{M_x} & 0 & 0 \\ 0 & \frac{1}{M_y} & 0 \\ 0 & 0 & \frac{1}{M_z} \end{pmatrix} \quad (4.9)$$

These approximate cartesian masses can be used to scale the maximum cartesian stiffnesses in their respective directions, by limiting the natural frequency in each direction. The directional natural frequencies are given by:

$$K_x = \frac{\omega_n^2}{M_x}, K_y = \frac{\omega_n^2}{M_y}, K_z = \frac{\omega_n^2}{M_z} \quad (4.10)$$

This gain limiting approach allows much higher cartesian stiffness to be achieved where arm geometry is favorable, without risking instability in unfavorable configurations.

4.2 Biped for Human-Robot Balance Feedback

The first machine built using these actuators was a 6-DOF lower body biped robot, built by another member of the lab, which has been used for whole-body teleoperation with bilateral balance feedback [19].

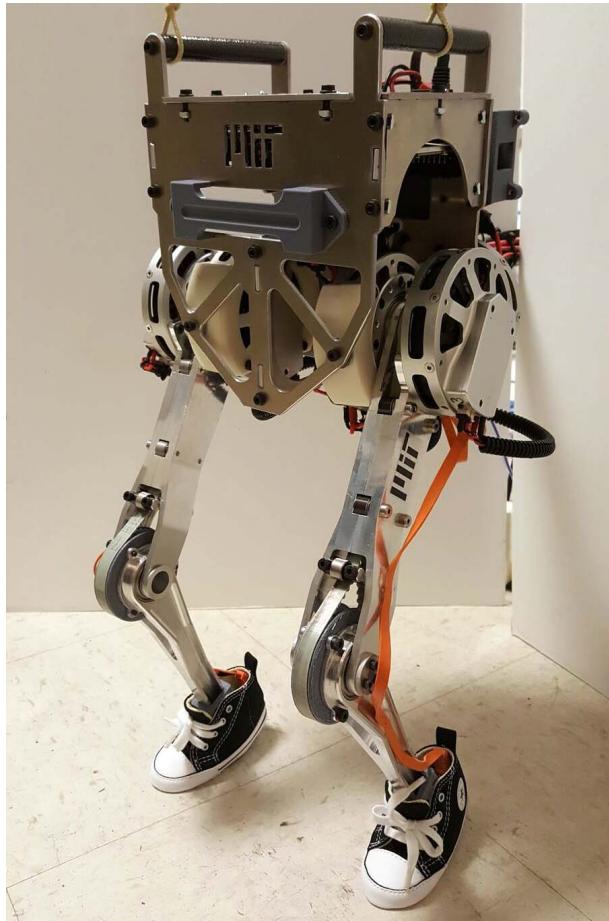


Figure 4-2: Little Hermes, a 6-DOF point-foot bipedal robot. Image used with permission

Chapter 5

Conclusion

This thesis presents the design, manufacturing, and control of a low cost modular robot actuator, intended for use in legged robots and other highly dynamic machines which must interact with the world without fear of impacts. At this point, 26 of these actuators have been built and used about the Lab, between a 6-DOF biped platform used for human teleoperation with bidirectional balance feedback, two 3-DOF robot arms, which have been used as haptic interfaces and for bilateral teleoperation, and for a 9 kg, 12-DOF quadruped robot. These actuators dramatically lowered the design time and cost that would normally be associated with building these complex robotic systems from scratch, and have let us begin testing exciting new control strategies and dynamic behaviors, which would be dangerous and difficult to execute on a larger machine. For example, we have used the quadruped to execute backflips generated by nonlinear trajectory optimization - a maneuver which very few robots other have the dynamic capability for. To our knowledge, this is the first implementation of a full flip on a quadruped robot.

In honesty, at a fundamental level, very little is novel about the work presented in this thesis. Although the core ideas behind the actuator described now are nothing new, hopefully the collection of *engineering work* documented here helps this approach to actuation proliferate, by making it more accessible both in terms of cost and design methodology. Already this work has enabled exciting new experiments on the robot platforms developed, and I have no doubt that in the coming months and years, it

will help us to test countless new control ideas for dynamic machines.

