Title
Design, Dynamics, and Control of Mobile Robotic Systems

Author
Morozovsky, Nicholas

Publication Date
2014

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Design, Dynamics, and Control of Mobile Robotic Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Nicholas Morozovsky

Committee in charge:

Professor Thomas Bewley, Chair
Professor Maurício de Oliveira
Professor Yoav Freund
Professor Frank Talke
Professor Mohan Trivedi

2014

The dissertation of Nicholas Morozovsky is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2014

EPIGRAPH

Never underestimate the
estimation problem.
—Unknown

TABLE OF CONTENTS

LIST OF SUPPLEMENTAL FILES

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to the senior PhD students in the Coordinated Robotics Lab, Christopher Schmidt-Wetekam and Andrew Cavender for their experience, help, and camaraderie. Thanks to my advisor, Prof. Thomas Bewley. Thanks also to the coauthors of my papers, Christopher Schmidt-Wetekam, Robert Moroto, and Prof. Thomas Bewley.

Thanks to Matt Bockman, Ishmael Rahimian, Travis Rowell, Ronald Truong, Yoshio Tsuruta for their assistance in building a fleet of 13 Switchblade robots.

Thanks to Monique Goeders, Michael Kaplan, Farzad Kharazi, Robert Moroto, and Joshua Munoz for their help in the design and fabrication of the first dyno prototype and Derrick Cheng and Dylan Drotman for their work on the second dyno prototype.

Thanks to Victor Ruiz, Amber Frauhiger, Alexandros Kasfikis, and Antonios Kountouris for their assistance in developing the SkySweeper prototype.

Thanks to National Instruments for their hardware and software donations, as well as their financial support, for the Switchblade, DC motor dynamometer, and Switchbot projects.

Thanks to WowWee for their generous support of the Switchblade nano and Switchbot projects.

Thanks to Brain Corporation for funding the eyeFling project and donating the bStem controller.

Thanks to Maxon motor ag for generous equipment donations that have powered the Switchblade nano and perching variants, SkySweeper, and provided a set of high tolerance motors to validate the DC motor dynamometer project.

Thanks to moviMED for donating image capture cards for the Switchblade project.

Parts of chapters 1, 2, 3, 4, 5, and 6 concerning Switchblade have appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Parts of chapters 1, 2, 3, 5, and 6 concerning SkySweeper have appeared in print: N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2339-2344, 2013. The dissertation author was the primary investigator and author of this paper.

Parts of chapters 1, 2, 4, 5, and 6 concerning the DC motor dynamometer have appeared in print: N. Morozovsky, R. Moroto, T. Bewley, "RAPID: an inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013. The dissertation author was the primary investigator and author of this paper.

Parts of chapters 1, 2, 3, 4, 5, and 6 concerning the perching maneuver have been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

VITA

2007	B. S. in Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA
2010-2012	Teaching Assistant, University of California, San Diego
2010-2014	Research Assistant, University of California, San Diego
2011	M. S. in Engineering Sciences (Mechanical Engineering), University of California, San Diego
2014	Ph. D. in Engineering Sciences (Mechanical Engineering), University of California, San Diego

PUBLICATIONS

N. Morozovsky, T. Bewley, "Stair Climbing via Successive Perching," *IEEE/ASME Trans. Mechatronics*, submitted Oct. 2014.

N. Morozovsky, R. Moroto, T. Bewley, "RAPID: an inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013.

N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2339-2344, 2013.

N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011.

C. Schmidt-Wetekam, N. Morozovsky, T. Bewley, "Inertial Rotation Center Position Estimation for a Perching Treaded Vehicle," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 4097-4102, 2011.

T. Bewley, C. Colburn, D. Zhang, J. Cessna, N. Morozovsky, A. Cavender, and C. Schmidt-Wetekam. "Real-time turbulent plume estimation with mobile sensors," *APS Division of Fluid Dynamics Meeting Abstracts*, vol. 1, 2010.

ABSTRACT OF THE DISSERTATION

Design, Dynamics, and Control of Mobile Robotic Systems

by

Nicholas Morozovsky

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2014

Professor Thomas Bewley, Chair

Mobile robotic systems are increasingly prevalent in several fields of academia, industry, and everyday life. This dissertation outlines the development of several means of locomotion for such systems and the tools created and used to design them. Novel mechanical designs with minimal actuators are combined with advanced control systems to create dynamic locomotion behavior. A single, unified Lagrangian dynamics architecture is used to represent a variety of robotic systems, derived from first principles and applied programmatically. Model-based control methods are heavily used to take advantage of the known dynamics and are also applied programmatically for the automatic generation of control laws and gain scheduling lookup tables. Improvements to encoder velocity estimation across a wide velocity range are described in detail. A number of key hardware

advances in different fields are exploited to create low-cost, but capable, mobile robotic systems: additive manufacturing (3D printing), powerful embedded microprocessors (Arduino, ARM, FPGA), solid state sensors (MEMS accelerometers and gyroscopes), and inexpensive brushed DC motors. This dissertation presents both notable results and tactical details of implementation that will be useful to those designing similar systems.

Several novel robotic systems are presented in this dissertation. First, two incarnations of a mobile balancing platform with two coaxial wheels with the center of mass above the center of rotation are presented. Active feedback control is required to stabilize the system. Next, Switchblade is a patent pending treaded inverted pendulum designed for maximum mobility over a wide range of terrain including tight spaces and significant obstacles (e.g. stairs). Multiple variants of the design are explored, which shows that the vehicle concept is broadly applicable across different length scales and use cases. SkySweeper is an under-actuated robotic system designed for multimodal locomotion along wires and cables. Finally, RAPID is a reconfigurable, automated dynamometer for characterizing small DC motors, the primary purpose of which is identifying the operating parameters of motors used in vehicles with model-based control and estimation algorithms.

Chapter 1

Introduction

Robots and robotic systems have the potential to impact many socially relevant applications from security and search and rescue to health care and recreation. Locomotion is a key challenge in enabling robots to operate in environments less structured than the laboratory (perception and manipulation are two other key challenges). A number of novel locomotion systems and techniques are presented in this dissertation, along with the tactical details of implementation, which would prove useful to others undertaking related research. Another major impediment to the broad adoption of robotic systems is the generally high cost of such systems. The approaches and solutions developed in this dissertation are not dependent on expensive components such as precision actuators or motion capture systems. The primary contributions of this dissertation are:

- Mechanical designs of multiple versions of a treaded inverted pendulum, utilizing appropriate materials and components at different length scales (Sec. 5.5).

- Control system designs to dynamically stabilize unstable equilibrium manifolds of treaded inverted pendula despite significant friction in the system (Sec. 3.2).

- Documentation of a broadly applicable method of programmatically deriving equations of motion and control systems for robotic systems (Sec. 2.2 and App. B).

- Algorithms to accurately estimate velocity, over a wide range including very low speed, from an imperfect quadrature encoder (Sec. 4.2).
- Design of a low degree of freedom cable-locomoting robot capable of surpassing obstructions on the cable (Sec. 1.4.4 and Sec. 5.6).
- Finite state machine controllers for multiple maneuvers of the above cable-locomoting robot (Sec. 3.3).
- Mechanical design of a reconfigurable motor dynamometer which can easily be manufactured with a 3D printer and optional laser cutter (Sec. 5.3).
- Algorithms and documentation to automatically identify parameters of DC motors with the above dynamometer system (Sec. 4.5).
- Mechanical design of two mobile inverted pendula which can easily be manufactured with a 3D printer and optional laser cutter (Sec. 5.4).

This dissertation is structured into the following chapters:

1. Introduction: summarizes the robotic systems presented and describes the motivation and background for the research presented, then the architecture and maneuvers of the specific robotic systems are further explained.
2. Dynamics: outlines the method of Lagrangian dynamics used to model the presented robotic systems, also details the motor model used for brushed DC motors.
3. Control: the control system design, including linearization, feedforward control, friction compensation, discretization, integral control, and trajectory planning.
4. Estimation: explains the algorithms used to estimate parameters and states of the robotic systems using low-cost sensors, such as inertial MEMS sensors and encoders.
5. Prototypes: describes the development of the physical prototypes and compares their experimental performance to simulation results.

6. Conclusions: summarizes the contributions made in this dissertation and suggests possible future directions.

DC Motor Dynamometer

This dissertation, in part, describes the development of an automated dynamometer to characterize brushed direct current (DC) motors. The unique mechanical design allows the testing of a wide range of motor sizes. The motor under test is subjected to a given pulse width modulated (PWM) voltage signal and position, current, and voltage measurements are simultaneously recorded from the integrated sensor suite. An electromechanical motor model is developed by combining the voltage and torque balance equations of the system. A least-squares algorithm is used to estimate the parameters that best fit the observed data to the specified gray-box model. The system retains a low cost by using off-the-shelf electronics and cheaply fabricated mechanical parts. The inertia and friction of the system are carefully modeled, removing the need for an expensive torque sensor. The mechanical drawings, electrical schematics, and software are open source and freely available for download. Consistent parameter estimates from a set of high-tolerance and well-documented identical motors demonstrate the accuracy and precision of the system.

Switchblade

A versatile unmanned ground vehicle (UGV) should be able to traverse rough terrain while retaining a small form factor for navigating confined spaces. Such a (patent pending) vehicle, dubbed Switchblade, is developed in the present work via an effective combination of a novel transforming mechanical design, capable onboard electronics, and advanced feedback control algorithms. A single chassis holds the actuators, sensors, electronics, and battery. Shafts protruding from either side of this chassis connect to tread assemblies. Rotation of this shaft causes the treads to advance for translational movement; rotation about this shaft causes the entire tread assembly to rotate with respect to the chassis. Vehicle orientation is estimated via onboard filtering of optical encoders and MEMS accelerometers and gyroscopes. In its horizontal configuration, Switchblade operates as a differential-drive treaded platform. In its various upright configurations, Switchblade operates as a mobile inverted pendulum, capable of surmounting obstacles, including stairs, that would otherwise be impassable by a vehicle of its size. Design-for-manufacturing (DFM) and design-for-assembly (DFA) techniques are employed to reduce cost, part count, complexity, and assembly time without sacrificing system capabilities. Results from a working prototype are discussed. The resulting platform is well suited for a variety of socially relevant applications, including reconnaissance, mine exploration, and search & rescue.

Stairs are a primary challenge for mobile robots navigating indoor, human environments. Stair climbing is a useful, if not necessary, capability for mobile robots in urban search & rescue, security, cleaning, telepresence, elder care, and other applications. Existing stair climbing robots are large, expensive, and not always reliable, especially when descending stairs. In this paper, we present a novel approach for stair climbing that is achievable by a small mobile robot with minimal actuators and sensors, and thus cost. The proposed robot has articulated tread assemblies on either side of a chassis. Using feedback control, the robot can balance on the edge of a single step. As the robot drives up the step, the chassis pivots to maintain the center of mass directly above the contact point. The dynamics of the system are derived with the Lagrangian method and a discrete-time integral controller with friction compensation is designed to stabilize a stair climbing trajectory. The algorithms used to estimate the state of the system with low-cost, noisy, internal sensors are explained in detail. No external motion capture system is used. Simulation results are compared to successful experimental results.

SkySweeper

SkySweeper is a mobile robot designed to operate in an environment of cables, wires, power lines, ropes, et cetera. The robot is comprised of two links pivotally connected at one end; a series elastic actuator at this "elbow" joint can actuate relative rotation between the two links. The SEA enables the robot to store additional spring potential energy before commencing a dynamic maneuver.

At the opposite end of each link is an actuated three-position clamp. The clamp can either be open, partially closed, such that the clamp can roll (translate) along the cable, or fully closed, such that the clamp can only pivot on the cable. By actuating the elbow joint and cleverly choosing the positions of the clamps, the robot can locomote on the cable in a number of different ways. The particular method of locomotion can be chosen to minimize energy consumption, maximize speed, or traverse an obstacle (e.g. a support from which the cable is suspended). The robot includes sensors (including rotary potentiometers and photodiodes) to determine the relative angle between the two links, the amount of spring deflection in the SEA, and whether or not a cable is within the grasp of each clamp. SkySweeper has the potential to locomote in a more energy efficient manner than existing cable-locomoting robots. It also operates with a minimal number of actuators, which reduces cost significantly. Potential applications include power and communication line inspection, suspension bridge inspection and construction, as well as entertainment. Data from a prototype, consisting largely of 3D-printed and off-the-shelf parts, are compared to dynamic simulation results.

1.1 Motivation

1.1.1 DC Motor Dynamometer

Due to their low cost, wide availability, and simple control implementation, brushed DC motors are desirable actuators for many robotics applications [1], [2]. However, the design process should accurately account for the role of a motor's dynamics. Motor dynamics may potentially be omitted from the system model, but excluding such dynamics can have significant unintended consequences. Robotic systems can experience performance degradation and loss of stability when actuator dynamics are ignored [3]. A potential approach is to assume that the manufacturer's specifications for a given motor are accurate. However, variations in the quality of documentation between manufacturers, often determined from unknown testing procedures, can result in motors whose dynamics are not reflected fully or accurately in the manufacturer's specifications. A motor's specifications may only contain values for one operating point; it may be necessary to operate the motor at, e.g., a different voltage. Moreover, motor specifications do not characterize the motor driver circuit, which operates in conjunction with the motor in actual implementation. Thus, in order to rigorously obtain a reliable actuator model, a given brushed DC motor and motor driver pair can be subjected to empirical testing and subsequent parameter identification.

1.1.2 Mobile and Treaded Inverted Pendula

A number of applications motivate small, simple UGVs that can robustly overcome complex terrain challenges, while also being able to navigate in confined spaces; such applications include patrol, search & rescue, mine exploration, and the disposal of improvised explosive devices (IEDs). In such applications, it is generally advantageous for the vehicles used to be inexpensive, so that multiple vehicles may be deployed to accomplish a given mission, and the loss of some is acceptable. The cost of a UGV may be reduced by minimizing its size and mechanical complexity, noting that advanced feedback control algorithms, once designed, may generally be implemented at low cost. In order for robots to be accepted and useful in indoor, human environments, they must be able to locomote unassisted. Three primary locomotion challenges, beyond stationary and moving obstacle avoidance, are stairs, doors, and thresholds (a degenerate case of stairs). In this dissertation, we focus on the stair climbing problem.

1.1.3 Cable Locomotion

Power lines, communication lines, hanging pipe, taut rope, and the like present an interesting environment for robotic systems to traverse. High wires, such as power lines (which may also be live at high voltage), are a dangerous environment in which using robots can improve human safety. Repetitive tasks such as inspection and monitoring naturally lend themselves to automation. Existing systems, which will be discussed, have many degrees of freedom and actuators, which tend to increase system complexity and cost. Application areas in this type of environment include power line inspection and maintenance, communication line inspection, maintenance, and surveillance, suspension bridge inspection, maintenance, and construction, as well as entertainment and toys.

1.2 Background

1.2.1 Dynamometers

Research in the field of dynamometers has been diverse in terms of both the actuators under test and the loading conditions. A small-scale dynamometer has been developed to characterize propeller blades for unmanned air vehicles (UAVs) [4]. This system uses six thin beam strain gauges to separately measure thrust and torque. The dynamometer developed in [5] has an active braking system designed to simulate different nonlinear loading conditions for the purpose of testing nonlinear control algorithms. A strain gauge measures the reaction torque on the braking assembly. The system was not designed for characterizing different motors. The test setup in [6] that was used to characterize a brushed DC motor with gearbox included a braking system with a rope pulling on a force meter. The sensors were not automatically measured which limited the tests that could be performed on the motor. A dynamometer for miniature piezoelectric actuators was built in [7]. Custom optical sensors measured deflection of the actuator. Multiple actuators were tested under different operating conditions, but the system is specific to small, bending type actuators. Dynamometers are also used to measure cutting force in machining operations [8]. Strain gauges and piezoelectric sensors are commonly used.

Many of the above dynamometers (and several others not mentioned) use strain gauges to measure force or torque. Strain gauges require precise calibration and specialized measurement electronics. They must also be carefully mounted in the system to prevent off-axis forces and torques from perturbing the desired measurement. Strain gauges significantly increase the cost and complexity of a dynamometer.

1.2.2 Mobile Inverted Pendula

Inverted pendula are often used in controls labs as a fundamental teaching tool. In recent years, mobile inverted pendula have become increasingly popular, including the Segway Personal Transporter [9]. The Segway is perhaps the only mobile inverted pendulum to have yet ventured outside of the sheltered lab environment on a large scale, and is itself largely operated on flat sidewalks. We now review existing mobile inverted pendula. The most common paradigm today is a two-wheeled platform ([10], [11]), with steering accomplished by differential drive. Another class of mobile inverted pendula uses a single ball instead of two wheels, thereby achieving holonomic locomotion ([12], [13]). Both designs have a fundamental weakness in common: the maximum obstacle size that such a vehicle can overcome is limited by the diameter of its wheels or ball.

Legged robots often use a linear inverted pendulum model and calculations of the zero moment point to maintain balance while stationary or moving [14]. Such robots have a great deal of flexibility when overcoming obstacles: they may step over or onto an obstacle [15], or even hop over an obstacle [2]; however, they are also mechanically complex, with many actuators and possible failure points.

1.2.3 Treaded Mobility Platforms

The standard treaded platform (manned versions of which were developed by the British in WW1, and smaller unmanned versions of which were developed by the Germans in WW2) consists of two fixed tread mechanisms mounted on opposite sides of a central chassis. This type of UGV performs well over a variety of both smooth and rough terrain (including loose dirt and gravel, sand, snow, mud, etc.), but cannot generally overcome obstacles larger than the radius of its tread sprocket. This limitation may be extended by adding additional idler sprockets to increase the height of the tread assembly (such as the trapezoidal shape of the treads of an M1 Abrams tank), or by adding secondary articulated treaded segments or “flippers” (e.g., the iRobot PackBot [16]). Additional treaded segments may be added to increase a treaded vehicle’s agility [17], but at the expense of significantly increased cost, complexity, and possible failure points; for instance, a serpentine robot may consist entirely of treaded segments [18]. Another notable treaded platform is the Vecna Robotics Battlefield Extraction-Assist Robot (BEAR; see [19]), which has two-segment tread assemblies pivotally attached to either side of a central torso which also has two manipulator arms. The BEAR can operate with its articulated tread assemblies in a number of different configurations, including dynamically balancing on either end of the tread assemblies with inverted pendulum control. Note that the BEAR is a large, complex vehicle, with tread segments large enough to overcome many common obstacles (stairs, medium-sized rubble, etc.) without utilizing balancing behavior.

1.2.4 Stair Climbing

A number of existing robots are capable of stair climbing, which can generally be classified into a few categories. These include humanoids, such as those featured in the DARPA Robotics Challenge [20] [21] [22]. Despite recent advances prompted by the challenge, it is not necessary, and indeed it is complex and costly, to locomote in a human manner in a human environment. Alternative form factors and control strategies can be simpler, cheaper, and faster. Traditional treaded vehicles [16] [23] that are long enough to span multiple step edges can climb stairs in a straightforward manner. Additional, articulated tread segments can aid in agility, particularly on the first and last steps [24] [25]. Another class of stair climbing robots utilize hybrid wheel-leg, or wheg, systems. This includes the popular RHex hexapod [26] and a number of robots with alternate wheg designs [27] [28] [29]. A unique design utilizes deformable wheels which can transform into treads to fit into tight places as well as pivot like whegs to overcome steps, but the prototype presented is still limited to small stairs [30]. Other robots employ a dedicated mechanism for stair climbing, such as hopping [31] [2] [32] or a system for raising and lowering the robot in two or more segments [33] [34] [35]. While these methods can be effective at climbing stairs, the weight and size of the dedicated stair climbing components detract from the robot's performance on flat ground and adds to cost. There has been recent research into stair climbing robots with mobile inverted pendulum dynamics [36] [37], but both are wheeled systems with wheel radii greater than the step height. Another wheeled inverted pendulum robot is smaller and uses a novel mechanism to climb its own central post, but requires space on each step to reposition itself and is slow [38]. A number of stair climbing robots are not capable of climbing standard stairs, that is, they can only climb stairs with an unrealistically low pitch angle, and/or rise. Some of the vehicles would be difficult, costly, and inefficient to scale to the size of standard stairs.

1.2.5 Cable Locomotion

We now review existing robots designed to locomote on cables. The application of power line inspection has been the largest motivator of cable-locomoting robotics research. An extensive survey paper was published in 2009 [39], which the reader is encouraged to review. A few key examples are discussed here. Expliner is from the Japanese company HiBot which is closely affiliated with the Tokyo Institute of Technology. It can roll on one or two cables and circumvent multiple types of obstacles by shifting its center of mass and lifting one of two pulley arms, lowering it on the other side of the obstacle, shifting its center of mass under the second arm, and lifting the first arm [40]. LineScout was developed at Hydro-Québec IREQ directly for field use, it has a sliding mechanism with a redundant pair of clamps that are only used for overcoming obstacles [41]. The dual-arm robot presented in [42] has three linear actuators, one in each arm and one in the chassis. Rotary actuators and powered clamps allow the robot to release one arm and pivot around obstacles as large as the robot. Cable Crawler, developed by Bühringer et al. at ETH Zürich, has large enough vertical and horizontal rollers to be able to passively roll over certain types of obstacles [43].

All the above mentioned robots perform quasi-static maneuvers with many degrees of freedom and many actuators. The systems are necessarily large, complex, and expensive. This leaves the field open to be disrupted with a mechanically simple design with few degrees of freedom, but agile, dynamic maneuvers.

1.3 Approach

In all of the robotic systems presented here, a common design approach is taken. Low-cost mechanical and electrical off-the-shelf components are utilized wherever possible (MEMS sensors, optical encoders, brushed DC motors) supplementing with custom printed circuit boards (PCBs) to simplify wiring. Rapid prototyping techniques (3D printing, laser cutting) are used for custom parts with a fast design-build-test iteration cycle. The design and fabrication of the prototypes is discussed in detail in Ch. 5. A list of “Rules of Robotics” may be found in Appendix A.

1.4 Robotic Systems

1.4.1 DC Motor Dynamometer

This dissertation, in part, presents the development of a dynamometer for brushed DC motors and motor drivers which we call the Reconfigurable Automated Parameter-Identifying Dynamometer (RAPID). RAPID is capable of simultaneously controlling a motor and recording sensor data that are post-processed to determine the electrical and mechanical parameters of the motor/motor driver system. The entire process is automated, requiring minimal user interaction. RAPID, in Fig. 1.1, is equipped with a suite of sensors to measure rotational position, current, and voltage. Furthermore, RAPID can accommodate a variety of motor geometries and specifications with its unique hardware design.

The intended application for RAPID is to characterize motors for model-based control systems in robotics applications. RAPID consists of off-the-shelf parts, custom parts that may be readily fabricated, and open source algorithms. As such, devices similar or identical to the one discussed in this work may easily be fabricated for use in many different settings, such as laboratory classes, academic research, and industrial fabrication and quality assurance. RAPID is open source: mechanical drawings and software may be freely downloaded at http://robotics.ucsd.edu/dyno or accessed in Supplemental File 5: DC motor dy-

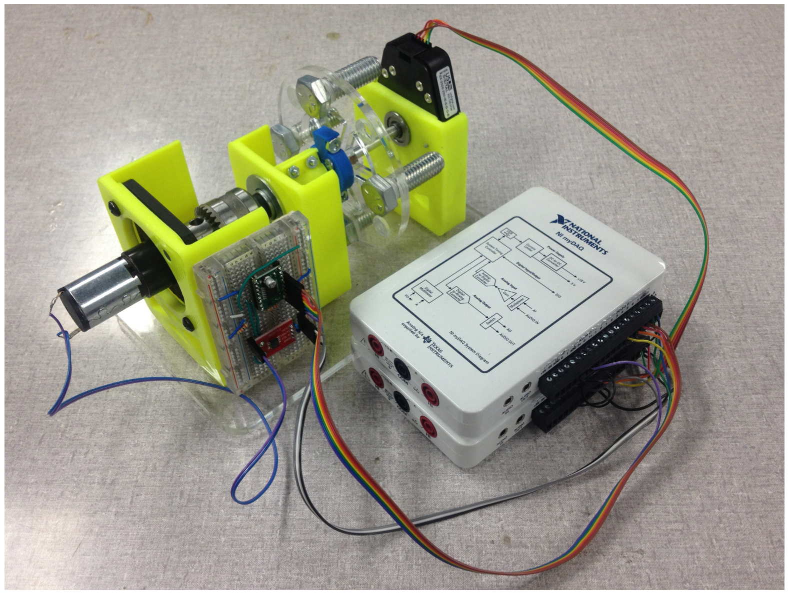


Figure 1.1: Dynamometer system.

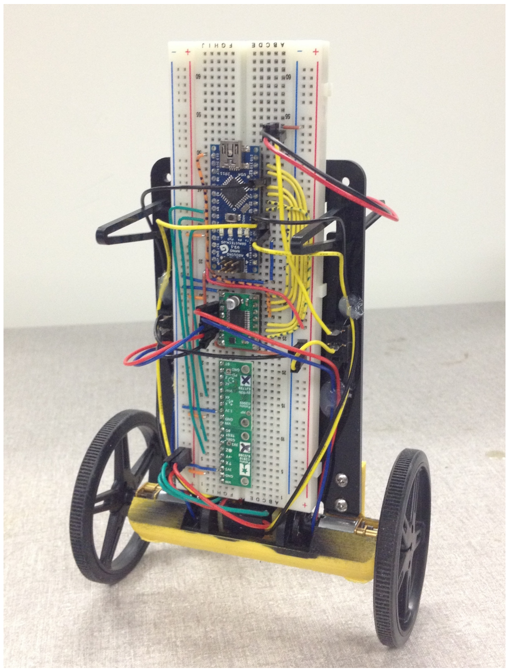
namometer files.

1.4.2 Mobile Inverted Pendulum

Two incarnations of the prototypical two-wheeled mobile inverted pendulum (i.e. "Segway" like vehicles [9], [10], [11]) have been developed. Though they have different components (motors, sensors, microprocessors, etc.) and parameters (mass, length, inertia, etc.), they share an identical dynamics formulation and control design process.

myMIP

The first, myMIP, is an educational platform which was created to serve as the lab component of the digital control systems class (MAE 143C) taught at the senior/masters crossover level in Fall 2012, see Fig. 1.2a. This design took knowledge gained from several generations of inverted pendulum robots to create the most affordable inverted pendulum possible from primarily off-the-shelf parts. The development schedule was rushed to get the kits to the students as fast as possible. It took only one month from drafting an initial Bill of Materials to passing



(a) myMIP system.

(b) eyeFling system.

Figure 1.2: Two mobile inverted pendulum systems.

out 45 kits. The bag of parts with electronics, motors, and mechanical parts cost \$128 each. By the end of the quarter, each student had a mobile working inverted pendulum robot.

eyeFling

The second, eyeFling, was developed for Brain Corporation as a test platform for their low-power bStem single board computer with the Qualcomm Snapdragon processor and neuromorphic computing algorithms, Fig. 1.2b. This robot is an updated version of the iFling robot, previously developed by Ben Sams in the lab. The major modifications were the addition of two pan/tilt cameras, mounting the larger bStem electronics, and optimization for fabrication with a low-cost, single extruder fused deposition modeling (FDM) 3D printer. Following the successful adoption of the first three eyeFling prototypes by research scientists at

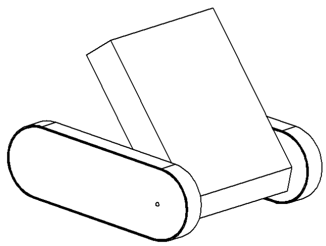
Brain Corporation, a further seventeen eyeFling prototypes were assembled.

1.4.3 Switchblade

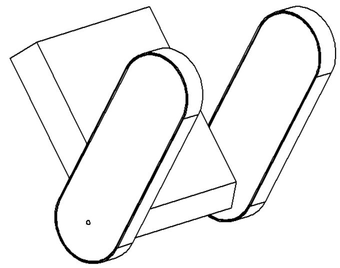
Figure 1.3: Completed Switchblade prototype.

As in a traditional treaded vehicle, Switchblade has a pair of tread assemblies, driven by an internal sprocket, mounted on either side of a central chassis (Fig. 1.4a). Uniquely, the tread assemblies can rotate continuously about the main drive axle of the chassis. Changing the angle between the chassis and tread assemblies moves the center of mass. There are no physical connections between the two tread assemblies to keep them parallel, but feedback control may be applied when it is desired to keep the two tread assemblies in line.

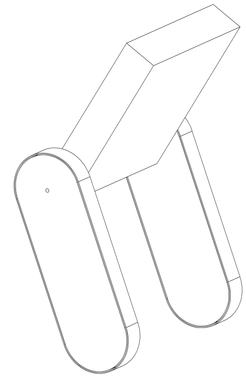
In a horizontal configuration (Fig. 1.4a) the robot functions much like any other treaded skid-steer robot, with the ability to independently drive each tread forward or backward to drive and turn. The treads act to minimize contact force on loose surfaces and maintain traction better than wheels. Note that the actuated tread assemblies make the robot impervious to high-centering. Note also that the robot operates just as easily “upside down” as “right side up.” Driving over



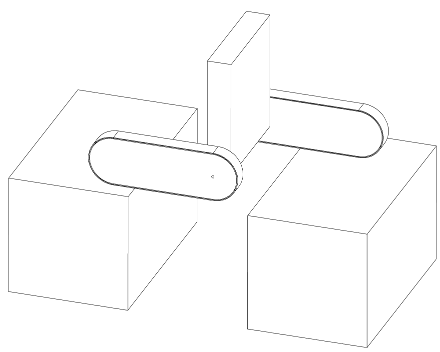
(a) Horizontal



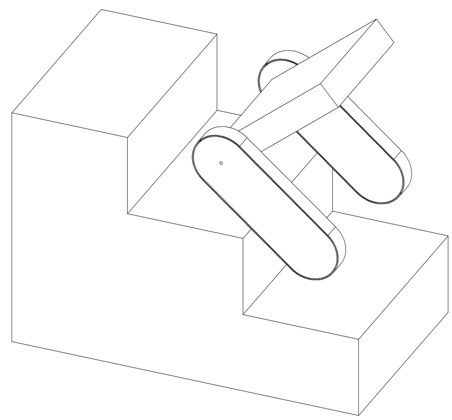
(b) V-balancing mode



(c) C-balancing mode



(d) Chasm crossing



(e) Perching on an edge

Figure 1.4: Different operating modes of the Switchblade design.

rough terrain may induce unwanted vibration in the chassis, this vibration can be reduced by pivoting the chassis in response to a disturbance. Given the nature of the coupled system, the greatest disturbance rejection is realizable at the end of the chassis farthest from the pivoting axis. One application of such an active suspension system is image stabilization for a camera mounted in or on the chassis.

The robot may balance on either end of the treads by taking advantage of the tread transferring torque to the idler sprocket. In a wheeled design, the idler wheel would be passive unless a second motor was driving it or if a chain or belt connected the front and rear wheels. When the robot is balancing on the sprockets coaxial with the pivoting axis, a side view of the robot resembles the letter “V”

(Fig. 1.4b), this maneuver is referred to as V-balancing mode. Alternatively, the robot may balance on the distal sprockets of the treads, a side view of which resembles a crude version of the letter "C" (Fig. 1.4c), this is called C-balancing mode. By driving the treads, rotating the angle of the tread assemblies, or actuating both simultaneously, the robot is able to maintain its balance.

Unlike the canonical inverted pendulum, the unstable equilibrium point (the angle of the chassis with respect to gravity at which the center of mass is directly over the contact point with the ground) is not constant and instead depends on the angle of the tread assemblies with respect to the chassis, and their relative mass distributions. This angle can be calculated with the known properties of the vehicle. There are multiple maneuvers, some dynamic and some quasi-static, for transitioning between the horizontal configuration and the upright configurations. One such maneuver for uprighting into V-balancing mode will be discussed in section 3.2.1.

The maximum height of the robot in V-balancing mode is the length of the treads with the center of mass approximately central, whereas in C-balancing mode, the maximum height is the length of the chassis plus the length of the tread assemblies minus the radius of the sprocket, with the center of mass above the treads. The added height in C-balancing mode allows the robot to stand up taller, climb larger obstacles, and to see farther with the onboard camera. The ground clearance of the chassis can be nearly the length of the tread assemblies, allowing the robot to pass over minor obstacles.

The separation angle between the chassis and the tread assemblies is hereafter referred to as the V-angle. The maximum V-angle achievable in V-balancing mode is dependent on the mass properties of the tread assemblies and chassis and is less than or equal to 180°. In the present design, the tread assemblies are longer than the chassis, such that the chassis may rotate continuously and pass through the tread assemblies while in C-balancing mode.

The maximum obstacle size that the robot can overcome is related to its overall height. When the robot is horizontal, its height is the diameter of the tread sprocket; when the robot is upright, its height is related to the tread length. The total height of the robot is thus variable over a large range, and can be adjusted as necessary: the robot can upright itself to overcome large obstacles, then lay itself back down to, e.g., pass freely underneath parked cars.

To overcome an obstacle larger than the sprocket radius, the robot approaches the obstacle while balancing upright, "leans" onto the obstacle by shifting its center of mass over the point of contact with the obstacle, then drives over the obstacle. An alternative maneuver to climb stairs uses the chassis of the robot as a lever. The robot approaches the step in a horizontal configuration, and rotates the chassis against the step; leveraging against this contact point appropriately while driving the treads, the center of mass may be pushed on top of the step. The tread assemblies may then be rotated up onto the step. After these moves, the robot is backwards relative to how it approached the step; to climb additional steps, the robot must reorient itself. If the length of the robot is less than the length of the step, the robot can flip itself in place on the step. The entire sequence can be repeated to climb multiple steps.

Using the independent nature of the tread assemblies, the robot may drive over a chasm or ditch nearly as wide as the tread assembly is long. The tread assemblies are rotated 180° apart and the chassis is positioned vertically, such that the center of mass is centered above the main drive axle (Fig. 1.4d). Balancing over a chasm is another unstable equilibrium; dynamic stabilization about the roll axis is accomplished by pivoting the tread assemblies to tilt the chassis side-to-side.

A particularly advanced, and challenging, maneuver is balancing on an edge, such as the edge of a step (Fig. 1.4e). Maintaining traction is critical; the material and shape of the treads and edge determine the critical slip angle. The contact angle can be controlled to a value less than this. Ultimately, the robot may drive up a step, contacting only the edge, while pivoting the chassis to keep the center of mass directly over the edge. This maneuver may be repeated up a staircase depending on the rise and pitch angle of the stairs, the length of the treads, and the critical slip angle.

The proposed method of stair climbing via successive perching combines dynamic inverted pendulum balancing with the mechanical simplicity of a treaded

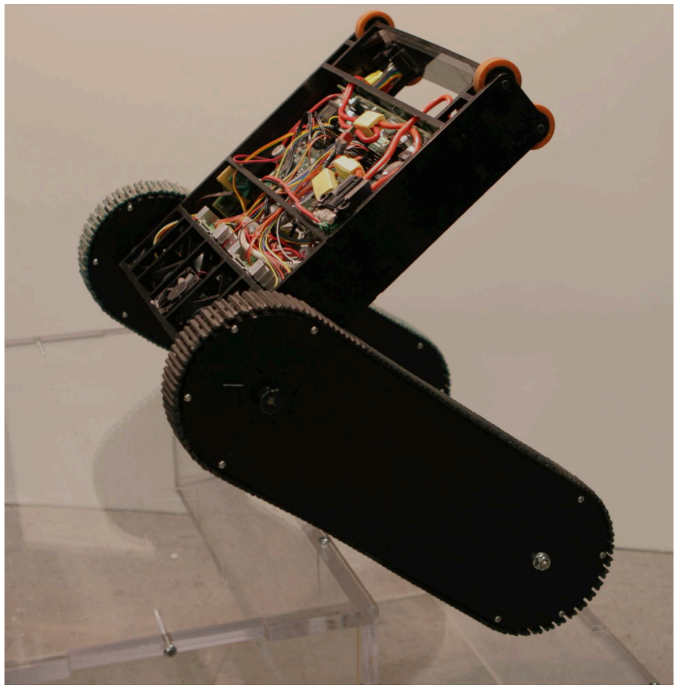


Figure 1.5: Switchblade robot perching on the edge of a stair.

design. The robot approaches the first step balanced on the far end of the tread assemblies with the chassis angled back to keep the center of mass above the contact point with the ground. Once the tread assemblies make contact with the first step edge, the chassis pivots forwards, shifting the center of mass above the step edge. The robot then drives up the step edge, pivoting the chassis appropriately to maintain the center of mass above the contact point and balancing dynamically (Fig. 1.5). At the top of the step, the robot transitions to balancing on the top face of the step and can then climb successive stairs similarly. See Supplemental File 1: Switchblade video for an animation of the stair climbing sequence. This maneuver is compatible with a wide range of step dimensions. This paradigm requires only that the length of the tread, not the sprocket radius, be greater than the rise of the step. Further, the length of the tread does not need to span multiple step edges, as in other treaded stair climbing robots. The smaller required size of the robot decreases cost as well as increases maneuverability in tight spaces, such as in a partially collapsed building. No additional or dedicated sensors or actuators are required for stair climbing, and no external feedback system, such as a vision system, is required or used.

We have previously described a rudimentary stair climbing maneuver that can be performed with the same vehicle. Climbing successive stairs required each step to have sufficient run for the robot to be able to to turn around or flip itself. This method is still effective for thresholds such as street curbs. The maneuver presented in this dissertation, dubbed successive perching, does not have this limitation on step geometry. An overview of the maneuvers the vehicle is capable of is shown in Supplemental File 1: Switchblade video. This vehicle design has been recognized for scoring well in both versatility and mechanical complexity metrics [44].

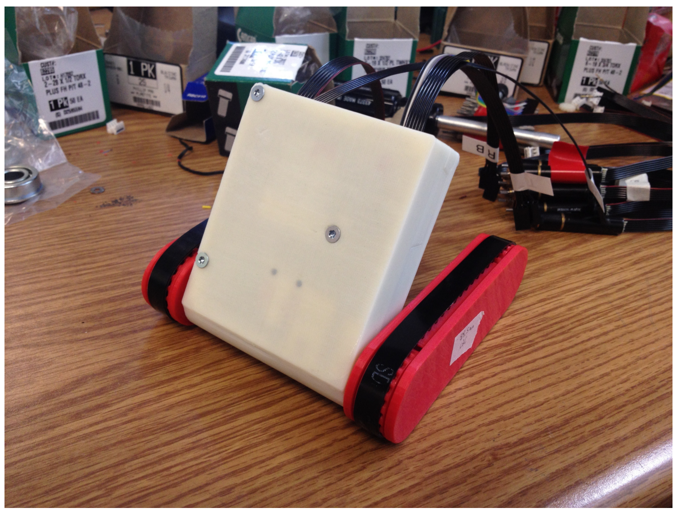


Figure 1.6: Completed Switchblade nano prototype.

Switchblade nano

The *nano* incarnation of the Switchblade platform (see Fig. 1.6) was an experiment to see how small the design could be shrunk and still operate in the same manner. The main motivation for decreasing size was potential commercialization and cost reduction. The robot may be seen performing multiple maneuvers in Supplemental File 2: Switchblade nano video.

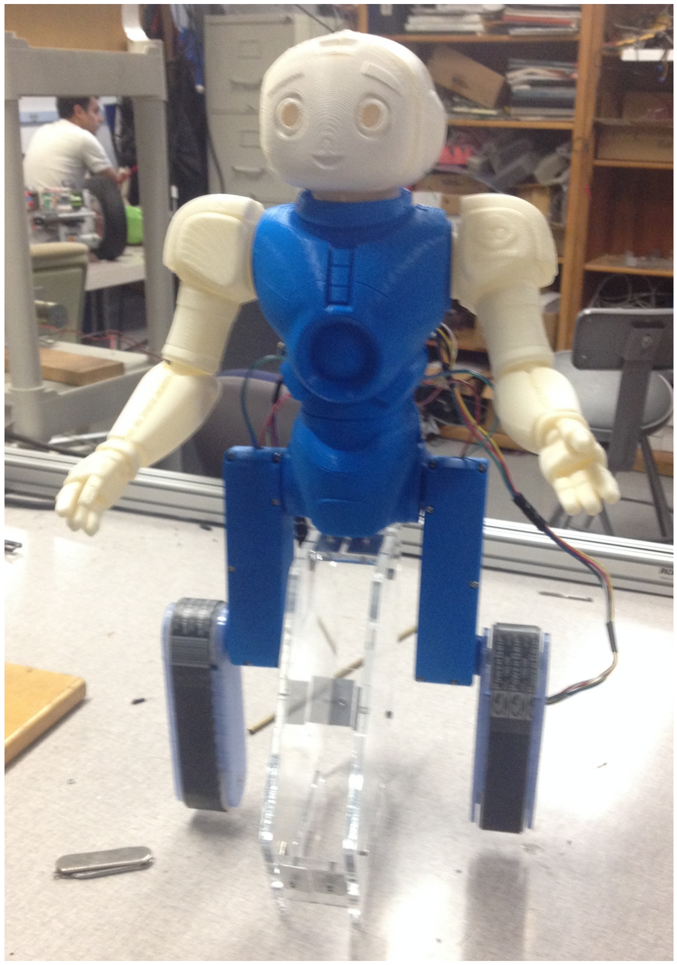


Figure 1.7: Switchbot prototype.