Bibliography

- [1] Sangok Seok, Albert Wang, David Otten and Sangbae Kim. *Actuator Design for High Force Proprioceptive Control in Fast Legged Locomotion* IEEE, 2012.
- [2] Will Bosworth, Michael Farid, Deborah Ajilo, Sangbae Kim, and Neville Hogan *Bounding, jumping, turning and stopping with a small, low-cost quadrupedal robot* Massachusetts Institute of Technology, 2015.
- [3] M. Hutter, C. Gehring, M. Bloesch, M. Hoepflinger, C. D. Remy, R. Siegwart, *StarlETH: A Compliant Quadrupedal Robot for fast, efficient, and versatile Locomotion*, Proc. of the International Conference on Climbing and Walking Robots (CLAWAR), 2012
- [4] M. Hutter et al., *ANYmal - a highly mobile and dynamic quadrupedal robot*, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016, pp. 38-44.
- [5] G. A. Pratt and M. M. Williamson, *Series elastic actuators*, in IEEE International Conference on Intelligent Robots and Systems (IROS) pp. 399–406, MIT, 1995.
- [6] Thomas H. Massie and J. K. Salisbury. *The PHANTOM Haptic Interface: A Device for Probing Virtual Objects*, Computer Graphics and Applications, IEEE 17.5 (1997): 6-10. 13, 22, 72.
- [7] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang and S. Kim, *Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots*, in IEEE Transactions on Robotics, vol. 33, no. 3, pp. 509-522, June 2017.
- [8] H. W. Park, Sangin Park and S. Kim, *Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D Running of MIT Cheetah 2*, 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 5163-5170.
- [9] G. Bledt, M.J. Powell, B. Katz, J. Di Carlo, P. Wensing, and S. Kim, *MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot Cheetah 3*: Submitted to IROS, 2018

- [10] G. Bledt, P. M. Wensing and S. Kim, *Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah*, 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 4102-4109.
- [11] J. Di Carlo, P. M. Wensing, G. Bledt, B. Katz, S. Kim, *Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control* Submitted to IROS, 2018
- [12] Mevey, James Robert, *Sensorless field oriented control of brushless permanent magnet synchronous motors*, 2009 Masters Thesis, Kansas State University.
- [13] Kirtly, J.L. *6.061 Introduction to Power Systems, Class Notes Chapter 12, Permanent Magnet "Brushless DC" Motors*, Massachusetts Institute of Technology, 2003.
- [14] András Zentai, Tamás Dabóczi *Improving Motor Current Control Using Decoupling Technique* EUROCON 2005
- [15] H. Olsson, K.J. Åström, C. Canudas de Wit, M. Gäfvert, P. Lischinsky, Friction Models and Friction Compensation, European Journal of Control, Volume 4, Issue 3, 1998, Pages 176-195
- [16] E. Favre, L. Cardoletti and M. Jufer, "Permanent-magnet synchronous motors: a comprehensive approach to cogging torque suppression," in IEEE Transactions on Industry Applications, vol. 29, no. 6, pp. 1141-1149, Nov/Dec 1993.
- [17] R. R. Playter and M. H. Raibert, *Control Of A Biped Somersault In 3D* Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1992, pp. 582-589.
- [18] A. S. Huang, E. Olson and D. C. Moore, *LCM: Lightweight Communications and Marshalling*, 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 4057-4062.
- [19] J. Ramos, M. Chua, B. Katz, S. Kim, *Facilitating Model-Based Control through Software-Hardware Co-Design*, ICRA 2018, Brisbane, Australia.
- [20] G. Kenneally, A. De and D. E. Koditschek, "Design Principles for a Family of Direct-Drive Legged Robots," in IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 900-907, July 2016.
- [21] NXP Semiconductor Appnote AN11113, *LFPAK MOSFET thermal design guide - Part 2*, November 16, 2011
- [22] N. Hogan, *Impedance Control: An Approach to Manipulation*, 1984 American Control Conference, San Diego, CA, USA, 1984, pp. 304-313.