Switchbot

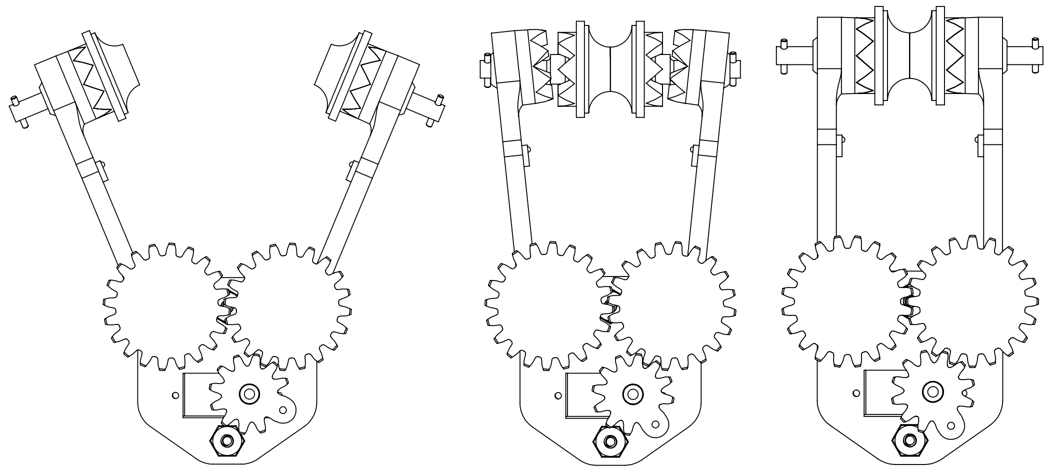
The last variant of the Switchblade platform is an anthropomorphic version with additional powered links between the chassis and tread segments (creating distinct upper and lower leg segments) as well as passive arms and a head (see Fig. 1.7). The motivation for this project is potential commercialization as a consumer electronics product, with surveillance and patrol features. The appearance of the upper body was designed by Adam Fairless. The robot may be seen performing multiple maneuvers in Supplemental File 3: Switchbot video.

Figure 1.8: SkySweeper prototype.

1.4.4 SkySweeper

In this dissertation, in part, we present a novel new design for a cable-locomoting robot, which has few actuators, but multiple modes of locomotion for achieving different objectives. SkySweeper is symmetrically comprised of two links of equal length which are pivotally connected with a rotary series elastic actuator (SEA) at one end [45], see Fig. 1.8. The SEA consists of a motor and a torsion spring connected in series. The motor housing is connected to the first link, the motor shaft is connected to one end of the spring, and the other end of the spring is connected to the second link. The motor exerts equal and opposite torques on the first link and the SEA shaft. The spring exerts equal and opposite torques on the SEA shaft and second link. At the opposite end of each link is an actuated clamp which can hold on to a cable. The clamp can be in one of three positions, as illustrated in Fig. 1.9:

1. Open, in which the clamp is completely open;
2. Rolling, in which the clamp is partially closed and may passively roll along the cable;
3. Pivoting, in which the clamp is fully closed and may only pivot on the cable.



(a) Open (b) Rolling (c) Pivoting

Figure 1.9: Different clamp positions.

With two three-position clamps, there are a total of nine possible configurations. A sensor in each clamp detects when a cable is within grasp of the clamp. The SEA in the joint allows the robot to store extra potential energy before commencing a dynamic maneuver. By appropriately combining the actuation of the elbow SEA and the clamps, several modes of locomotion are possible. The robot may be seen performing multiple maneuvers in Supplemental File 4: SkySweeper video.

Inchworm

In this maneuver, the robot has both clamps on the cable, one in pivoting position and the other in rolling position. The SEA is first actuated to increase the angle between the two links, then the positions of the clamps are alternated, the direction of the SEA is reversed to close the angle between the two links, and the clamp positions alternate again. The entire procedure is repeated which creates a successive “inchworm”-like motion to traverse the cable. This sequence of motions can be performed slowly (quasi-statically) for precise position control or quickly (dynamically with the SEA) to move faster.

Swing & Roll

The robot begins with both clamps on the cable, the motor in the SEA is allowed to spin idly. The first clamp is in the pivoting position and then the second clamp opens, causing the second link to pivot and fall away from the cable. As the center of mass rotates under the first clamp, the first clamp switches to the rolling position and the momentum from the swinging second link causes the robot to roll along the cable. If the first clamp were in rolling position the entire time, there would be zero net horizontal displacement. This maneuver requires nearly zero control effort (just the small amount of energy required to actuate the clamps) and instead converts some of the gravitational potential energy into translational kinetic energy. Rolling resistance limits how far the robot will roll before coming to a stop. If the cable has a downward slope sufficient to overcome the rolling resistance, the robot will continue to roll. This maneuver allows efficient locomotion on horizontal and downward sloping cables.

Swing-Up

After the robot performs the previous maneuver, it comes to a rest with only the first clamp on the cable and both links hanging down vertically. In order to perform the inchworm maneuver, it is necessary to swing the second link up to grasp the cable. The swing-up maneuver starts with the first clamp in pivoting position on the cable and the second clamp open hanging down. A sinusoidal input is sent to the motor and the robot pivots and swings until the second clamp can grasp the cable. The frequency and magnitude of the sinusoid are chosen based on the physical parameters of the system. This maneuver is also useful for installing the robot on the cable.

Backflip

Instead of rolling along the cable, as in the inchworm and swing & roll maneuvers, the robot may also move along the cable by flipping end-over-end. The robot starts with both clamps on the cable in the pivoting position. In this configuration, all degrees of freedom of the robot are constrained, except for the spring in the SEA. The motor is driven to preload the spring and then the second clamp is opened. The force of the SEA and gravity cause the second link to rapidly pivot away from the cable. The motor continues (with less power) to rotate the second link relative to the first as the entire robot pivots about the first clamp until the second clamp grasps the cable. The spring is then preloaded in the opposite direction and the first clamp opens. The entire robot pivots about the second clamp in a similar manner as before until the first clamp can grasp the cable. This sequence of motions can be repeated for successive flips. An important advantage of this maneuver is that overhead obstacles, such as supports from which the cable hangs, may be bypassed.

All of the aforementioned maneuvers happen in the plane of the robot. The clamps constrain the robot from twisting out of plane. In an application environment, high winds could make some of the maneuvers (swing-up, backflip) untenable. Existing cable-locomoting robots are also susceptible to high winds. Due to the symmetry of the links and clamps, all maneuvers can be performed to move in either direction.

1.5 Acknowledgements

Part of this chapter concerning Switchblade has appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning SkySweeper has appeared in print: N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc.*

IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems, pp. 2339-2344, 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the DC motor dynamometer has appeared in print: N. Morozovsky, R. Moroto, T. Bewley, "RAPID: an inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the perching maneuver has been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

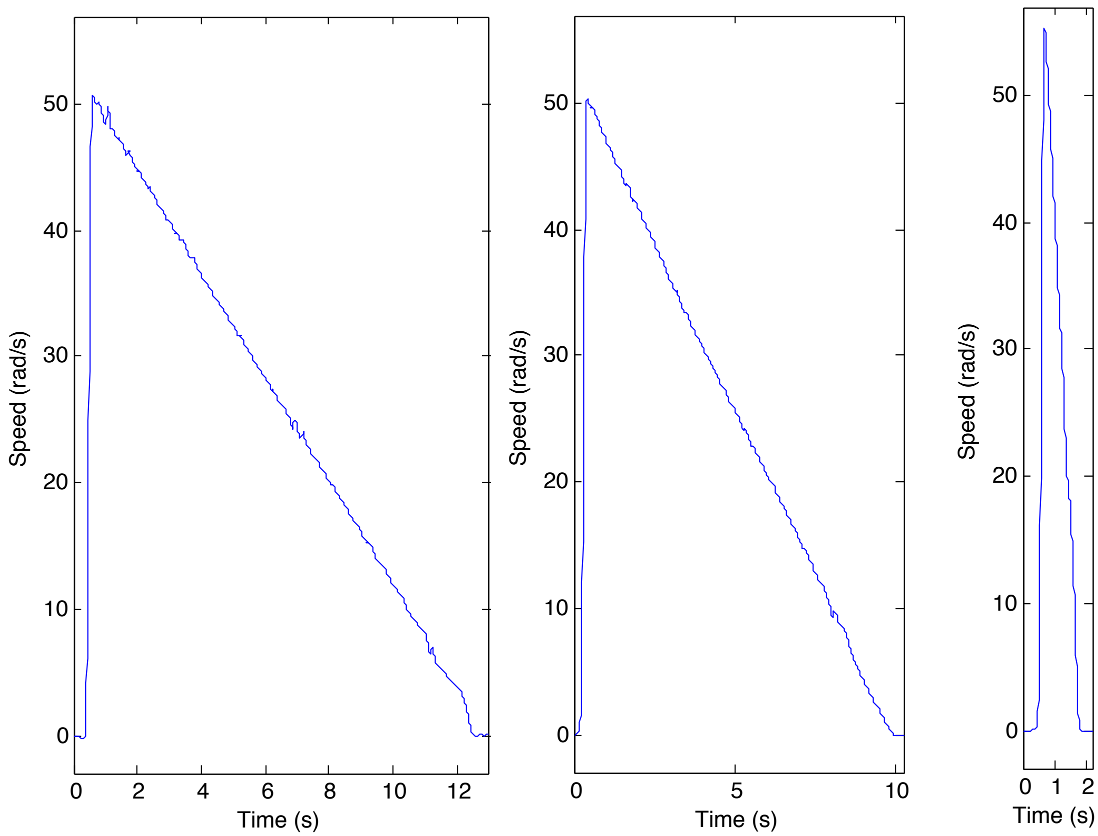
Chapter 2

Dynamics

In this chapter, the dynamics of the DC motor dynamometer are derived from first principles, Newton's second law and a voltage balance across the motor terminals. Next, the Lagrangian method of determining equations of motion is shown, without loss of generality, for a constrained system. The motor model from the dynamometer fits cleanly into the dynamical model. Finally, the particular equations of motion and constraints of the different robotic systems is shown. While the repeated presentation of the same process may appear redundant, it is useful for learning purposes to see the process applied multiple times. It is also educational to first derive the equations of motion by hand (with copious calculation of partial derivatives) before skipping to the programmatic derivation of the equations, see Sec. 2.2. The author only found it necessary to derive the equations for a mobile inverted pendulum by hand before switching to the programmatic method. Systems with higher degrees of freedom will have exponentially more partial derivatives to calculate.

2.1 DC Motor Dynamometer

The dynamometer system (Fig. 1.1), comprised of the motor under test and the reconfigurable inertial load constitute a simple single degree of freedom mechanical system, and as such, it is straightforward to write the equations of motion from Newton's second law.



(a) Fully loaded. (b) Half loaded. (c) Unloaded.

Figure 2.1: Results from spin down tests with different loading configurations of inertial disc.

In order to understand the dynamics of the base system (without a motor installed), we perform multiple spin down tests. We manually apply an impulse to the system and log the position data as the inertial disc slows to a stop due to friction (Fig. 2.1). A total of 44 trials were run, clockwise and counter-clockwise, at different initial speeds, and in three configurations of the inertial disc: fully loaded (8 bolts and 24 nuts, Fig. 2.1a), half loaded (4 bolts and 12 nuts, Fig. 2.1b), and unloaded (0 bolts and 0 nuts, Fig. 2.1c). The linear slope of the system slowing down demonstrates that Coulomb friction is dominant and viscous and quadratic drag are negligible. We would then expect to be able to model the dynamics with the equation:

$$J\frac{d\omega}{dt} = -c \cdot \mathrm{sgn}(\omega), \tag{2.1}$$

where c is the Coulomb friction coefficient. To verify this model, we performed a simple linear regression for each of the trials. In each case, the coefficient of determination, R^2, of the linear fit (2.1) exceeded 0.98, indicating a good fit.

The bearings are the main source of Coulomb friction in the system. Adding nuts and bolts to the inertial disc increases both inertia and weight, which increases the radial load on the bearings, increasing friction. The friction is not directly proportional to the inertia, otherwise the ratio c/J would be constant and spin down tests with different inertias would have the same slope. This is clearly not the case, as can be seen by the different slopes in Fig. 2.1. We thus use the following two term model to characterize the Coulomb friction:

$$c = \alpha + \beta J. \tag{2.2}$$

Since the nuts and bolts are all added at the same distance from the axis of rotation, the added weight is directly proportional to the added inertia. Thus the increased Coulomb friction from the increased weight is captured by the βJ term. From (2.1) and (2.2), we can solve for the minimum, unloaded inertia of the base system. We call the empirically measured, averaged slopes of the spin down tests x, y, and z for the unloaded, half loaded, and fully loaded cases, respectively. Using (2.1) and noting that the signum function will always be the opposite sign of $\frac{d\omega}{dt}$ for a spin down trial, we can write:

$$\begin{aligned} (J_B)x &= -[\alpha + \beta(J_B)], \\ (J_B + J_N)y &= -[\alpha + \beta(J_B + J_N)], \\ (J_B + 2J_N)z &= -[\alpha + \beta(J_B + 2J_N)], \end{aligned}$$

where J_B is the inertia of the unloaded base system and J_N is the inertia of 4 bolts and 12 nuts. These three equations can be solved for the three unknowns: J_B, α, and β in terms of the known values J_N, x, y, and z:

$$\begin{aligned} J_B &= 2J_N(y-z)/(x-2y+z), \\ \alpha &= \frac{-2J_N(x-y)(y-z)(x-z)}{(x-2y+z)^2}, \\ \beta &= \frac{2(x-y)(y-z)}{x-2y+z} - y. \end{aligned} \tag{2.3}$$

We find $J_B = 3.37 \cdot 10^{-5}$ kg·m^2, $\alpha = 1.72 \cdot 10^{-3}$ N·m, and $\beta = 2.68$ N·m/(kg·m^2).

With the dynamics of the base system understood, we can proceed to derive equations from first principles that describe the electrical and mechanical dynamics of the motor and base system together. Taking into account the inductance l and resistance r of the motor armature wire, and the back EMF generated when the motor spins, which is equal to the motor constant, k, times the motor shaft velocity, ω, we can write the following voltage balance across the motor terminals:

$$V = l\frac{di}{dt} + ri + k\omega, \tag{2.4}$$

where i is the current through the motor armature wire and V is the voltage across the motor terminals. The torque generated by the motor is

$$\tau = ki, \tag{2.5}$$

where the torque equals the same motor constant, k, times the current, i. The net torque from the motor, subtracting frictional losses, is

$$\tau_N = ki - b\omega - c_M \cdot sgn(\omega), \tag{2.6}$$

where b and c_M are the viscous and Coulomb friction coefficients of the motor, respectively. The two equations above (2.4), (2.6) form the commonly accepted brushed DC motor model [46]; however we choose to use a simplified electrical model with the the average voltage and without inductance:

$$\overline{V} = ri + k\omega, \tag{2.7}$$

where $\overline{V} = V_S\mathbf{u}$ is defined in section 5.3.2. The inductance of the armature wire of a small DC motor can be on the order of 10^{-4} Henries, which is multiple orders of magnitude smaller than the other parameters being estimated. The PWM signal to the motor also increases the frequency of $\frac{di}{dt}$ faster than the sampling frequency. Both of these factors make it difficult to estimate the inductance. The simplified model (2.7) also allows us to write an equation for the torque generation of the motor (not including frictional losses) as a function of the signed PWM duty cycle $u \in [-1, 1]$ and speed ω, instead of a function of current as in (2.5):

$$\tau = \sigma u - \zeta\omega, \quad \text{where} \qquad (2.8)$$
$$\sigma = kV_S/r, \quad \zeta = k^2/r.$$

The expression for the stall torque, σ, is found by setting $u = 1$ and $\omega = 0$ in (2.5) and (2.7), and solving for τ. The expression for the back EMF damping coefficient (an electrical term that does not include frictional effects), ζ, can be found by setting $u = 1$ and $i = 0$ in (2.5), (2.7), and (2.8) and solving for ζ. This is a more useful formulation than (2.5) because it is more practical to control a motor by voltage PWM (see section 5.3.2) than controlling current. For these reasons, we choose not to estimate the inductance.

If the motor under test has a gearbox, with reduction γ, the total effective inertia of the motor and gearbox is

$$J_E = J_{Gearbox} + \gamma^2 J_{Motor}.$$

The total system inertia, J_S, includes the minimal base inertia, found in (2.3), any additional inertia from nuts and bolts added to the inertial disc (known *a priori*), and the inertia of the motor under test:

$$J_S = J_B + n_{Bolts}J_{Bolt} + n_{Nuts}J_{Nut} + J_E. \qquad (2.9)$$

We can write the torque balance of the entire system, starting with (2.1) and adding the torque of the motor (2.6):

$$J_S \frac{dw}{dt} = ki - b\omega - c_S \cdot sgn(\omega), \qquad (2.10)$$

where Coulomb friction from the motor and base system are combined into the one parameter c_S:

$$c_S = \alpha + \beta(J_S - J_E) + c_M. \qquad (2.11)$$

We subtract the effective inertia of the motor and gearbox so that the Coulomb friction calculated from the results of the spin down tests only reflects the inertial disc with any nuts or bolts and all Coulomb friction contributions from the motor are lumped into c_M.

This model is only applicable when the system is in motion, as it accounts for Coulomb and viscous friction, but not stiction. We thus ignore measurements where the velocity is zero. In the general case, Coulomb friction is nonlinear because of the step at zero velocity, however, in this case, we are ignoring data with zero velocity and thus fitting two linear models to the data with non-zero velocity, bypassing the nonlinearity.

Thus we have two equations (2.7) and (2.10) that completely describe the electrical and mechanical dynamics of the motor in RAPID. We use gray-box modeling to find the five unknown parameters of the system: r, k, J_S, b, and c_S that best fit the observed data to the prescribed model [47]. The effective inertia of the motor, J_E, and the Coulomb friction of the motor, c_M, can be recovered from J_S and c_S with equations (2.9) and (2.11) respectively.

2.2 Lagrangian Dynamics

The Lagrangian method of dynamics is a powerful tool to describe the behavior of rigid body systems and is broadly applicable to robotics. Lagrangian dynamics has several advantages over other dynamical formulations (though they are mathematically equivalent), it does not require the explicit calculation of internal forces and it is straightforward to impose both holonomic and non-holonomic constraints [48]. Furthermore, it lends itself to programmatic derivation of equations of motions starting only with geometric relations. App. B contains MATLAB code (using the symbolic math toolbox) that automatically generates equations of motion for a mobile inverted pendulum given just the system configuration, without requiring the calculation by hand of the kinetic or potential energies. The nonlinear equations can be used directly as the plant model for a simulation and can be linearized to design a controller, see Ch. 3.

The Lagrangian of a system is written as the difference of the kinetic energy, T, and potential energy, V, of the system

$$\mathcal{L} = T - V. \tag{2.12}$$

A system with n degrees of freedom has a vector of generalized coordinates $q \in \mathbb{R}^n$ and generalized forces $Q \in \mathbb{R}^n$. The Euler-Lagrange equations are

$$\frac{d}{dt}\left(\frac{\delta \mathcal{L}}{\delta \dot{q}_i}\right) - \frac{\delta \mathcal{L}}{\delta q_i} = Q_i, \tag{2.13}$$

which can be rewritten into the form

$$M(q)\ddot{q} + F(q, \dot{q}) = Q,$$

where $M(q)$ is the positive definite mass matrix and $F(q, \dot{q})$ is the vector of forces. Constraints in the form

$$A(q)\dot{q} = 0 \tag{2.14}$$

can be appended thusly

$$M(q)\ddot{q} + F(q, \dot{q}) = Q + A(q)^T \lambda, \tag{2.15}$$

where λ is the Lagrange multiplier. We can avoid the need to calculate λ by first defining $S(q)$ as an orthonormal basis for the null space of $A(q)$. Then premultiplying (2.15) by $S(q)^T$

$$S(q)^T M(q)\ddot{q} + S(q)^T F(q, \dot{q}) = S(q)^T Q. \tag{2.16}$$

In the case where S is constant (not a function of q), ν and $\dot{\nu}$ can be defined by $\dot{q} = S\nu$ and $\ddot{q} = S\dot{\nu}$ and (2.16) can be further simplified

$$S^T M(q) S\dot{\nu} + S^T F(q, \dot{q}) = S^T Q.$$

In which case it is trivial to solve for $\dot{\nu}$ since $S^T M(q) S$ is positive definite, and therefore, invertible.

$$\dot{\nu} = (S^T M(q) S)^{-1} S^T [Q - F(q, \dot{q})],$$

and similarly

$$\ddot{q} = S[S^T M(q)S]^{-1} S^T [Q - F(q, \dot{q})]. \tag{2.17}$$

which is simply a set of nonlinear second order differential equations which can be marched forward in time by traditional means. In the case where the constraints are dependent on q, an alternate technique can be used to apply the constraints, see the latter half of Sec. 2.4.4 for an example.

In the (common) case that the generalized forces Q are exerted by DC motors in the robot, we can set $Q = B\tau$, where $\tau \in \mathbb{R}^{n_u}$ is the vector of motor torques, n_u is the number of motors, and the matrix $B \in \mathbb{R}^{n \times n_u}$ maps the motor torques to the generalized coordinate, often with equal and opposite elements for an internally mounted motor. We can then substitute into (2.17)

$$\ddot{q} = S[S^T M(q)S]^{-1} S^T [B\tau - F(q, \dot{q})]. \tag{2.18}$$

Next substituting the motor model (2.8) derived in section 2.1

$$\ddot{q} = S[S^T M(q)S]^{-1} S^T \{B[\Sigma u - Z(\dot{q})] - F(q, \dot{q})\}, \tag{2.19}$$

where $\Sigma \in \mathbb{R}^{n_u \times n_u}$ is the diagonal matrix of motor stall torques and $Z(q) \in \mathbb{R}^{n_u}$ is the vector of back EMF damping terms.

The following sections develop equations of motion for robotic systems using the above approach, which is broadly applicable to robotics systems.

2.3 Mobile Inverted Pendula

Both the myMIP (Fig. 1.2a) and eyeFling (Fig. 1.2b) share the same dynamics, but with different parameters (masses, lengths, inertias, and motor parameters). Following the above discussion of Lagrangian dynamics, we start by defining the generalized coordinates in Fig. 2.2

$$q = \begin{pmatrix} \gamma & x & \phi & \theta \end{pmatrix}^T,$$

where γ is the angle of the motor shaft with respect to vertical, x is the horizontal displacement of the center of rotation, ϕ is the wheel angle with respect to vertical,

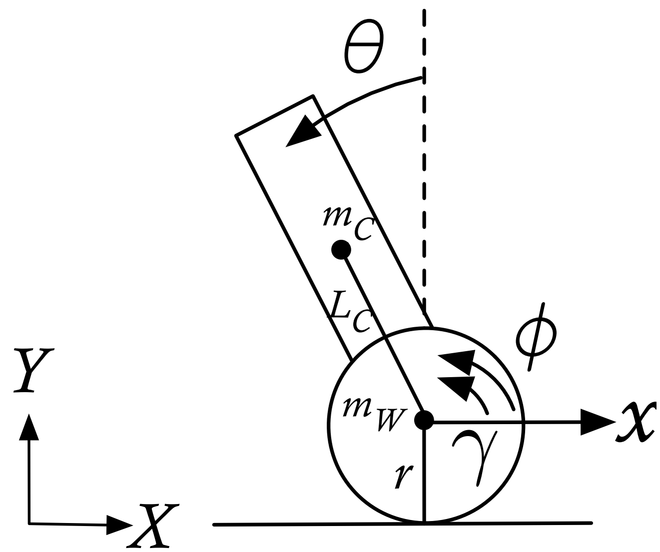


Figure 2.2: Coordinate system for Mobile Inverted Pendulum.

and θ is the angle of the chassis with respect to vertical. The motor is housed in the chassis, so the relative rotation of the motor shaft is given by $\gamma - \theta$. The kinetic energies of the motors, wheels, and chassis are given, respectively, by

$$\begin{aligned}
T_M &= \frac{1}{2} J_M (\dot{\gamma} - \dot{\theta})^2, \\
T_W &= \frac{1}{2} (m_W \dot{x}^2 + J_S \dot{\phi}^2), \\
T_C &= \frac{1}{2} [m_C (\dot{x} - L_C \dot{\theta} \cos\theta)^2 + L_C^2 \dot{\theta}^2 \sin\theta^2 + J_C \dot{\theta}^2].
\end{aligned}$$

The potential energy is

$$V = g[m_C L_C \cos\theta + r(m_W + m_C)].$$

The Lagrangian, following (2.12), is $\mathcal{L} = T_M + T_W + T_C - V$. Solving the Euler-Lagrange equations (2.13), we can write equations of motion in the form of (2.18), where τ is the torque from the motor and the matrix B maps the torque to the generalized coordinates, in this case, applying equal and opposite torques between the motor shaft and chassis

$$B = \begin{pmatrix} 2 \\ 0 \\ 0 \\ -2 \end{pmatrix},$$

where the factor of two comes from the fact that there are two motors (right and left). Using the motor model developed above (2.8)

$$\ddot{q} = S[S^T M(q)S]^{-1} S^T \{B[\sigma u - \zeta(\dot{\gamma} - \dot{\theta})] - F(q, \dot{q})\}. \tag{2.20}$$

In this system, we wish to constrain the wheel to the ground (no slip condition, $x = -r\phi$) and the motor shaft to the wheel (no backlash, $\gamma = \phi$). The constraint matrix A can then be written

$$A = \begin{bmatrix} 0 & 1 & r & 0 \\ 1 & 0 & -1 & 0 \end{bmatrix}, \tag{2.21}$$

an orthonormal basis for the null space of which is

$$S = \begin{bmatrix} 1/\sqrt{r^2+2} & 0 \\ -r/\sqrt{r^2+2} & 0 \\ 1/\sqrt{r^2+2} & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.22}$$

which completes the equations of motion (2.20) for a mobile inverted pendulum.

2.4 Switchblade

2.4.1 V-Balance

For the V-Balance maneuver shown in Fig. 1.4b, the equations of motions can be derived as follows. By symmetry, we simplify the model to three bodies in two dimensions, the angles are defined in Fig. 2.3, where θ and α are the angles of the chassis and tread assembly from vertical respectively, ϕ is the rotation angle of the tread sprocket, and x is the horizontal position of the robot. Motors on the

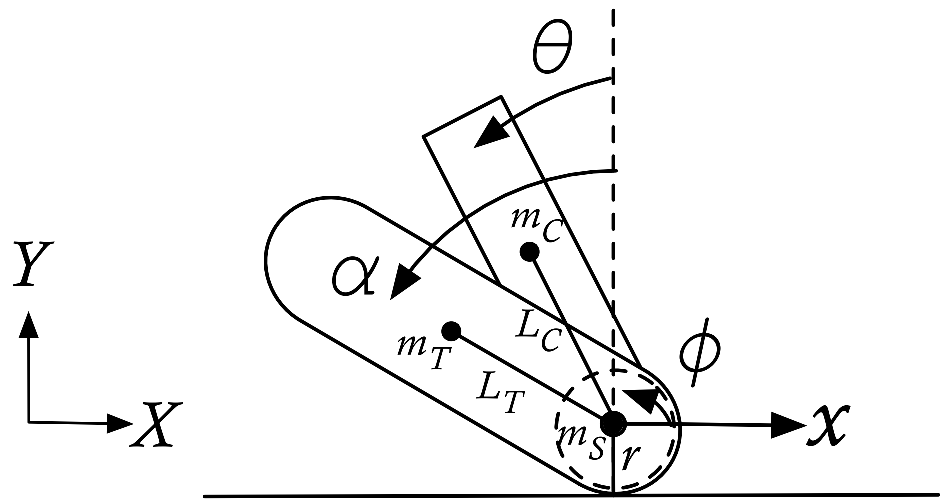


Figure 2.3: Coordinate system for V-balance mode and parameters.

robot may exert torques along the axis of rotation between the tread sprocket and the chassis and between the chassis and the tread assembly.

The kinetic energies of the sprocket, chassis, and tread assembly are given respectively by

$$T_S = \frac{1}{2}\left[m_S\dot{x}^2 + J_S\dot{\phi}^2\right],$$
$$T_C = \frac{1}{2}\left[m_C\left(\dot{x} - L_C\dot{\theta}\cos\theta\right)^2 + m_C\left(L_C\dot{\theta}\sin\theta\right)^2 + J_C\dot{\theta}^2\right],$$
$$T_T = \frac{1}{2}\left[m_T\left(\dot{x} - L_T\dot{\alpha}\cos\alpha\right)^2 + m_T\left(L_T\dot{\alpha}\sin\alpha\right)^2 + J_T\dot{\alpha}^2\right].$$

The gravitational potential energy is given by

$$V = m_C g L_C \cos\theta + m_T g L_T \cos\alpha.$$

We define the generalized coordinates as

$$q = \begin{pmatrix} x & \phi & \alpha & \theta \end{pmatrix}^T. \tag{2.23}$$

The Lagrangian can be written as $\mathcal{L} = T_S + T_C + T_T - V$. By solving the Euler-Lagrange equations, we can write the equations of motion in the form

$$M(q)\ddot{q} + F(q,\dot{q}) = B\tau, \tag{2.24}$$

where the τ vector represents the control input torques for motors located in the robot's chassis. The first element represents the motor torque between the chassis and the tread sprocket and the second element represents the motor torque between the chassis and the tread assembly. Note also that:

$$M(q) = \begin{bmatrix} m_S + m_C + m_T & 0 & -m_T L_T \cos\alpha & -m_C L_C \cos\theta \\ 0 & J_S & 0 & 0 \\ -m_T L_T \cos\alpha & 0 & m_T L_T^2 + J_T & 0 \\ -m_C L_C \cos\theta & 0 & 0 & m_C L_C^2 + J_C \end{bmatrix},$$

$$F(q,\dot{q}) = \begin{pmatrix} m_C L_C \dot{\theta}^2 \sin\theta + m_T L_T \dot{\alpha}^2 \sin\alpha \\ 0 \\ -m_T g L_T \sin\alpha \\ -m_C g L_C \sin\theta \end{pmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}.$$

We next impose a no-slip constraint between the tread sprocket and the ground, which will be shown also achieves a coordinate reduction, via

$$x + r\phi = 0, \tag{2.25}$$

or equivalently

$$A\dot{q} = 0, \quad A = \begin{pmatrix} 1 & r & 0 & 0 \end{pmatrix}.$$

In this system, A is not dependent on q. We append (2.24) with the inner product of the constraint matrix A with λ, the Lagrange multiplier:

$$M(q)\ddot{q} + F(q,\dot{q}) = B\tau + A^T\lambda. \tag{2.26}$$

A basis for the null space of A is given by

$$S = \begin{bmatrix} -r & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.27}$$

Given that $\dot{q}$ is in this space, we define ν accordingly as

$$\dot{q} = S\nu, \quad \nu = \begin{pmatrix} \dot{\phi} & \dot{\alpha} & \dot{\theta} \end{pmatrix}^T. \tag{2.28}$$

Premultiplying by S^T and using (2.28), we can rewrite (2.26) as

$$S^T M(q) S\dot{\nu} + S^T F(q,\dot{q}) = S^T B\tau. \tag{2.29}$$

Noting that the dynamics of x and ϕ are directly coupled by (2.25), we can choose a reduced coordinate set $q_r = \begin{pmatrix} \phi & \alpha & \theta \end{pmatrix}^T$. Likewise truncating the top row of (2.27), $S_r = I_{3\times 3}$ and we see that $\dot{q}_r = \nu$ directly from (2.28).

Finally, we model the torque output τ from each motor linearly as (derived from (2.8))

$$\tau = \sigma u - \zeta\omega, \quad \sigma = \frac{k\gamma V}{R}, \quad \zeta = \frac{(k\gamma)^2}{R}, \tag{2.30}$$

where σ is the stall torque, u is the control input (limited to $[-1, 1]$), ζ is the damping coefficient of the motor, ω is the speed of the motor shaft relative to the motor body, k is the motor constant, γ is the gear ratio of the transmission, V is the nominal voltage applied across the terminals, and R is the terminal resistance. Substituting the motor model, we can rewrite (2.29) using (2.30), where $Z(q)$ is effectively Rayleigh's dissipation function

$$S^T M(q) S\dot{\nu} + S^T F(q,\dot{q}) = S^T B[\Sigma\mathbf{u} - Z(\dot{q})], \tag{2.31}$$

$$\sigma = \begin{pmatrix} \sigma_S & 0 \\ 0 & \sigma_T \end{pmatrix}, \quad Z(\dot{q}) = \begin{pmatrix} \zeta_S(\dot{\phi} - \dot{\theta}) \\ \zeta_T(\dot{\alpha} - \dot{\theta}) \end{pmatrix}.$$

Concatenating q_r and ν yields a complete state vector x (recycling the superfluous generalized coordinate x for lack of a longer alphabet). Rewriting (2.31), we see that this nonlinear system is affine in the inputs:

$$x = \begin{pmatrix} q_r \\ \nu \end{pmatrix}, \quad \dot{x} = f(x) + \Gamma(x)u,$$

$$f(x) = \begin{pmatrix} \nu \\ -[S^T M(q) S]^{-1} S^T [F(q,\dot{q}) + BZ(\dot{q})] \end{pmatrix},$$

$$\Gamma(x) = \begin{pmatrix} 0_{3\times 2} \\ \left[S^T M(q) S\right]^{-1} S^T B\Sigma \end{pmatrix}.$$

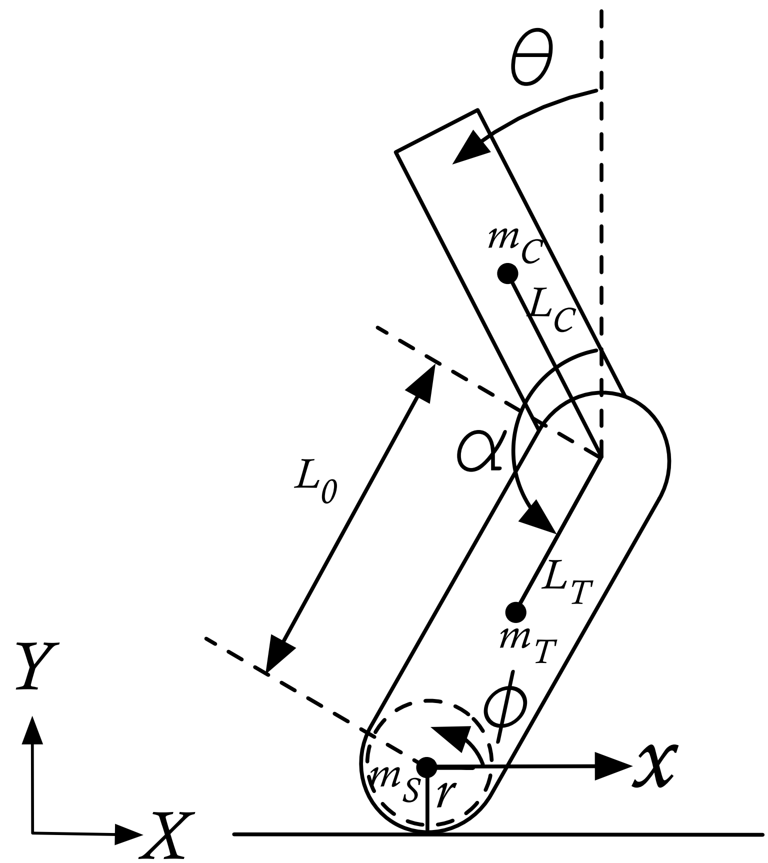


Figure 2.4: Generalized coordinates for Switchblade C-Balancing.

2.4.2 C-Balancing

The dynamics for the C-Balancing maneuver can be derived similarly as in the above section, with different expressions for kinetic and potential energy as functions of similar generalized coordinates which are defined in Fig. 2.4. The full equations of motion are left as an exercise to the reader. The chassis is shorter than the tread assemblies and can rotate continuously, passing freely through the tread assemblies.

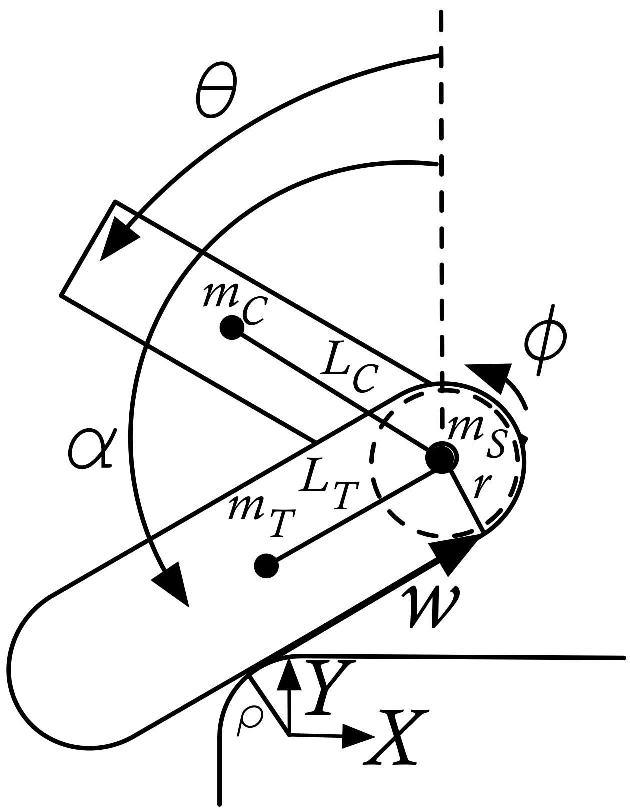


Figure 2.5: Generalized coordinates and key dimensions.

2.4.3 Perching

In order to create a simulation and design a controller, we first derive the dynamics of the system using the Lagrangian method. By symmetry, we simplify the model to three bodies in two dimensions, the generalized coordinates are defined in Fig. 2.5, θ and α are the angles of the chassis and unified right and left tread assemblies from vertical respectively, ϕ is the rotation angle of the unified right and left tread sprockets, and w is the distance between the tread sprocket and the contact point on the step edge measured along the tread. The radius of curvature of the step edge is ρ; a sharp edge is modeled by setting $\rho = 0$. The inertial frame XY is fixed at the center of the curved edge. Motors on the robot exert torques between the tread sprocket and the chassis and between the chassis and the tread assembly.