- [23] D. Lee and M. W. Spong, *Passive Bilateral Teleoperation With Constant Time Delay*, in IEEE Transactions on Robotics, vol. 22, no. 2, pp. 269-281, April 2006.
- [24] . Mattila, J. Koivumäki, D. G. Caldwell and C. Semini, *A Survey on Control of Hydraulic Robotic Manipulators With Projection to Future Trends*, in IEEE/ASME Transactions on Mechatronics, vol. 22, no. 2, pp. 669-680, April 2017.
- [25] William T. Townsend and Jeffrey A. Guertin, *Teleoperator slave –WAM design methodology*, Industrial Robot 26(3):167-177 · April 1999
- [26] Hogan, Neville, et al. *MIT-MANUS: a workstation for manual therapy and training*. Robot and Human Communication, 1992. Proceedings., IEEE International Workshop on. IEEE, 1992. 13, 22, 60, 158
- [27] Michele Focchi, Andrea del Prete, Ioannis Havoutis, Roy Featherstone, Darwin G. Caldwell, Claudio Semini, *High-slope terrain locomotion for torque-controlled quadruped robots*, Autonomous Robots, vol. 41, pp. 259–272, Jan 2017.
- [28] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, *CasADi – A software framework for nonlinear optimization and optimal control*, Mathematical Programming Computation, In Press, 2018.
- [29] A. Wachter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming, vol. 106, no. 1, pp. 25–57, Mar 2006
- [30] <http://hebirobotics.com/products/>
- [31] <http://www.up-board.org/up/>
- [32] <https://www.bostondynamics.com/>
- [33] <https://www.youtube.com/watch?v=fRj34o4hN4I>

Appendix A

Code and Design Files

- Motor controller hardware: https://github.com/bgkatz/3phase_integrated
- Motor controller firmware: https://os.mbed.com/users/benkatz/code/Hobbyking-Cheetah_Compact/
- SPIne hardware: <https://github.com/bgkatz/SPIne>
- SPIne firmware: <https://os.mbed.com/users/benkatz/code/SPIne/>
- Actuator data and analysis tools <https://github.com/bgkatz/actuator>

Appendix B

Videos

- Quadruped doing backflips: <https://youtu.be/BmYM000pF3A>
- Bilateral teleoperation: <https://youtu.be/FnQGPGG-vuQ>

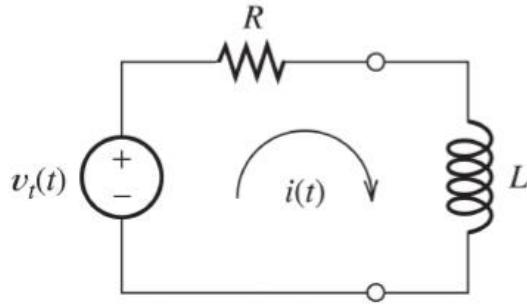
Appendix C

Discrete Time Current Control

C.1 System Model in Continuous and Discrete Time

The goal is to control current $i(t)$ with control effort $v(t)$ (often approximated by PWM).

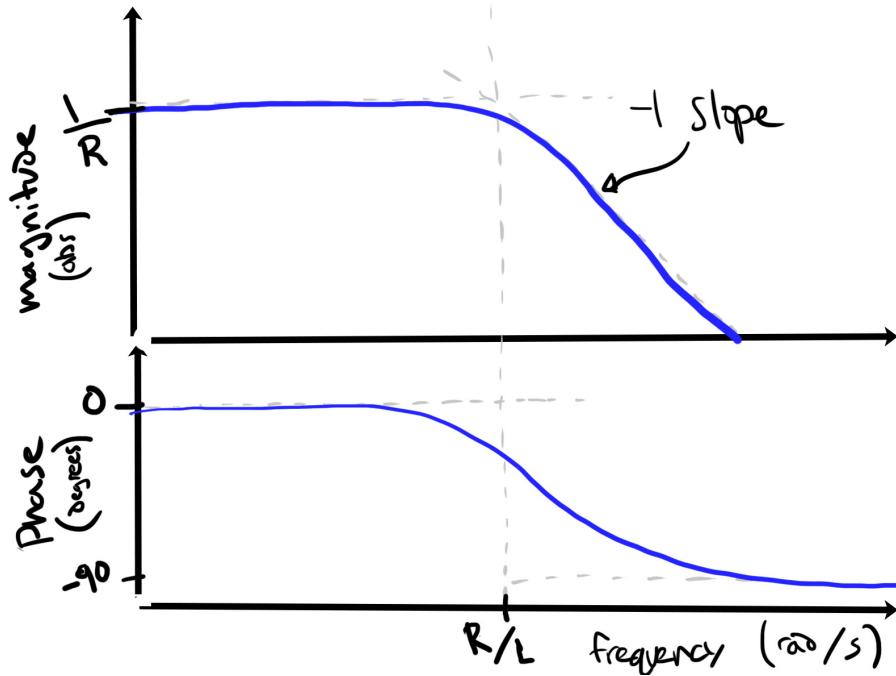
For electric motors, the electrical dynamics (the RL part) tend to be several orders of magnitude faster than the mechanical dynamics, so it is reasonable to ignore the mechanical dynamics, including back-emf, for the purpose of current control.



The continuous time frequency response, $\frac{I(s)}{V(s)}$ has the transfer function

$$\frac{I(s)}{V(s)} = \frac{1}{(Ls + R)} \quad (\text{C.1})$$

and the following bode plot. It is a first order low-pass filter, with a DC gain of resistance $\frac{1}{R}$, and a cutoff frequency of $\frac{R}{L}$:



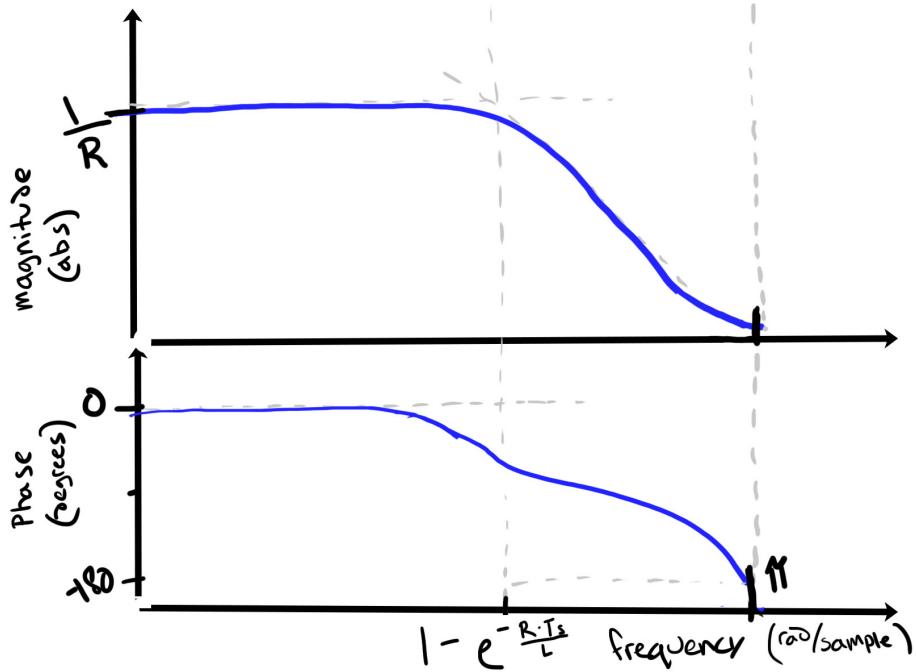
Since control is going to be performed in discrete time, it makes sense to convert the continuous time model above to discrete time, and then design the controller directly in discrete time. The conversion is done by taking the zero order hold equivalent of the circuit transfer function, to map it to the Z-domain. Through some algebra, or with the help of MATLAB, this can be shown to be:

$$\frac{I(z)}{V(z)} = \frac{\frac{1}{R} \left(1 - e^{-\frac{RT_s}{L}}\right)}{z - e^{-\frac{RT_s}{L}}} \quad (\text{C.2})$$

Where T_s is the sample time, in seconds (e.g. 0.00005 for a 20 kHz loop frequency)
The MATLAB syntax for this conversion is:

```
s = tf('s');
sys = 1/(L*s + R);
sys_discrete = c2d(sys, Ts)
```

In discrete time, the system has the following bode plot. It looks very similar to the continuous time one, except that phase drops to -180 at π radians per sample:



C.2 Controller Design

A discrete time PI controller written in the form

$$\frac{U(z)}{E(z)} = k \left(1 + \frac{ki}{z-1} \right)$$

, where U is control effort, and E is error, has a pole at $z = 1$, a zero at $z = 1 - ki$, and a high-frequency gain of k . This way, changing k only changes the loop gain, and changing ki only changes the zero location, unlike the typical parallel PI expression.

If we set the zero of the controller near the pole of the RL system, then the loop return ratio just looks like an integrator. This can be done by setting

$$ki = 1 - e^{-\frac{RT_s}{L}} \quad (\text{C.3})$$

as is apparent from the bode plot above.