The kinetic energies of the unified sprockets, unified tread assemblies, chassis, unified sprocket motors, and unified tread assembly motors are given, respectively, by

$$\begin{aligned}
T_S =& \frac{1}{2}\{m_S[(\dot{w}\cos\alpha - \dot{\alpha}(r\cos\alpha + \rho\cos\alpha + w\sin\alpha))^2 \\
&+ (\dot{w}\sin\alpha - \dot{\alpha}(r\sin\alpha - w\cos\alpha + \rho\sin\alpha))^2] + J_S\dot{\phi}^2\}, \\
T_T =& \frac{1}{2}\{m_T[(\dot{\alpha}(r\cos\alpha - L_T\sin\alpha + \rho\cos\alpha + w\sin\alpha) - \dot{w}\cos\alpha)^2 \\
&+ (\dot{\alpha}(L_T\cos\alpha + r\sin\alpha - w\cos\alpha + \rho\sin\alpha) - \dot{w}\sin\alpha)^2] + J_T\dot{\alpha}^2\}, \\
T_C =& \frac{1}{2}\{m_C((\dot{w}\cos\alpha - \dot{\alpha}(r\cos\alpha + \rho\cos\alpha + w\sin\alpha) + L_C\dot{\theta}\sin\theta)^2 \\
&+ (\dot{\alpha}(r\sin\alpha - w\cos\alpha + \rho\sin\alpha) - \dot{w}\sin\alpha + L_C\dot{\theta}\cos\theta)^2) + J_C\dot{\theta}^2\}, \\
T_{SM} =& \frac{1}{2}\left[J_{SM}(\dot{\phi} - \dot{\theta})^2\right], \quad T_{TM} = \frac{1}{2}\left[J_{TM}(\dot{\alpha} - \dot{\theta})^2\right],
\end{aligned}$$

where m_S, m_T, J_S, J_T, J_{SM}, and J_{TM} represent the combined masses and inertias of the right and left components for simplicity. The gravitational potential energy is given by

$$\begin{aligned}
V = g[&m_S(r\sin\alpha - w\cos\alpha + \rho\sin\alpha) \\
&+ m_T(L_T\cos\alpha + r\sin\alpha - w\cos\alpha + \rho\sin\alpha) \\
&+ m_C(L_C\cos\theta + r\sin\alpha - w\cos\alpha + \rho\sin\alpha)].
\end{aligned}$$

We define the generalized coordinates as

$$q = \begin{pmatrix} w & \phi & \alpha & \theta \end{pmatrix}^T.$$

The Lagrangian can be written as $\mathcal{L} = T_S + T_T + T_C + T_{SM} + T_{TM} - V$. By solving the Euler-Lagrange equations, we can write the equations of motion in the form

$$M(q)\ddot{q} + F(q,\dot{q}) = B\tau + \frac{\delta P}{\delta \dot{q}}, \tag{2.32}$$

The (positive definite) mass matrix, $M(q)$, is given by:

$$
\begin{aligned}
M_{1,1}(q) =& m, & M_{1,2}(q) = 0, \\
M_{1,3}(q) =& -(r+\rho)m, & M_{1,4}(q) = -L_C m_C \sin(\alpha-\theta), \\
M_{2,2}(q) =& J_S + J_{SM}, & M_{2,3}(q) = 0, \\
M_{2,4}(q) =& -J_{SM}, \\
M_{3,3}(q) =& J_T + J_{TM} + m_T L_T (L_T - 2w) + m[(r+\rho)^2 + w^2], \\
M_{3,4}(q) =& L_C m_C [(r+\rho)\sin(\alpha-\theta) - w\cos(\alpha-\theta)] - J_{TM}, \\
M_{4,4}(q) =& m_C L_C^2 + J_C + J_{SM} + J_{TM},
\end{aligned}
$$

where $m = m_S + m_T + m_C$. The vector $F(q, \dot{q})$ is given by:

$$
\begin{aligned}
F_1(q,\dot{q}) =& m_T L_T \dot{\alpha}^2 - m w \dot{\alpha}^2 - m g \cos\alpha + m_C L_C \dot{\theta}^2 \cos(\alpha-\theta), \\
F_2(q,\dot{q}) =& 0, \\
F_3(q,\dot{q}) =& 2\dot{\alpha} m w \dot{w} + m g (r+\rho)\cos\alpha + g(m w - m_T L_T)\sin\alpha \\
& - 2 L_T \dot{\alpha} m_T \dot{w} - m_C L_C \dot{\theta}^2 [(r+\rho)\cos(\alpha-\theta) + w\sin(\alpha-\theta)], \\
F_4(q,\dot{q}) =& L_C m_C [(r+\rho)\dot{\alpha}^2 \cos(\alpha-\theta) - 2\dot{\alpha}\dot{w}\cos(\alpha-\theta) \\
& + \dot{\alpha}^2 w \sin(\alpha-\theta) - g\sin\theta],
\end{aligned}
$$

The right hand side of (2.32) is the sum of the generalized forces of the system. The power dissipation function P [49] accounts for the Coulomb friction of the treads rubbing against the tread assemblies and the Coulomb friction between the chassis and tread assemblies

$$
P = -\frac{\mu_k m g r}{\sin\alpha}(\dot{\phi} - \dot{\alpha}) - c_T(\dot{\alpha} - \dot{\theta}), \tag{2.33}
$$

where μ_k is the coefficient of kinetic friction between the treads and tread assemblies and $mg/\sin\alpha$ is the normal force acting on the treads from the step edge at equilibrium. The coefficient c_T is a constant defined by the physical parameters of the system (mass, length, coefficient of kinetic friction between the chassis and tread assemblies, and gravitational acceleration). The contribution to the generalized forces can be determined from $F_q = \delta P/\delta \dot{q}$ taking special care to maintain the sign of the direction-dependent force by using the signum function

$$\frac{\delta P}{\delta \dot{q}} = \begin{pmatrix} 0 \\ -\frac{\mu_k mgr}{\sin\alpha} \cdot \operatorname{sgn}(\dot{\phi} - \dot{\alpha}) \\ \frac{\mu_k mgr}{\sin\alpha} \cdot \operatorname{sgn}(\dot{\phi} - \dot{\alpha}) - c_T \cdot \operatorname{sgn}(\dot{\alpha} - \dot{\theta}) \\ c_T \cdot \operatorname{sgn}(\dot{\alpha} - \dot{\theta}) \end{pmatrix}. \tag{2.34}$$

The matrix B in (2.32) maps τ, the control input torque vector for the motors in the chassis, to the generalized coordinates:

$$B = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix}.$$

The first element of τ represents the motor torque between the chassis and the tread sprockets and the second element represents the motor torque between the chassis and the tread assemblies. We model the torque output τ_κ from each motor linearly as (derived from (2.8))

$$\tau_\kappa = \sigma_\kappa u_\kappa - \zeta_\kappa \omega_\kappa, \quad \sigma_\kappa = \frac{\gamma_\kappa k_\kappa V}{R_\kappa}, \quad \zeta_\kappa = \frac{(\gamma_\kappa k_\kappa)^2}{R_\kappa}, \tag{2.35}$$

where σ_κ is the stall torque, u_κ is the control input (limited to $[-1, 1]$), ζ_κ is the back EMF damping coefficient of the motor, ω_κ is the speed of the motor shaft relative to the motor body, k_κ is the motor constant, γ_κ is the gear ratio of the transmission, V is the nominal battery voltage, and R_κ is the terminal resistance [50]. Substituting the motor model (2.35), we can rewrite (2.32):

$$M(q)\ddot{q} + F(q, \dot{q}) = B[\Sigma u - Z(\dot{q})] + \frac{\delta P}{\delta \dot{q}}, \tag{2.36}$$

$$\Sigma = \begin{bmatrix} \sigma_S & 0 \\ 0 & \sigma_T \end{bmatrix}, \quad Z(\dot{q}) = \begin{pmatrix} \zeta_S(\dot{\theta} - \dot{\phi}) \\ \zeta_T(\dot{\theta} - \dot{\alpha}) \end{pmatrix}.$$

We next impose a no-slip constraint between the tread sprocket and the step edge, which will be shown also achieves a coordinate reduction, via

$$w + r(\phi - \alpha) + \rho(\pi/2 - \alpha) = 0. \tag{2.37}$$

In two dimensions, this is a holonomic constraint which we can differentiate with respect to time to write in the form

$$A_1\dot{q} = 0, \quad A_1 = \begin{pmatrix} 1 & r & -(r+\rho) & 0 \end{pmatrix}.$$

We can write additional constraints depending on whether the treads are in stiction with (not moving relative to) the tread assemblies ($\dot{\phi} = \dot{\alpha}$)

$$A_2\dot{q} = 0, \quad A_2 = \begin{pmatrix} 0 & 1 & -1 & 0 \end{pmatrix},$$

or the tread assemblies are in stiction with the chassis ($\dot{\alpha} = \dot{\theta}$)

$$A_3\dot{q} = 0, \quad A_3 = \begin{pmatrix} 0 & 0 & 1 & -1 \end{pmatrix}.$$

These three constraints can be combined by stacking the row vectors to form a constraint matrix A_β. In this system, A_β is not dependent on q. We append (2.36) with the inner product of the constraint matrix A_β with λ_β, the Lagrange multiplier:

$$M(q)\ddot{q} + F(q, \dot{q}) = B[\Sigma u - Z(\dot{q})] + \frac{\delta P}{\delta \dot{q}} + A_\beta^T \lambda_\beta. \tag{2.38}$$

We assume that the no-slip constraint always holds, but are interested in the different combinations of tread/tread assembly and chassis/tread assembly stiction. This results in four possible constraint matrices

- No-slip only, $A_{NS} = A_1$
- No-slip with treads in stiction, $A_T = [A_1; A_2]$
- No-slip with chassis in stiction, $A_C = [A_1; A_3]$
- No-slip with treads and chassis in stiction, $A_{TC} = [A_1; A_2; A_3]$

We can find orthonormal bases S_β for the null spaces of A_β

$$S_{NS} = \begin{bmatrix} -r/\sqrt{r^2+1} & (r+\rho)/\sqrt{(r+\rho)^2+1} & 0 \\ 1/\sqrt{r^2+1} & 0 & 0 \\ 0 & 1/\sqrt{(r+\rho)^2+1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.39}$$

$$S_T = \begin{bmatrix} \rho/\sqrt{\rho^2+2} & 0 \\ 1/\sqrt{\rho^2+2} & 0 \\ 1/\sqrt{\rho^2+2} & 0 \\ 0 & 1 \end{bmatrix},$$

$$S_C = \begin{bmatrix} -r/\sqrt{r^2+1} & (r+\rho)/\sqrt{(r+\rho)^2+2} \\ 1/\sqrt{r^2+1} & 0 \\ 0 & 1/\sqrt{(r+\rho)^2+2} \\ 0 & 1/\sqrt{(r+\rho)^2+2} \end{bmatrix},$$

$$S_{TC} = \begin{bmatrix} \rho/\sqrt{\rho^2+3} \\ 1/\sqrt{\rho^2+3} \\ 1/\sqrt{\rho^2+3} \\ 1/\sqrt{\rho^2+3} \end{bmatrix}.$$

Given that $\dot{q}$ is in this space, we define ν_β and $\dot{\nu}_\beta$ accordingly

$$\dot{q} = S_\beta \nu_\beta, \quad \ddot{q} = S_\beta \dot{\nu}_\beta \tag{2.40}$$

since S_β are constant-valued matrices. Premultiplying by S_β^T and using (2.40), we can rewrite (2.38) as

$$S_\beta^T M(q) S_\beta \dot{\nu}_\beta + S_\beta^T F(q,\dot{q}) = S_\beta^T B[\Sigma u - Z(\dot{q})] + S_\beta^T \frac{\delta P}{\delta \dot{q}}.$$

Solving for the acceleration terms $\dot{\nu}_\beta$,

$$\dot{\nu}_\beta = [S_\beta^T M(q) S_\beta]^{-1} S_\beta^T \{B[\Sigma u - Z(\dot{q})] + \frac{\delta P}{\delta \dot{q}} - F(q,\dot{q})\}. \tag{2.41}$$

Premultiplying by S_β and using (2.40)

$$\ddot{q} = S_\beta [S_\beta^T M(q) S_\beta]^{-1} S_\beta^T \{B[\Sigma u - Z(\dot{q})] + \frac{\delta P}{\delta \dot{q}} - F(q,\dot{q})\}, \tag{2.42}$$

which is a set of second order nonlinear differential equations which can be marched forward in time by traditional means, choosing the appropriate S_β as time progresses as a function of the state, see section 3.2.3.

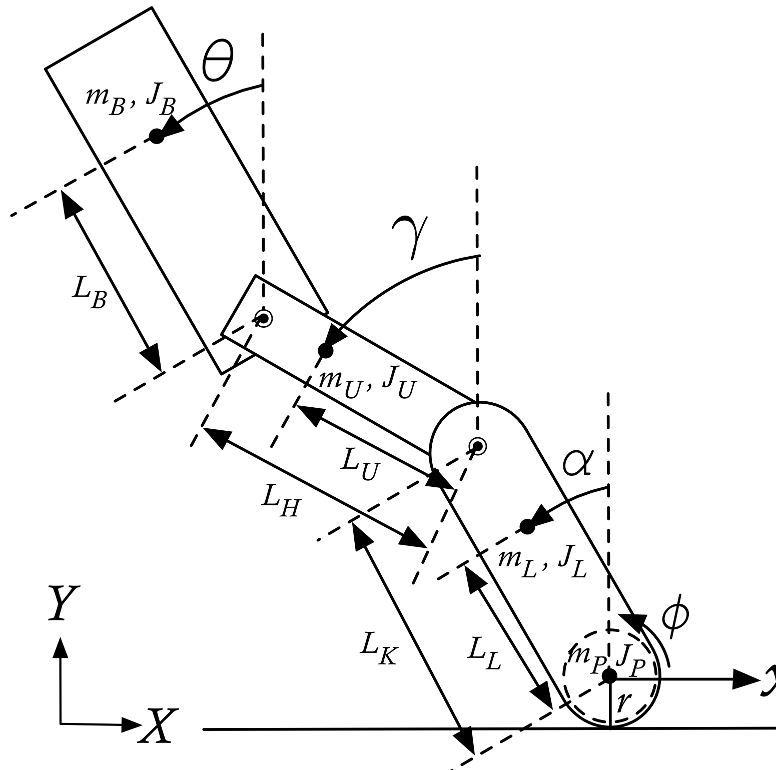


Figure 2.6: Generalized coordinates of four degree of freedom Switchbot model.

2.4.4 Switchbot

Four Degree of Freedom Model

To start, we simplify the robot to four bodies in two dimensions: the upper body, unified left and right upper legs, unified left and right pulleys that drive the treads. The generalized coordinates are labeled in Fig. 2.6, θ is the angle of the upper body from vertical, γ is the angle of the unified upper legs from vertical, α is the angle of the unified lower legs from vertical, ϕ is the angle of the unified pulleys, and x is the horizontal displacement of the pulleys in the inertial frame XY. Motors in the robot exert equal and opposite torques on the upper body and each upper leg (hip motors), and between each upper leg and lower leg (knee motors). Motors embedded in the lower legs exert torque on the pulleys (pulley motors), but have a reaction torque that is out of plane. When the robot is driving forward, the left and right out of plane reaction torques cancel each other out. The kinetic energies of the unified pulleys, unified lower legs, unified upper legs, and upper body are given, respectively, by

$$\begin{aligned}
T_P =& m_P\dot{x}^2 + J_P\dot{\phi}^2,\\
T_L =& m_L(L_L^2\dot{\alpha}^2 - 2L_L\dot{\alpha}\dot{x}\cos\alpha + \dot{x}^2) + J_L\dot{\alpha}^2,\\
T_U =& m_U[(L_K\dot{\alpha}\cos\alpha - \dot{x} + L_U\dot{\gamma}\cos\gamma)^2 + (L_K\dot{\alpha}\sin\alpha + L_U\dot{\gamma}\sin\gamma)^2] + J_U\dot{\gamma}^2,\\
T_B =& \frac{1}{2}\{m_B[(L_K\dot{\alpha}\cos\alpha - \dot{x} + L_H\dot{\gamma}\cos\gamma + L_B\dot{\theta}\cos\theta)^2\\
&+ (L_B\dot{\theta}\sin\theta + L_K\dot{\alpha}\sin\alpha + L_H\dot{\gamma}\sin\gamma)^2] + J_B\dot{\theta}^2\}.
\end{aligned}$$

Note that the factors of $\frac{1}{2}$ that may commonly be expected for kinetic energy terms are cancelled out in T_P, T_L, and T_U because the mass and inertia terms are multiplied by two to account for the left and right sides. The gravitational potential energy is given by

$$\begin{aligned}
V =& g[2m_Pr + 2m_L(r + L_L\cos\alpha) + 2m_U(r + L_K\cos\alpha + L_U\cos\gamma)\\
&+ m_B(r + L_K\cos\alpha + L_H\cos\gamma + L_B\cos\theta)].
\end{aligned}$$

We define the generalized coordinates as

$$q = \begin{pmatrix} x & \phi & \alpha & \gamma & \theta \end{pmatrix}^T.$$

The Lagrangian can be written as $\mathcal{L} = T_P + T_L + T_U + T_B - V$. By solving the Euler-Lagrange equations, we can write the equations of motion in the form

$$M(q)\ddot{q} + F(q,\dot{q}) = B\tau, \tag{2.43}$$

The (positive definite) mass matrix, $M(q)$, is given by:

$$\begin{aligned}
&M_{1,1}(q) = m_B + 2m_U + 2m_L + 2m_P, \quad M_{1,2}(q) = 0,\\
&M_{1,3}(q) = -(L_Km_B + 2L_Lm_L + 2L_Km_U)\cos\alpha,\\
&M_{1,4}(q) = -(L_Hm_B + 2L_Um_U)\cos\gamma, \quad M_{1,5}(q) = -L_Bm_B\cos\theta,\\
&M_{2,2}(q) = 2J_P, \quad M_{2,3}(q) = 0, \quad M_{2,4}(q) = 0, \quad M_{2,5}(q) = 0,\\
&M_{3,3}(q) = 2J_L + L_K^2m_B + 2L_L^2m_L + 2L_K^2m_U,\\
&M_{3,4}(q) = L_K(L_Hm_B + 2L_Um_U)\cos(\alpha - \gamma), \quad M_{3,5}(q) = L_BL_Km_B\cos(\alpha - \theta),\\
&M_{4,4}(q) = m_BL_H^2 + 2m_UL_U^2 + 2J_U, \quad M_{4,5}(q) = L_BL_Hm_B\cos(\gamma - \theta),\\
&M_{5,5}(q) = m_BL_B^2 + J_B,
\end{aligned}$$

The vector $F(q, \dot{q})$ is given by:

$$
\begin{aligned}
F_1(q,\dot{q}) =& (L_K m_B + 2L_L m_L + 2L_K m_U)\dot{\alpha}^2 \sin\alpha + (L_H m_B + 2L_U m_U)\dot{\gamma}^2 \sin\gamma \\
& + L_B m_B \dot{\theta}^2 \sin\theta, \\
F_2(q,\dot{q}) =& 0, \\
F_3(q,\dot{q}) =& (L_H m_B + 2L_U m_U) L_K \dot{\gamma}^2 \sin(\alpha-\gamma) \\
& - (L_K m_B + 2L_L m_L + 2L_K m_U) g \sin\alpha + L_B L_K m_B \dot{\theta}^2 \sin(\alpha-\theta), \\
F_4(q,\dot{q}) =& L_B L_H m_B \dot{\theta}^2 \sin(\gamma-\theta) - (L_H m_B + 2L_U m_U) g \sin\gamma \\
& - (L_H m_B + 2L_U m_U) L_K \dot{\alpha}^2 \sin(\alpha-\gamma), \\
F_5(q,\dot{q}) =& - L_B m_B [L_K \dot{\alpha}^2 \sin(\alpha-\theta) + L_H \dot{\gamma}^2 \sin(\gamma-\theta) + g\sin\theta],
\end{aligned}
$$

The matrix B in (2.43) maps τ, the control input torque vector for the motors in the chassis, to the generalized coordinates:

$$
B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}.
$$

The first element of τ represents the motor torque exerted on the pulleys, the second element is the motor torque exerted between the lower and upper legs (knee motors), and the third element is the motor torque exerted between the upper legs and the body (hip motors). We model the torque output τ_κ from each motor linearly as (derived from (2.8))

$$
\tau_\kappa = \sigma_\kappa u_\kappa - \zeta_\kappa \omega_\kappa, \quad \sigma_\kappa = \frac{\gamma_\kappa k_\kappa V}{R_\kappa}, \quad \zeta_\kappa = \frac{(\gamma_\kappa k_\kappa)^2}{R_\kappa}, \tag{2.44}
$$

where σ_κ is the stall torque, u_κ is the control input (limited to $[-1, 1]$), ζ_κ is the back EMF damping coefficient of the motor, ω_κ is the speed of the motor shaft relative to the motor body, k_κ is the motor constant, γ_κ is the gear ratio of the transmission, V is the nominal battery voltage, and R_κ is the terminal resistance

[50]. Substituting the motor model (2.44), we can rewrite (2.43):

$$M(q)\ddot{q} + F(q,\dot{q}) = B[\Sigma u - Z(\dot{q})], \tag{2.45}$$

$$\Sigma = \begin{bmatrix} 2\sigma_P & 0 & 0 \\ 0 & 2\sigma_K & 0 \\ 0 & 0 & 2\sigma_H \end{bmatrix}, \quad Z(\dot{q}) = \begin{pmatrix} 2\zeta_P(\dot{\phi} - \dot{\alpha}) \\ 2\zeta_K(\dot{\alpha} - \dot{\gamma}) \\ 2\zeta_H(\dot{\gamma} - \dot{\theta}) \end{pmatrix}.$$

The factors of 2 come from the combined left and right motors. We next impose a no-slip constraint between the tread sprocket and the step edge, which will be shown also achieves a coordinate reduction, via

$$x + r\phi = 0. \tag{2.46}$$

In two dimensions, this is a holonomic constraint which we can differentiate with respect to time to write in the form

$$A\dot{q} = 0, \quad A = \begin{pmatrix} 1 & r & 0 & 0 & 0 \end{pmatrix}.$$

In this system, A is not dependent on q. We append (2.45) with the inner product of the constraint matrix A with λ, the Lagrange multiplier:

$$M(q)\ddot{q} + F(q,\dot{q}) = B[\Sigma u - Z(\dot{q})] + A^T\lambda. \tag{2.47}$$

An orthonormal basis for the null space of A is given by

$$S = \begin{bmatrix} -r/\sqrt{r^2+1} & 0 & 0 & 0 \\ 1/\sqrt{r^2+1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.48}$$

Given that $\dot{q}$ is in this space, we define ν accordingly as

$$\dot{q} = S\nu, \quad \nu = \begin{pmatrix} \dot{\phi} & \dot{\alpha} & \dot{\gamma} & \dot{\theta} \end{pmatrix}^T. \tag{2.49}$$

Premultiplying by S^T and using (2.49) and its time derivative, we can rewrite (2.47) as

$$S^T M(q) S\dot{\nu} + S^T F(q,\dot{q}) = S^T B[\Sigma u - Z(\dot{q})]. \tag{2.50}$$

Solving for the acceleration terms $\dot{\nu}$,

$$\dot{\nu} = [S^T M(q)S]^{-1} S^T \{B[\Sigma u - Z(\dot{q})] - F(q,\dot{q})\}. \tag{2.51}$$

Premultiplying by S and using (2.49)

$$\ddot{q} = S[S^T M(q)S]^{-1} S^T \{B[\Sigma u - Z(\dot{q})] - F(q,\dot{q})\}, \tag{2.52}$$

which is a set of second order nonlinear differential equations which can be marched forward in time by traditional means.

Seven Degree of Freedom Model

In order to simulate and control maneuvers where the left and right legs are not together, we must expand the model to seven bodies in two dimensions, separating the left and right upper and lower legs and pulleys. The generalized coordinates are labeled in Fig. 2.7, we reuse coordinates from Fig. 2.6, but append them with R or L subscripts to denote the right or left component. The no-slip constraint on the right pulley is implicitly defined in this formulation $x \triangleq -r\phi_R$. The parameters (masses, lengths, inertias, stall torques, and damping coefficients) are the same as in the four degree of freedom system. The kinetic energies of the right and left pulleys, right and left lower legs, right and left upper legs, and upper body are given, respectively, by

$$\begin{aligned}
T_{PR} =& \frac{1}{2}[(m_P r^2 + J_P)\dot{\phi}_R^2], \\
T_{PL} =& \frac{1}{2}\{m_P[(L_K\dot{\alpha}_L \sin\alpha \ - L_K\dot{\alpha}_R \sin\alpha_R + L_H\dot{\gamma}_L \sin\gamma_L - L_H\dot{\gamma}_R \sin\gamma_R)^2 \\
& + (r\dot{\phi}_R - L_K\dot{\alpha}_L \cos\alpha_L + L_K\dot{\alpha}_R \cos\alpha_R - L_H\dot{\gamma}_L \cos\gamma_L + L_H\dot{\gamma}_R \cos\gamma_R)^2 \\
& + J_P\dot{\phi}_L^2]\}, \\
T_{LR} =& \frac{1}{2}\{m_L(L_L^2\dot{\alpha}_R^2 + 2\cos\alpha_R L_L\dot{\alpha}_R\dot{\phi}_R r + \dot{\phi}_R^2 r^2) + J_L\dot{\alpha}_R^2\},
\end{aligned}$$

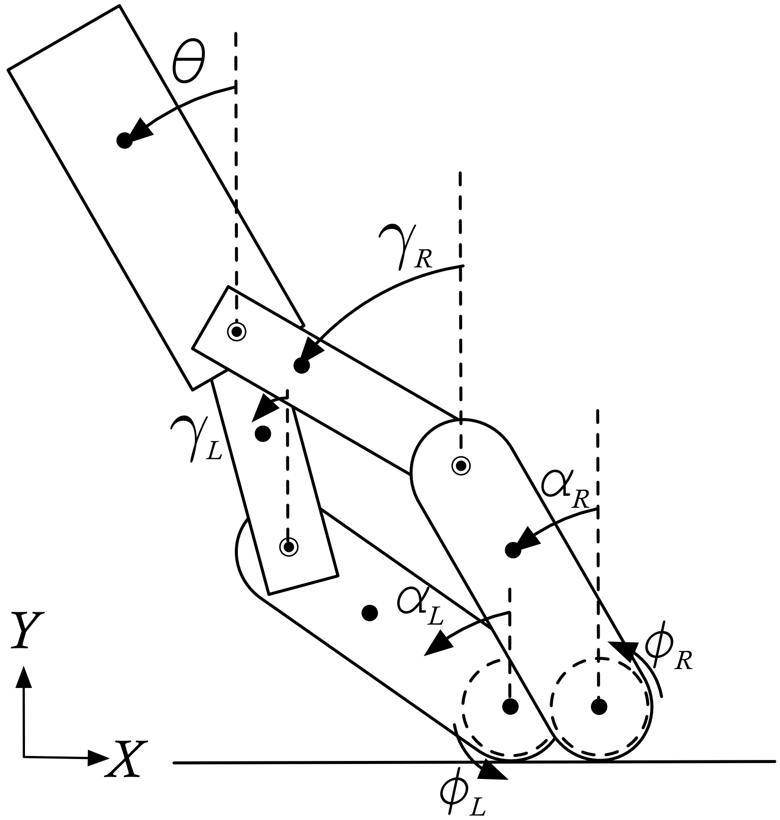


Figure 2.7: Generalized coordinates of seven degree of freedom Switchbot.

$$
\begin{aligned}
T_{LL} =& \frac{1}{2}\{m_L[(L_K\dot{\alpha}_R\sin\alpha_R - (L_K - L_L)\dot{\alpha}_L\sin\alpha_L - L_H\dot{\gamma}_L\sin\gamma_L + L_H\dot{\gamma}_R\sin\gamma_R)^2 \\
&+ (r\dot{\phi}_R + L_K\dot{\alpha}_R\cos\alpha_R - (L_K - L_L)\dot{\alpha}_L\cos\alpha_L - L_H\dot{\gamma}_L\cos\gamma_L + L_H\dot{\gamma}_R\cos\gamma_R)^2] \\
&+ J_L\dot{\alpha}_L^2\}, \\
T_{UR} =& \frac{1}{2}\{m_U[(L_K\dot{\alpha}_R\sin\alpha_R + L_U\dot{\gamma}_R\sin\gamma_R)^2 + (r\dot{\phi}_R + L_K\dot{\alpha}_R\cos\alpha_R + L_U\dot{\gamma}_R\cos\gamma_R)^2] \\
&+ J_U\dot{\gamma}_R^2\}, \\
T_{UL} =& \frac{1}{2}\{m_U[(L_K\dot{\alpha}_R\sin\alpha_R + L_H\dot{\gamma}_R\sin\gamma_R - (Lh - Lu)\dot{\gamma}_L\sin\gamma_L)^2 \\
&+ (r\dot{\phi}_R + L_K\dot{\alpha}_R\cos\alpha_R + L_H\dot{\gamma}_R\cos\gamma_R - (L_H - L_U)\dot{\gamma}_L\cos\gamma_L)^2] + J_U\dot{\gamma}_L^2\}, \\
T_B =& \frac{1}{2}\{m_B[(L_B\dot{\theta}\sin\theta + L_K\dot{\alpha}_R\sin\alpha_R + L_H\dot{\gamma}_R\sin\gamma_R)^2 \\
&+ (r\dot{\phi}_R + L_K\dot{\alpha}_R\cos\alpha_R + L_H\dot{\gamma}_R\cos\gamma_R + L_B\dot{\theta}\cos\theta)^2] + J_B\dot{\theta}^2\}.
\end{aligned}
$$

The gravitational potential energy is given by

$$\begin{aligned}
V =& g\{m_B(r + L_K \cos\alpha_R + L_H \cos\gamma_R + L_B \cos\theta) \\
&+ m_L[2r + L_K \cos\alpha_R + L_L \cos\alpha_R - L_H \cos\gamma_L + L_H \cos\gamma_R - (L_K - L_L)\cos\alpha_L] \\
&+ m_P(2r - L_K \cos\alpha_L + L_K \cos\alpha_R - L_H \cos\gamma_L + L_H \cos\gamma_R) \\
&+ m_U([2r + 2L_K \cos\alpha_R + L_H \cos\gamma_R + L_U \cos\gamma_R - (L_H - L_U)\cos\gamma_L]\}.
\end{aligned}$$

We define the generalized coordinates as

$$q = \begin{pmatrix} \phi_R & \phi_L & \alpha_R & \alpha_L & \gamma_R & \gamma_L & \theta \end{pmatrix}^T.$$

The Lagrangian can be written as $\mathcal{L} = T_{PR}+T_{PL}+T_{LR}+T_{LL}+T_{UR}+T_{UL}+T_B-V$. By solving the Euler-Lagrange equations, we can write the equations of motion in the form

$$M(q)\ddot{q} + F(q,\dot{q}) = B\tau, \tag{2.53}$$

The (positive definite) mass matrix, $M(q)$, is given by:

$$\begin{aligned}
&M_{1,1}(q) = J_P + (m_B + 2m_U + 2m_L + 2m_P)r^2, \quad M_{1,2}(q) = 0, \\
&M_{1,3}(q) = r(L_K m_B + L_K m_L + L_L m_L + L_K m_P + 2L_K m_U)\cos\alpha_R, \\
&M_{1,4}(q) = -r(L_K m_L - L_L m_L + L_K m_P)\cos\alpha_L, \\
&M_{1,5}(q) = r(L_H m_B + L_H m_L + L_H m_P + L_H m_U + L_U m_U)\cos\gamma_R, \\
&M_{1,6}(q) = -r(L_H m_L + L_H m_P + L_H m_U - L_U m_U)\cos\gamma_L, \\
&M_{1,7}(q) = r L_B m_B \cos\theta, \\
&M_{2,2}(q) = J_P, \quad M_{2,3}(q) = 0, \quad M_{2,4}(q) = 0, \\
&M_{2,5}(q) = 0, \quad M_{2,6}(q) = 0, \quad M_{2,7}(q) = 0, \\
&M_{3,3}(q) = J_L + L_K^2 m_B + L_K^2 m_L + L_L^2 m_L + L_K^2 m_P + 2L_K^2 m_U, \\
&M_{3,4}(q) = -L_K \cos(\alpha_L - \alpha_R)(L_K m_L - L_L m_L + L_K m_P), \\
&M_{3,5}(q) = L_K \cos(\alpha_R - \gamma_R)(L_H m_B + L_H m_L + L_H m_P + L_H m_U + L_U m_U), \\
&M_{3,6}(q) = -L_K \cos(\alpha_R - \gamma_L)(L_H m_L + L_H m_P + L_H m_U - L_U m_U), \\
&M_{3,7}(q) = L_B L_K m_B \cos(\alpha_R - \theta),
\end{aligned}$$

$$M_{4,4}(q) = J_L + L_K^2 m_P + m_L(L_K - L_L)^2,$$
$$M_{4,5}(q) = -L_H \cos(\alpha_L - \gamma_R)(L_K m_L - L_L m_L + L_K m_P),$$
$$M_{4,6}(q) = L_H \cos(\alpha_L - \gamma_L)(L_K m_L - L_L m_L + L_K m_P), \quad M_{4,7}(q) = 0,$$
$$M_{5,5}(q) = J_U + L_H^2 m_B + L_H^2 m_L + L_H^2 m_P + L_H^2 m_U + L_U^2 m_U,$$
$$M_{5,6}(q) = -L_H \cos(\gamma_L - \gamma_R)(L_H m_L + L_H m_P + L_H m_U - L_U m_U),$$
$$M_{5,7}(q) = L_B L_H m_B \cos(\gamma_R - \theta),$$
$$M_{6,6}(q) = J_U + L_H^2 m_L + L_H^2 m_P + L_H^2 m_U + L_U^2 m_U - 2L_H L_U m_U, \quad M_{6,7}(q) = 0,$$
$$M_{7,7}(q) = m_B L_B^2 + J_B.$$

The vector $F(q, \dot{q})$ is given by:

$$\begin{aligned}
F_1(q,\dot{q}) =& r \sin\alpha_L (L_K m_L - L_L m_L + L_K m_P)\dot{\alpha}_L^2 \\
&- r \sin\alpha_R (L_K m_B + L_K m_L + L_L m_L + L_K m_P + 2L_K m_U)\dot{\alpha}_R^2 \\
&+ r \sin\gamma_L (L_H m_L + L_H m_P + L_H m_U - L_U m_U)\dot{\gamma}_L^2 \\
&- r \sin\gamma_R (L_H m_B + L_H m_L + L_H m_P + L_H m_U + L_U m_U)\dot{\gamma}_R^2 \\
&- L_B m_B r \sin\theta \dot{\theta}^2, \\
F_2(q,\dot{q}) =& 0, \\
F_3(q,\dot{q}) =& L_K^2 \dot{\alpha}_L^2 m_L \sin(\alpha_L - \alpha_R) + L_K^2 \dot{\alpha}_L^2 m_P \sin(\alpha_L - \alpha_R) - L_K g m_B \sin\alpha_R \\
&- L_K g m_L \sin\alpha_R - L_L g m_L \sin\alpha_R - L_K g m_P \sin\alpha_R - 2L_K g m_U \sin\alpha_R \\
&- L_K L_L \dot{\alpha}_L^2 m_L \sin(\alpha_L - \alpha_R) + L_H L_K \dot{\gamma}_R^2 m_B \sin(\alpha_R - \gamma_R) \\
&- L_H L_K \dot{\gamma}_L^2 m_L \sin(\alpha_R - \gamma_L) + L_H L_K \dot{\gamma}_R^2 m_L \sin(\alpha_R - \gamma_R) \\
&- L_H L_K \dot{\gamma}_L^2 m_P \sin(\alpha_R - \gamma_L) + L_H L_K \dot{\gamma}_R^2 m_P \sin(\alpha_R - \gamma_R) \\
&- L_H L_K \dot{\gamma}_L^2 m_U \sin(\alpha_R - \gamma_L) + L_H L_K \dot{\gamma}_R^2 m_U \sin(\alpha_R - \gamma_R) \\
&+ L_K L_U \dot{\gamma}_L^2 m_U \sin(\alpha_R - \gamma_L) + L_K L_U \dot{\gamma}_R^2 m_U \sin(\alpha_R - \gamma_R) \\
&+ L_B L_K m_B \dot{\theta}^2 \sin(\alpha_R - \theta), \\
F_4(q,\dot{q}) =& (L_K m_L - L_L m_L + L_K m_P)(-L_K \sin(\alpha_L - \alpha_R)\dot{\alpha}_R^2 + L_H \sin(\alpha_L - \gamma_L)\dot{\gamma}_L^2 \\
&- L_H \sin(\alpha_L - \gamma_R)\dot{\gamma}_R^2 + g \sin\alpha_L),
\end{aligned}$$

$$
\begin{aligned}
F_5(q,\dot q) =& L_H^2\dot\gamma_L^2 m_L\sin(\gamma_L-\gamma_R) + L_H^2\dot\gamma_L^2 m_P\sin(\gamma_L-\gamma_R) + L_H^2\dot\gamma_L^2 m_U\sin(\gamma_L-\gamma_R)\\
&- L_H g m_B\sin\gamma_R - L_H g m_L\sin\gamma_R - L_H g m_P\sin\gamma_R - L_H g m_U\sin\gamma_R\\
&- L_U g m_U\sin\gamma_R - L_H L_K\dot\alpha_R^2 m_B\sin(\alpha_R-\gamma_R) + L_H L_K\dot\alpha_L^2 m_L\sin(\alpha_L-\gamma_R)\\
&- L_H L_K\dot\alpha_R^2 m_L\sin(\alpha_R-\gamma_R) - L_H L_L\dot\alpha_L^2 m_L\sin(\alpha_L-\gamma_R)\\
&+ L_H L_K\dot\alpha_L^2 m_P\sin(\alpha_L-\gamma_R) - L_H L_K\dot\alpha_R^2 m_P\sin(\alpha_R-\gamma_R)\\
&- L_H L_K\dot\alpha_R^2 m_U\sin(\alpha_R-\gamma_R) - L_K L_U\dot\alpha_R^2 m_U\sin(\alpha_R-\gamma_R)\\
&- L_H L_U\dot\gamma_L^2 m_U\sin(\gamma_L-\gamma_R) + L_B L_H m_B\dot\theta^2\sin(\gamma_R-\theta),\\
F_6(q,\dot q) =& L_H g m_L\sin\gamma_L - L_H^2\dot\gamma_R^2 m_P\sin(\gamma_L-\gamma_R) - L_H^2\dot\gamma_R^2 m_U\sin(\gamma_L-\gamma_R)\\
&- L_H^2\dot\gamma_R^2 m_L\sin(\gamma_L-\gamma_R) + L_H g m_P\sin\gamma_L + L_H g m_U\sin\gamma_L\\
&- L_U g m_U\sin\gamma_L - L_H L_K\dot\alpha_L^2 m_L\sin(\alpha_L-\gamma_L) + L_H L_K\dot\alpha_R^2 m_L\sin(\alpha_R-\gamma_L)\\
&+ L_H L_L\dot\alpha_L^2 m_L\sin(\alpha_L-\gamma_L) - L_H L_K\dot\alpha_L^2 m_P\sin(\alpha_L-\gamma_L)\\
&+ L_H L_K\dot\alpha_R^2 m_P\sin(\alpha_R-\gamma_L) + L_H L_K\dot\alpha_R^2 m_U\sin(\alpha_R-\gamma_L)\\
&- L_K L_U\dot\alpha_R^2 m_U\sin(\alpha_R-\gamma_L) + L_H L_U\dot\gamma_R^2 m_U\sin(\gamma_L-\gamma_R),\\
F_7(q,\dot q) =& - L_B m_B(L_K\sin(\alpha_R-\theta)\dot\alpha_R^2 + L_H\sin(\gamma_R-\theta)\dot\gamma_R^2 + g\sin\theta).
\end{aligned}
$$

The matrix B in (2.53) maps τ, the control input torque vector for the motors in the chassis, to the generalized coordinates:

$$
B = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & -1 & 0 & 1 & 0\\
0 & 0 & 0 & -1 & 0 & 1\\
0 & 0 & 0 & 0 & -1 & -1
\end{bmatrix}.
$$

The first element of τ represents the motor torque exerted on the right pulley, the second the left pulley, the third the between the right lower and upper leg, the fourth between the left lower and upper leg, the fifth between the right upper leg and the body, and the sixth between the left upper leg and the body. We model the torque output τ_κ from each motor linearly as in (2.44) (derived from (2.8)) and we can rewrite (2.53):

$$M(q)\ddot{q} + F(q,\dot{q}) = B[\Sigma u - Z(\dot{q})], \tag{2.54}$$

$$\Sigma = \begin{bmatrix} \sigma_P & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_P & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_K & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_K & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_H & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_H \end{bmatrix}, \quad Z(\dot{q}) = \begin{pmatrix} \zeta_P(\dot{\phi}_R - \dot{\alpha}_R) \\ \zeta_P(\dot{\phi}_L - \dot{\alpha}_L) \\ \zeta_K(\dot{\alpha}_R - \dot{\gamma}_R) \\ \zeta_K(\dot{\alpha}_L - \dot{\gamma}_L) \\ \zeta_H(\dot{\gamma}_R - \dot{\theta}) \\ \zeta_H(\dot{\gamma}_L - \dot{\theta}) \end{pmatrix}.$$

At this point we need to constrain the system such that the left pulley is in contact with the ground and does not slip. The method of imposing constraints on the Lagrangian dynamics heretofore used is inadequate because the constraint matrices A_β and their orthonormal null space bases S_β are functions of q. Instead, we use the method developed by Udwadia and Kalaba [51]. Constraints are written in the form

$$A(q,\dot{q})\ddot{q} = b(q,\dot{q}) \tag{2.55}$$

By inspection, when the left pulley is in contact with the ground

$$L_K(\cos\alpha_R - \cos\alpha_L) + L_H(\cos\gamma_R - \cos\gamma_L) = 0$$

which can be differentiated twice with respect to time to be written in the form of (2.55)

$$A_y(q,\dot{q}) = \begin{pmatrix} 0 & 0 & -L_K\sin\alpha_R & L_K\sin\alpha_L & -L_H\sin\gamma_R & L_H\sin\gamma_L & 0 \end{pmatrix} \tag{2.56}$$

$$b_y(q,\dot{q}) = L_K\dot{\alpha}_R^2\cos\alpha_R - L_K\dot{\alpha}_L^2\cos\alpha_L + L_H\dot{\gamma}_R^2\cos\gamma_R - L_G\dot{\gamma}_L^2\cos\gamma_L.$$

Assuming that the right and left pulleys are collocated when $q = 0_{7\times1}$, then in order for the left pulley to not slip on the ground

$$-r\phi_L = -r\phi_R - L_K\sin\alpha_R - L_H\sin\gamma_R + L_H\sin\gamma_L + L_K\sin\alpha_L \tag{2.57}$$

which can again be differentiated twice with respect to time to be written the form of (2.55)

$$A_x(q,\dot{q}) = \begin{pmatrix} -r & r & -L_K\cos\alpha_R & L_K\cos\alpha_L & -L_H\cos\gamma_R & L_H\cos\gamma_L & 0 \end{pmatrix} \tag{2.58}$$

$$b_x(q,\dot{q}) = -\,L_K\dot{\alpha}_R^2\sin\alpha_R - L_H\dot{\gamma}_R^2\sin\gamma_R + L_H\dot{\gamma}_L^2\sin\gamma_L + L + K\dot{\alpha}_L^2\sin\alpha_L.$$

Concatenating (2.56) and (2.58)

$$A(q,\dot{q}) = \begin{bmatrix} A_y(q,\dot{q}) \\ A_x(q,\dot{q}) \end{bmatrix}, \quad b(q,\dot{q}) = \begin{pmatrix} b_y(q,\dot{q}) \\ b_x(q,\dot{q}) \end{pmatrix}. \tag{2.59}$$

We append the constraints (2.59) to (2.53), moving $F(q,\dot{q})$ to the right hand side, with

$$M\ddot{q} = B(\Sigma u - Z) - F + MA_M^+\{b - AM^{-1}[B(\Sigma u - Z) - F]\}, \tag{2.60}$$

dropping the dependence of M, F, A, A_M^+, Z, and b on q and $\dot{q}$ for clarity. A_M^+ is the generalized Moore-Penrose M-inverse of the constraint matrix A, which is related to the Moore-Penrose pseudoinverse by

$$(AM^{-1/2})^+ = M^{1/2}A_M^+$$

with $AM^{-1}A^T$ invertible, we can rewrite (2.60) as

$$M\ddot{q} = B(\Sigma u - Z) - F + A^T(AM^{-1}A^T)^{-1}\{b - AM^{-1}[B(\Sigma u - Z) - F]\},$$

and premultiplying both sides by M^{-1}

$$\ddot{q} = M^{-1}\left\{B(\Sigma u - Z) - F + A^T(AM^{-1}A^T)^{-1}\{b - AM^{-1}[B(\Sigma u - Z) - F]\}\right\}, \tag{2.61}$$

we arrive at a set of second order nonlinear differential equations which can be marched forward in time by traditional means.