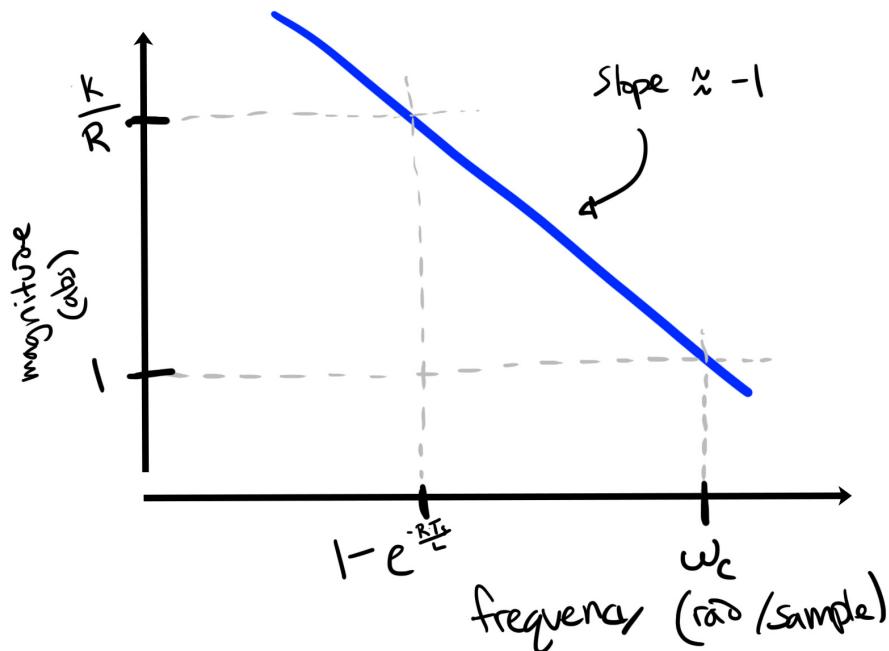
An approximate continuous time version of this is setting

$$ki = \frac{RT_s}{L} \quad (\text{C.4})$$

This calculation will be very close to the discrete time version if the RL time constant $\frac{L}{R}$ is much larger than T_s (around 10x or more), but become less and less accurate as the electrical time constant gets close to T_s .

Now the only parameter to choose is the loop gain. Loop gain should be chosen by balancing the desired crossover frequency of the current loop with effects like current sensor noise, which will be amplified by high loop gain. It should be quite reasonable to achieve a crossover frequency around $\frac{\pi}{10}$ or so radians per sample, or even faster. With a 20 kHz loop, that corresponds to 1 kHz crossover frequency.

The loop gain required for a given crossover frequency can be approximated by looking at the magnitude of the bode plot of the return ratio. In this case, the return ratio is the RL circuit in series with the PI controller. With the choice of k_i presented above, this just looks like an integrator:

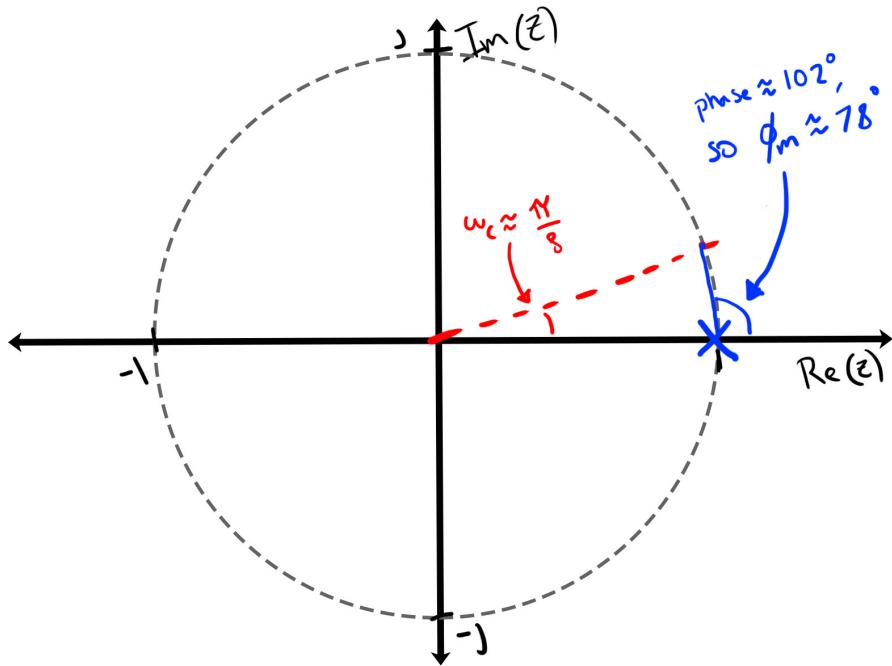


The loop gain k required to achieve the desired crossover frequency ω_c can be easily calculated using known points along the bode plot. A crossover frequency of ω_c means that the magnitude of the return ratio crosses 1 (0 dB) at frequency ω_c . Also, we know that at the location of the RL pole and PI controller zero, the