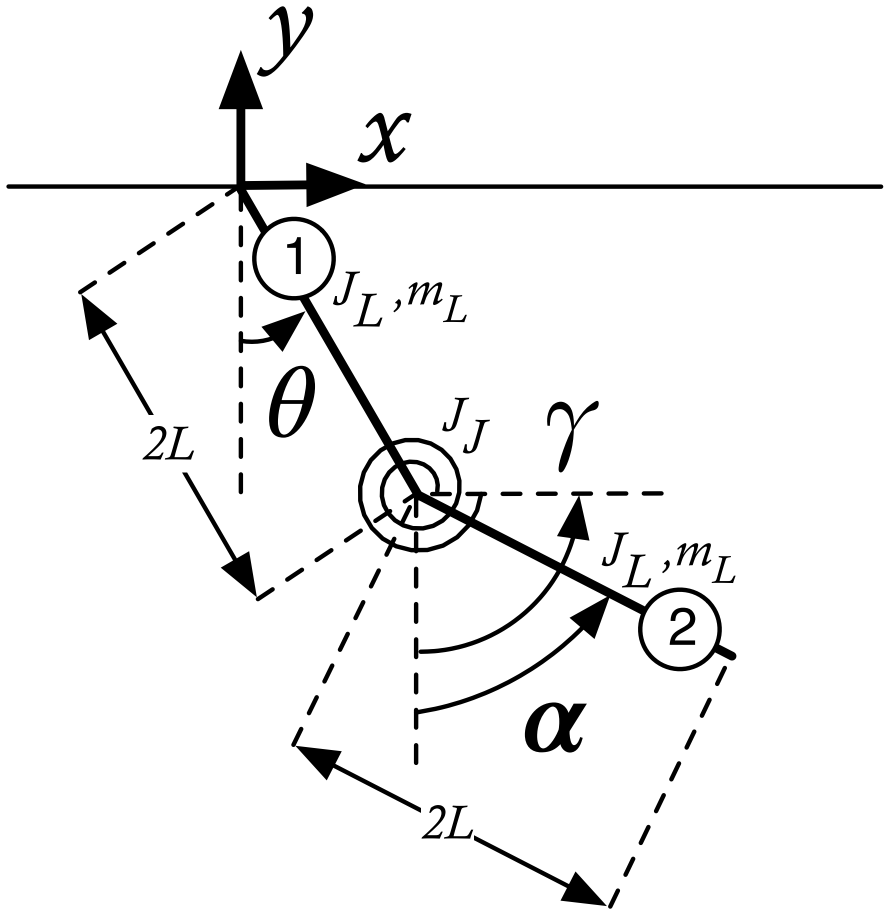


Figure 2.8: Generalized coordinates of SkySweeper.

2.5 SkySweeper

We simplify the model to three bodies in two dimensions. The generalized coordinates are defined in Fig. 2.8, where θ and α are the angles of the first and second links, respectively, from vertical, γ is the rotation angle of the SEA shaft from vertical, and x, y is the position of the clamping end of the first link. The SEA spring deflection is then given by $\alpha - \gamma$ and the link separation angle is $\theta + \pi - \alpha$. The mass is assumed to be distributed uniformly along both links, each with length $2L$ and mass m_L, the SEA shaft has rotational inertia J_J, but negligible mass compared to each link. In this model, the cable is assumed to be horizontal and rigid.

The kinetic energies of the first link, joint, and second link are given respectively by:

$$T_1 = \frac{1}{2}\Big\{m_L[(\dot{x} + L\dot{\theta}\cos\theta)^2 + (\dot{y} + L\dot{\theta}\sin\theta)^2] + J_L\dot{\theta}^2\Big\},$$
$$T_J = \frac{1}{2}\Big\{J_J\dot{\gamma}^2\Big\},$$
$$T_2 = \frac{1}{2}\Big\{m_L[(\dot{x} + 2L\dot{\theta}\cos\theta + L\dot{\alpha}\cos\alpha)^2 + (\dot{y} + 2L\dot{\theta}\sin\theta + L\dot{\alpha}\sin\alpha)^2] + J_L\dot{\alpha}^2\Big\}.$$

The spring and gravitational potential energy is given by:

$$V = \frac{1}{2}k(\alpha - \gamma)^2 + m_L g[2y - L(3\cos\theta + \cos\alpha)].$$

We define q, the vector of n generalized coordinates, as

$$q = \begin{pmatrix} x & y & \theta & \gamma & \alpha \end{pmatrix}^T.$$

The Lagrangian can be written as $\mathcal{L} = T_1 + T_J + T_2 - V$. By solving the Euler-Lagrange equations, we can write the equations of motion in the form

$$M(q)\ddot{q} + F(q, \dot{q}) = B\tau, \tag{2.62}$$

The (positive definite) mass matrix, $M(q)$, is given by:

$$\begin{aligned}
M_{1,1}(q) &= 2m_L & M_{2,5}(q) &= m_L L\sin\alpha \\
M_{1,2}(q) &= 0 & M_{3,3}(q) &= 5m_L L^2 + J_L \\
M_{1,3}(q) &= 3m_L L\cos\theta & M_{3,4}(q) &= 0 \\
M_{1,4}(q) &= 0 & M_{3,5}(q) &= 2m_L L^2\cos(\alpha - \theta) \\
M_{1,5}(q) &= m_L L\cos\alpha & M_{4,4}(q) &= J_J \\
M_{2,2}(q) &= 2m_L & M_{4,5}(q) &= 0 \\
M_{2,3}(q) &= 3m_L L\sin\theta & M_{5,5}(q) &= m_L L^2 + J_L \\
M_{2,4}(q) &= 0 & &
\end{aligned}$$

and the vector $F(q, \dot{q})$ is given by:

$$F(q, \dot{q}) = \begin{pmatrix} -m_L L[3\dot{\theta}^2\sin\theta + \dot{\alpha}^2\sin\alpha] \\ m_L[L(3\dot{\theta}^2\cos\theta + \dot{\alpha}^2\cos\alpha) + 2g] \\ m_L L[3g\sin\theta - 2L\dot{\alpha}^2\sin(\alpha - \theta)] \\ -k(\alpha - \gamma) \\ m_L L[2L\dot{\theta}^2\sin(\alpha - \theta) + g\sin\alpha] + k(\alpha - \gamma) \end{pmatrix}.$$

The vector B maps τ, the control input torque for the motor in the elbow SEA, to the generalized coordinates:

$$B = \begin{pmatrix} 0 & 0 & -1 & 1 & 0 \end{pmatrix}^T .$$

Depending on the positions of the clamps, there may be holonomic and/or non-holonomic constraints on the system. We use the method of undetermined Lagrange multipliers, similarly to [1] and [11], to apply both holonomic and non-holonomic constraints. We first write the constraints in Pfaffian form: $A(q)\dot{q} = 0$ and append (2.62) with the inner product of the constraint matrix $A(q)$ with the Lagrange multiplier λ:

$$M(q)\ddot{q} + F(q, \dot{q}) = B\tau + A(q)^T \lambda. \tag{2.63}$$

Then we solve for $S(q)$, the orthonormal basis for the null space of $A(q)$. Given that $\dot{q}$ is in this space, we define the reduced coordinate vector ν accordingly as

$$\dot{q} = S(q)\nu. \tag{2.64}$$

Premultiplying by $S(q)^T$ and using (2.64), we can rewrite (2.63):

$$S(q)^T M(q) S(q)\dot{\nu} + S(q)^T F(q, \dot{q}) = S(q)^T B\tau. \tag{2.65}$$

The acceleration of the full coordinate vector, $\ddot{q}$, can be recovered using (2.64) and its time derivative.

The choice of $S(q)$ depends on the positions of the two clamps. If both clamps are open, there are no constraints on the system (and the robot will simply fall): $A_{OO}(q) = ()$ and $S_{OO}(q) = I_{5x5}$. If clamp one is in rollling position and clamp two is open, there is a holonomic constraint $y = 0$, which can be expressed $A_{RO}(q) = (0\ 1\ 0\ 0\ 0)$ with the corresponding orthonormal null space basis:

$$S_{RO}(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

If clamp one is in pivoting position and clamp two is open, there is the same holonomic constraint as above plus the non-holonomic constraint $\dot{x} = 0$. Concatenating both constraints yields the matrices:

$$A_{PO}(q) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, S_{PO}(q) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In the case where clamp one is in pivoting position and clamp two is in rolling position, there is an additional holonomic constraint that the height of the second clamp is zero: $y - 2L(\cos\theta + \cos\alpha) = 0$, accordingly:

$$A_{PR}(q) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2L\sin\theta & 0 & 2L\sin\alpha \end{bmatrix}, S_{PR}(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{-\sin\alpha}{\sin\theta\sqrt{\sin^2\alpha/\sin^2\theta+1}} \\ 1 & 0 \\ 0 & \frac{1}{\sqrt{\sin^2\alpha/\sin^2\theta+1}} \end{bmatrix}.$$

When both clamps are in the rolling position, we can remove the $\dot{x} = 0$ constraint, which gives:

$$A_{RR}(q) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2L\sin\theta & 0 & 2L\sin\alpha \end{bmatrix},$$

$$S_{RR}(q) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{-\sin\alpha}{\sin\theta\sqrt{\sin^2\alpha/\sin^2\theta+1}} \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\sqrt{\sin^2\alpha/\sin^2\theta+1}} \end{bmatrix}.$$

If both clamps are in the pivoting position, we add an additional non-holonomic constraint to $A_{PR}(q)$, that the horizontal speed at the second clamp is zero: $\dot{x} +$

$2L(\dot{\theta}\cos\theta + \dot{\alpha}\cos\alpha) = 0$ which fully constrains the system except for γ:

$$A_{PP}(q) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2L\sin\theta & 0 & 2L\sin\alpha \\ 1 & 0 & 2L\cos\theta & 0 & 2L\cos\alpha \end{bmatrix}, \; S_{PP}(q) \quad = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T$$

The additional three possible clamping configurations can be modeled by an appropriate coordinate transformation and using the above constraint matrices, essentially mirroring the two links. $M(q)$, $F(q, \dot{q})$, and B also change slightly with the coordinate transformation to account for the asymmetry of the SEA.

Finally, we model the torque output τ from the brushed direct current motor linearly as (derived from (2.8)):

$$\tau = \sigma u - \zeta\omega, \quad \sigma = \frac{\Gamma k_M V}{r}, \quad \zeta = \frac{\Gamma k_M^2}{r}, \tag{2.66}$$

where σ is the stall torque, u is the control input (limited to $[-1, 1]$), ζ is the back EMF damping coefficient of the motor, ω is the speed of the motor shaft relative to the motor body (in this case, $\dot{\gamma} - \dot{\theta}$), Γ is the gear ratio of the transmission, k_M is the motor constant, V is the supply voltage, and r is the terminal resistance [50]. Substituting the motor model (2.66) and moving the $F(q, \dot{q})$ term to the right hand side, we can rewrite (2.65):

$$S(q)^T M(q) S(q)\dot{\nu} = S(q)^T\{B[\sigma u - \zeta(\dot{\gamma} - \dot{\theta})] - F(q, \dot{q})\}. \tag{2.67}$$

Formulating the Lagrangian dynamics does not yield any insight into the internal forces of the system, such as the normal force between the clamp and the wire. Since knowledge of such forces is useful for design purposes (e.g. how much force the clamp will have to withstand without opening during a dynamic maneuver and what coefficient of static friction is required to prevent slipping), we can formulate equations for these forces as a function of the state variables (the generalized coordinates and their time derivatives). If clamp one is closed and clamp two is open, the reaction force along clamp one, and its x and y components, are given by:

$$R = m_L\{g[\cos\theta + \cos\alpha\cos(\theta - \alpha)] + L[\dot{\theta}^2 + \dot{\alpha}^2\cos(\theta - \alpha)]\} \tag{2.68}$$

$$R_x = R\sin\theta, \quad R_y = R\cos\theta.$$

Note that by design, the clamp cannot exert a reaction torque and may only exert a horizontal reaction force when in the pivoting position.

2.6 Backlash

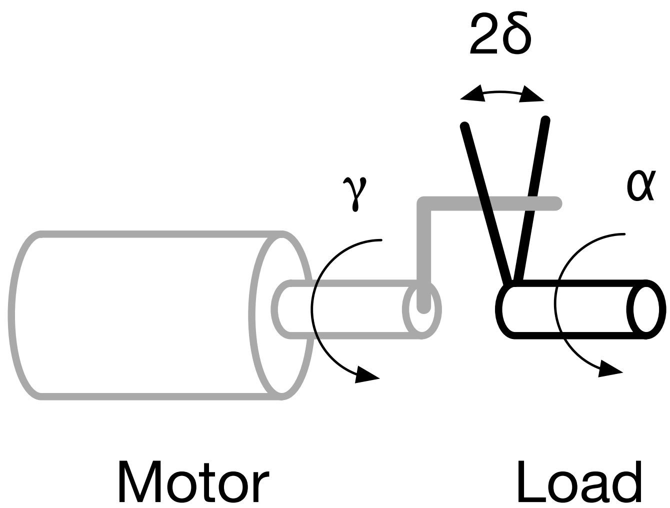


Figure 2.9: Illustration of backlash between motor and load.

Backlash is generally present in robotic systems, but frequently ignored. The nonlinear nature of backlash can make system modeling and control design much more complicated. In some cases, backlash can be reasonably ignored, if it is sufficiently small or if system is preloaded (e.g. by gravity), but it can cause instability in some systems. Backlash can be modeled as a switched system with two different constraint matrices (2.14), depending on whether or not the system is in backlash. An additional degree of freedom is introduced such that the motor shaft angle, γ, is not the same as the output shaft angle, α, the maximum gap between them is defined as 2δ, see Fig. 2.9. When the motor shaft engages the output shaft and $\dot{\gamma} = \dot{\alpha}$, the constraint matrix in the Lagrangian mathematically enforces the constraint. With the system defined thusly and two different S matrices (one perhaps being the identity matrix in the case of no other constraints), we only need to determine conditions between switching between the constraint matrices (coupling and uncoupling conditions).

- If the system is uncoupled, it will become coupled when the absolute value of the relative position is greater than or equal to δ and the relative velocity has the same sign as the relative position.
- If the system is coupled, it will become uncoupled when the sign of the relative acceleration, calculated from the uncoupled dynamics, is the opposite of the sign of the relative position.

Example code as may be included in the plant model of a simulation is below:

```
gap = x(2) - x(3);
relVel = x(6) - x(7);
relAccel = xdot(6) - xdot(7); % uncoupled relative acceleration

if coupledOld == 0 % uncoupled
    % if gap is >= backlash and relative speed is same sign as gap, couple
    if (abs(gap) >= delta) && (sign(relVel) == sign(gap)) % positive or negative engagement
        coupled = sign(gap);
        resetVel = ( J2*x(6) + Jg*x(7) )/(Jg+J2);
        resetPos = [x(1); x(3)+sign(gap)*delta; x(3); x(4); x(5); resetVel; resetVel; x(8);];
    else % stay uncoupled
        coupled = 0;
    end
else    % coupled
    % if relative acceleration is opposite sign as gap, uncouple
    if sign(relAccel) == -sign(gap) % accelerating to open gap, uncouple
        coupled = 0;
    else % stay coupled
        coupled = coupledOld;
    end
end
```

Much prior research has been done on the topic of backlash (see [52] for a survey paper), but to the author's knowledge, the use of different constraint matrices in a switched system with Lagrangian dynamics and the relative acceleration uncoupling condition is novel.

2.7 Acknowledgements

Part of this chapter concerning Switchblade has appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning SkySweeper has appeared in print: N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2339-2344, 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the DC motor dynamometer has appeared in print: N. Morozovsky, R. Moroto, T. Bewley, "RAPID: an inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the perching maneuver has been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Control

3.1 Mobile Inverted Pendulum

We can again treat the design of controllers for myMIP (Fig. 1.2a) and eyeFling (Fig. 1.2b) as one, since the only relevant difference between the two is physical parameters. Given the previously defined constraints (2.21), both γ and x are superfluous coordinates, so we can define a reduced coordinated vector $q_r = (\phi \quad \theta)^T$. We also define $\underset{\sim}{S}$ as the bottom two rows of (2.22), then $\dot{q}_r = \underset{\sim}{S}\nu$ and $\ddot{q}_r = \underset{\sim}{S}\dot{\nu}$. We finally define a full state vector x and see that this nonlinear system is affine in the input

$$x = \begin{pmatrix} q_r \\ \dot{q}_r \end{pmatrix}, \quad \dot{x} = f(x) + g(x)u,$$

$$f(x) = \begin{pmatrix} \dot{q}_r \\ \underset{\sim}{S}[S^T M(q) S]^{-1} S^T[-B\zeta(\dot{\phi} - \dot{\theta}) - F(q, \dot{q})] \end{pmatrix},$$

$$g(x) = \begin{pmatrix} 0_{2\times 1} \\ \underset{\sim}{S}[S^T M(q) S]^{-1} S^T B\sigma \end{pmatrix}.$$

To design a controller to stabilize the unstable equilibrium at the origin (balancing upright), we linearize the system about the origin

$$\dot{x} = \mathrm{A}x + \mathrm{B}u,\ \mathrm{A} = \frac{\delta f}{\delta x}\Big|_{x=0_{4\times 1}},\ \mathrm{B} = g(x)\Big|_{x=0_{4\times 1}}.$$

Since the controller will be implemented with digital electronics, we next discretize the system with a step size $h = 0.01$ seconds.

$$
\begin{aligned}
x_{k+1} &= F x_k + G(u_k), \\
F &= \Phi(h), \quad \Phi(\tau) = \mathbf{e}^{\mathrm{A}\tau}, \\
G &= \Theta(h), \quad \Theta(\tau) = \int_0^{\tau} \mathbf{e}^{\mathrm{A}\eta} \mathrm{B} d\eta.
\end{aligned}
$$

The linear quadratic regulator (LQR) method is used to create a controller, we define weighting matrices using Bryson's method [53]. The values of Q_C and R_C, the continuous time weighting matrices, will vary slightly for myMIP and eyeFling, though we choose $N_C = 0_{4\times 1}$ for both. The continuous time weighting matrices are converted to discrete time by

$$
\begin{aligned}
Q_D &= \int_0^h \Phi^T(\tau) Q_C \Phi(\tau) d\tau, \\
R_D &= \int_0^h \Theta^T(\tau) Q_C \Theta(\tau) + R_C d\tau, \\
N_D &= \int_0^h \Phi^T(\tau) Q_C \Theta(\tau) d\tau.
\end{aligned}
$$

A discrete state feedback matrix K is found using the discrete-time LQR method and the control law takes the form $u = Kx$. See App. B for the programmatic linearization and discretization of the system and the calculation of the discrete time state feedback matrix.