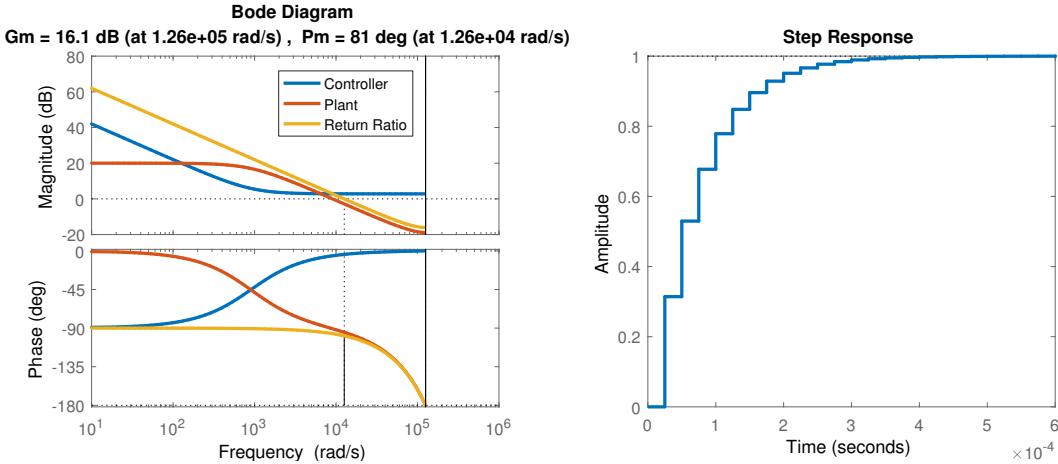
integrator has a gain of $\frac{k}{R}$. So the necessary k can be computed as:

$$k = R \left(\frac{\omega_c}{1 - e^{-\frac{RT_s}{L}}} \right) \quad (\text{C.5})$$

The resulting phase margin can be calculated by looking at the pole-zero map of the return ratio (i.e. an integrator, which is a pole at $z = 1$). The crossover point is the intersection of the unit circle and the line through the origin at angle w_c . Phase at that point is the angle from the pole at $z = 1$ to the intersection, and phase margin is 180 minus that angle. So, for example, with $w_c = \frac{\pi}{2}$, phase margin should be 45° , or $w_c = \frac{\pi}{8}$, $\phi_m \approx 78^\circ$



Here are an example bode plot and step response for a controller design this way:



C.3 Implementation in Code

Here is bare-bones implementation of the current loop in C syntax. Also, as an important implementation note, gain calculations assume the output of the controller is in Volts, and the current measurement is in Amps. If you need to set a PWM duty cycle, for example, loop gain k must be scaled appropriately. Also, the value of the integral should be limited, so it can't "wind up". A reasonable limit might be the maximum possible control effort - for example, if you are using a 12-Volt power supply, cap the value of the integral to ± 12 volts.

```
*****
current = sample_current()
current_error = current_ref - current;
integral += k*ki*current_error;
//Saturate the integrator to prevent windup
integral = fmaxf(fminf(integral, integral_max), -integral_max);
voltage = k*current_error + integral
//Saturate controller output to be within valid range.
voltage = fmaxf(fminf(voltage, voltage_max), -voltage_max);
set_voltage();
*****
```

C.4 MATLAB Gain Calculating Script

Insert the appropriate values for R, L, and Ts.

```
clear()

%%% System Parameters %%%
R = .5;           % Resistance in Ohms
L = 1e-3;         % Inductance in Henries
Ts = .00005;       % Sample period
wc = pi/10;        % Crossover frequency, in Radians per sample
%%%

s = tf('s');
sys = 1/(L*s + R); % Continuous time transfer function
z = tf('z', Ts);
sys_d = c2d(sys, Ts);    % Zero order hold equivalent

ki = 1-exp(-R*Ts/L)      % Calculate Ki
k = R*((wc)/(1-exp(-R*Ts/L))) % Calculate loop gain

controller = k*(1 + ki/((z-1)));    % PI controller transfer function
fp = series(controller, sys_d);      % Forward Path
cl = feedback(fp, 1);                % Unity feedback

%%% Plot open-loop, return ratio, and controller bode plots %%%
figure;
hold all
bode(controller);
bode(sys_d);
```

```
margin(fp);  
legend('Controller', 'Plant', 'Return Ratio')  
  
%%% Plot closed-loop step response %%%  
figure;step(cl);
```