3.2 Switchblade

3.2.1 V Balancing

Equilibrium Manifold

We wish to stabilize the unstable equilibrium manifold with the center of mass directly over the contact point. By statics, the equilibrium condition is

$$
m_C L_C \sin\theta^* + m_T L_T \sin\alpha^* = 0.
$$

This expression can equivalently be found by examining the dynamics and setting the velocity and acceleration terms to zero. Given the values of the parameters m_C, m_T, L_C, and L_T of the developed prototype (see Sec. 5.5), it is possible to solve for α^* and θ^* only in a limited range. We will design a controller for the equilibrium point at $\alpha^* = \theta^* = 0$.

Linearization and LQR

We begin by linearizing the system about the desired operating point, the unstable equilibrium with all states and control inputs equal to zero.

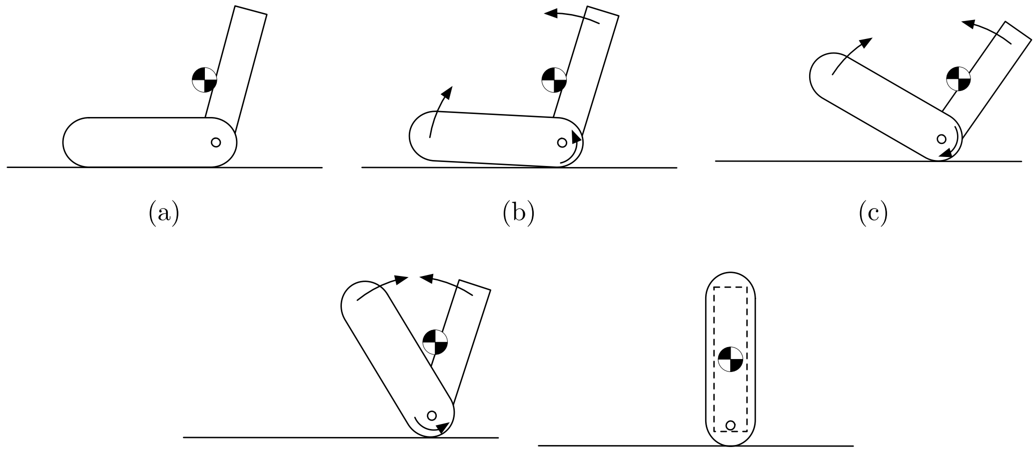
$$\dot{x} = \mathrm{A}x + \mathrm{B}u, \quad \mathrm{A} = \left.\frac{\delta f}{\delta x}\right|_{x=0}, \quad \mathrm{B} = g(x)\Big|_{x=0}. \tag{3.1}$$

A state feedback gain matrix K_c is found using the linear quadratic regulator (LQR) method. The weighting matrices are determined by Bryson's method [53]:

$$\begin{aligned} Q_c &= diag\left(\left[\frac{1}{(3\pi)^2}\ \frac{1}{(1/8)^2}\ \frac{1}{(1/8)^2}\ \frac{1}{(8\pi)^2}\ \frac{1}{(7/2)^2}\ \frac{1}{(7/2)^2}\right]\right) \\ R_c &= diag\left(\left[\frac{1}{(1/4)^2}\ \frac{1}{(1/2)^2}\right]\right), \quad N_c = 0_{6\times 2}. \end{aligned}$$

As noted, the motor model in (2.30) is valid for a bounded control input u, so each element from $u = K_c x$ is saturated at unity magnitude.

An important finding is that simply running the controller from certain statically stable positions (e.g. the tread assembly horizontal $\alpha = 90°$ and the chassis just past vertical $\theta = -15°$) is sufficient to upright and stabilize the robot, see Fig. 3.1. Given these initial conditions, the center of mass is near the end of the treads by the chassis (Fig. 3.1a), and the control law derived from LQR will drive the treads backwards (Fig. 3.1b), which will cause the robot to tip forwards leaving only the tread sprocket in contact with the ground (Fig. 3.1c). Simultaneously, the V-angle is reduced by actuation of the motors between the chassis and tread assemblies (Fig. 3.1d) and the treads are driven until the sprocket is back in the original position (Fig. 3.1e).



(a) (b) (c) (d) (e)

Figure 3.1: Maneuver for uprighting into V-balance mode with LQR control with center of mass indicated.

Discretization

Since the control will be implemented with digital electronics, we must discretize the system; we choose a sample time of $h = 0.01s$. We convert our continuous-time system from (3.1) using the matrix exponential:

$$F = \Phi(h), \quad \Phi(\tau) = \mathbf{e}^{\mathrm{A}\tau},$$
$$G = \Theta(h), \quad \Theta(\tau) = \int_0^\tau \mathbf{e}^{\mathrm{A}\eta}\mathrm{B}d\eta.$$

The continuous time weighting matrices are also transformed (given that $N_c = 0$):

$$Q_d = \int_0^h \Phi^T(\tau)Q_c\Phi(\tau)d\tau,$$
$$R_d = \int_0^h \Theta^T(\tau)Q_c\Theta(\tau) + R_c d\tau,$$
$$N_d = \int_0^h \Phi^T(\tau)Q_c\Theta(\tau)d\tau.$$

A new state feedback matrix K_d is found using the discrete-time LQR method. Comparing the simulation results between the continuous-time and discrete-time state feedback controllers (both applied to the continuous-time dynamic model of the nonlinear plant) reveals negligible performance loss.

3.2.2 C-Balancing

We wish to stabilize the unstable equilibrium manifold with the center of mass directly over the contact point. By statics, the equilibrium condition is

$$m_C L_C \sin\theta^* = (m_T L_T + m_C L_0)\sin\alpha^* \tag{3.2}$$

although this expression can equivalently be found by examining the dynamics and setting the velocity and acceleration terms to zero. Given the values of the parameters m_C, m_T, L_C, L_T, and L_0 of the developed prototype (see Sec. 5.5), it is possible to solve for α^* for any $\theta^* \in [-\pi, \pi]$, but not vice versa. As the magnitude of θ varies from 0 to π, the height of the center of mass, and thus the dynamics and the eigenvalues of the linearized system, change significantly. Unlike V-balancing, it is not possible to derive a single state feedback matrix that performs well across all values of θ. Instead, we implement a lookup table for gain scheduling. Entries are generated by spanning $\theta^* \in [0, \pi]$, solving for α^* using (3.2), and linearizing about that equilibrium point, similarly to the control design for V-balancing. In practice, the state feedback matrix is linear interpolated between adjacent entries in the lookup table based on the absolute value of θ^*.

Also unlike V-balancing, a dedicated sequence of maneuvers is required to transition from horizontal into C-balancing. A quasistatic method to transform from horizontal to C-balancing mode is used (Fig. 3.2). First, the robot lifts itself into an “A” shape by synchronously rotating the tread assemblies out and pushing them against the ground until the rear panel of the chassis is parallel to, and in contact with, the ground (Fig. 3.2a - Fig. 3.2g). One of the tread assemblies is then rotated up and over the top of the chassis until it makes contact with the ground on the opposite side of the robot (Fig. 3.2h - Fig. 3.2j). Then both of the tread assemblies are simultaneously rotated towards the chassis until parallel in concert with the treads rotating to keep the center of mass constant (Fig. 3.2k). This action lifts the rear of the chassis off the ground and the robot

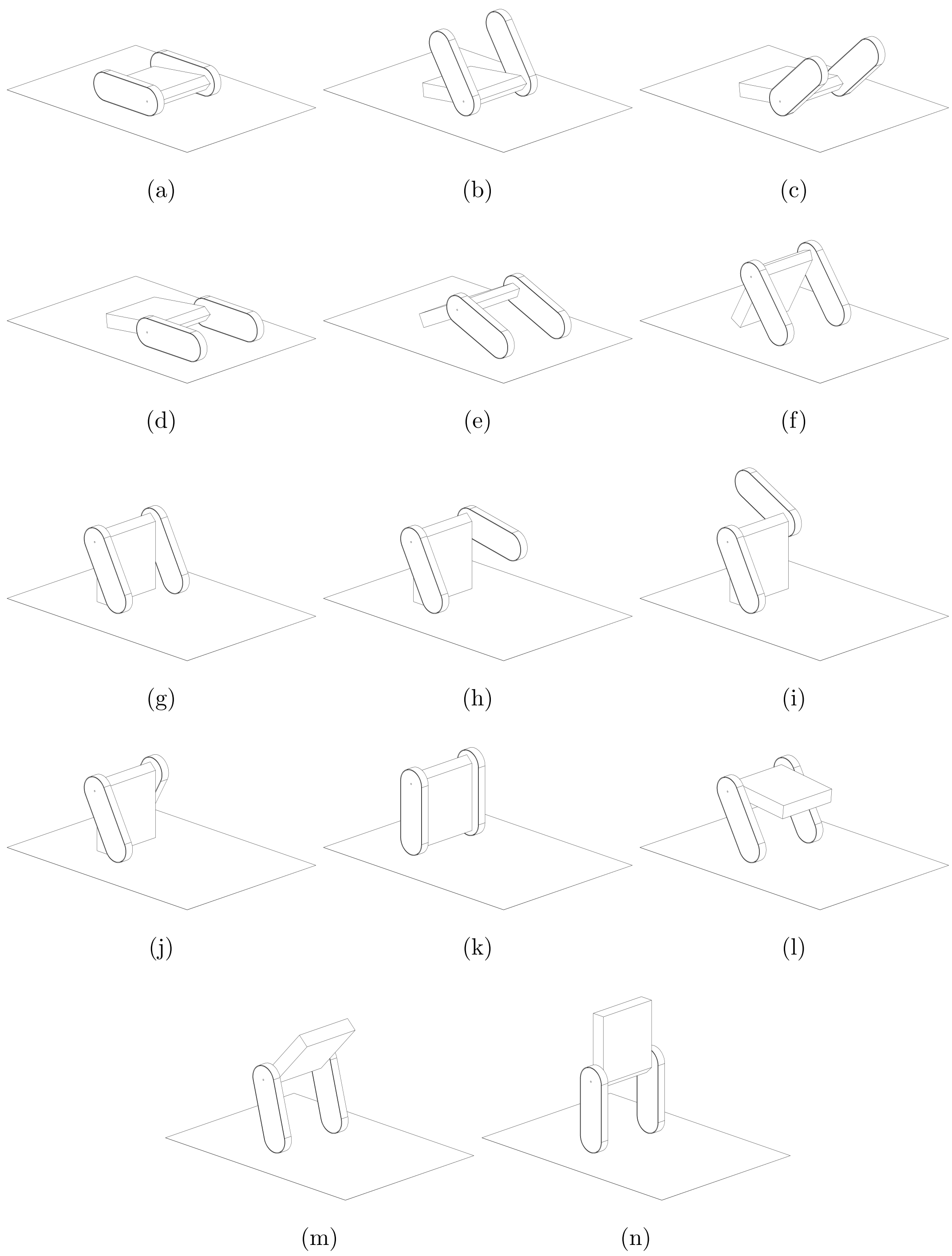


Figure 3.2: Quasistatic maneuver for uprighting into C-Balancing mode.

is now in C-balancing mode. The chassis may be rotated to any arbitrary angle while maintaining balance (Fig. 3.2l - Fig. 3.2n). This sequence of moves can be reversed to transition back to horizontal.

3.2.3 Perching

When the system is in stiction, the control authority is reduced. In the worst case, when both the treads and chassis are in stiction ($S_\beta = S_{TC}$), there is no effect of the motor torque on the treads or tread assemblies until enough torque is applied to break the stiction ($S_{TC}^T B = 0_{1\times 2}$). We therefore focus our control design on the case where neither the treads nor the chassis are in stiction ($S_\beta = S_{NS}$). Noting that the dynamics of w are directly coupled to ϕ and α by (2.37), we can choose a reduced coordinate set $q_r = \begin{pmatrix} \phi & \alpha & \theta \end{pmatrix}^T$ such that $\dot{q}_r = \underset{\sim}{S}_{NS}\nu_{NS}$ where $\underset{\sim}{S}_{NS}$ is the bottom three rows of (2.39). Similarly, $\ddot{q}_r = \underset{\sim}{S}_{NS}\dot{\nu}_{NS}$.

Concatenating q_r and $\dot{q}_r$ yields a complete state vector x. Rewriting (2.41) and premultiplying by $\underset{\sim}{S}_{NS}$ to recover $\ddot{q}_r$ from $\dot{\nu}_{NS}$, we see that this nonlinear system is affine in the inputs:

$$x = \begin{pmatrix} q_r \\ \dot{q}_r \end{pmatrix}, \quad \dot{x} = f(x) + \Gamma(x)u,$$
$$f(x) = \begin{pmatrix} \dot{q}_r \\ \underset{\sim}{S}_{NS}[S_{NS}^T M(q) S_{NS}]^{-1} S_{NS}^T [\frac{\delta P}{\delta \dot{q}} - BZ(\dot{q}) - F(q,\dot{q})] \end{pmatrix},$$
$$\Gamma(x) = \begin{bmatrix} 0_{3\times 2} \\ \underset{\sim}{S}_{NS}[S_{NS}^T M(q) S_{NS}]^{-1} S_{NS}^T B\Sigma \end{bmatrix}.$$

Equilibrium Conditions

We seek to find equations to describe the equilibrium manifold of the system, that is, we seek to find expressions for x^* and u^* such that $\dot{x} = f(x^*) + \Gamma(x^*)u^* = 0_{6\times 1}$ given $\dot{q}_r = 0_{3\times 1}$.

Given that $S_{NS}^T M(q) S_{NS}$ is positive definite, we can solve the matrix expression

$$S_{NS}^T\{B[\Sigma u - Z(\dot{q})] + \frac{\delta P}{\delta \dot{q}} - F(q,\dot{q})\} = 0_{3\times 1} \tag{3.3}$$

from (2.41) and simplify by appropriately substituting (2.37) and $\dot{q}_r = 0_{3\times 1}$ to get three equations

$$\begin{aligned}
&-(\sigma_S u_1^* + mgr\cos\alpha^*)/\sqrt{r^2+1} = 0,\\
&-[\sigma_T u_2^* + g(mw - m_T L_T)\sin\alpha^*]/\sqrt{(r+\rho)^2+1} = 0,\\
&\ \sigma_S u_1^* + \sigma_T u_2^* + m_C L_C g \sin\theta^* = 0. \qquad (3.4)
\end{aligned}$$

By inspecting the above equations, we see that there are unique solutions for the feedforward terms u_1^* and u_2^* for a given x:

$$u_1^* = -mgr\cos\alpha^*/\sigma_S, \qquad (3.5)$$

$$u_2^* = g(m_T L_T - mw)\sin\alpha^*/\sigma_T. \qquad (3.6)$$

We also see that there is a unique solution for θ^*, combining (3.4), (3.5), and (3.6)

$$\theta^* = \arcsin\left(\frac{mr\cos\alpha^* - (m_T L_T - mw)\sin\alpha^*}{m_C L_C}\right). \qquad (3.7)$$

The expression (3.7) can also be derived from a static analysis where the center of mass is constrained to be directly above the step edge.

We define $\tilde{x} = x - x^*$ and $\tilde{u} = u - u^*$ such that

$$\dot{\tilde{x}} = f(\tilde{x} + x^*) + \Gamma(\tilde{x} + x^*)(\tilde{u} + u^*). \qquad (3.8)$$

Friction Compensation

The signum function in (2.34) due to Coulomb friction cannot be linearized about the origin. Instead we ignore this term in the linearization and add a separate friction compensator u_F to the controller [54]. The linearizable plant dynamics are

$$f_l(x) = \begin{pmatrix} \dot{q}_r \\ -S_{NS}[S_{NS}^T M(q) S]^{-1} S_{NS}^T [BZ(\dot{q}) + F(q,\dot{q})] \end{pmatrix}. \qquad (3.9)$$

There are a number of factors to be considered when designing the friction compensator. The compensator should mitigate both stiction and Coulomb friction without destabilizing the equilibrium manifold. An obvious choice to "eliminate" the Coulomb friction may be

$$u_F = \begin{pmatrix} -\frac{\mu_k mgr}{\sigma_S \sin\alpha}\cdot \mathrm{sgn}(\dot{\phi} - \dot{\alpha}) \\ -(c_T/\sigma_T)\cdot \mathrm{sgn}(\dot{\alpha} - \dot{\theta}) \end{pmatrix}.$$

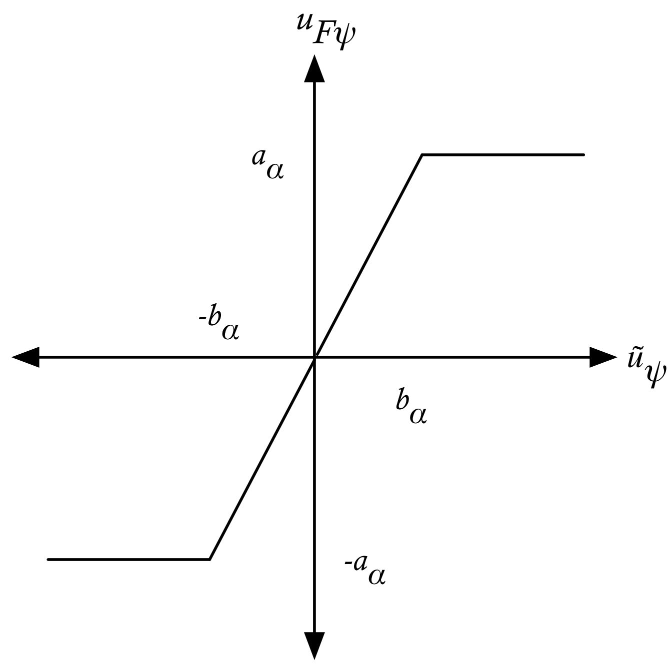


Figure 3.3: Friction compensator $u_{F\psi}$ as a function of $\tilde{u}_\psi$.

However, practical matters such as backlash and chatter in the physical system limit the use of the signum function as a candidate compensator. We can instead saturate a steep line passing through the origin, see Fig. 3.3. It is also inherently difficult to measure near-zero relative velocity with optical encoders (more in section 4.2.1) and so smoother performance is possible when using the sign of $\tilde{u}_\psi$ instead of the relative velocity. We also desire a function that is simple to implement on an embedded controller, see section 5.5.1. We come to a friction compensator with the form

$$u_F = \begin{pmatrix} \min(\max(\tilde{u}_1 a_S/b_S,\, -a_S),\, a_S)/\sin\alpha \\ \min(\max(\tilde{u}_2 a_T/b_T,\, -a_T),\, a_T) \end{pmatrix} \tag{3.10}$$

which is illustrated in Fig. 3.3 and where $a_S \leq \mu_k mgr/\sigma_S$, $a_T \leq c_T/\sigma_T$, and a_α, b_α can be tuned empirically on the physical system. The effect of the friction compensator can be seen in section 5.5.3.

Linearization and Integral Control

We linearize the system at the origin of the transformed system (3.8) using the linearizable plant dynamics (3.9)

$$\dot{\tilde{x}} = \mathrm{A}\tilde{x} + \mathrm{B}(\tilde{u} + u^*), \quad \mathrm{A} = \frac{\delta f_l(\tilde{x} + x^*)}{\delta \tilde{x}}\Big|_{\tilde{x}=0}, \quad \mathrm{B} = \Gamma(\tilde{x} + x^*)\Big|_{\tilde{x}=0}.$$

In order to increase the robustness of the system to disturbances such as parameter and sensor error, we augment the state vector $\tilde{x}$ with the integrated regulation error ξ [55], defined by

$$\dot{\xi} = \begin{pmatrix} (\phi - \alpha) - (\phi^* - \alpha^*) \\ (\alpha - \theta) - (\alpha^* - \theta^*) \end{pmatrix},$$

noting that $\phi - \alpha$ is approximately w when $r \gg \rho$. We further define

$$\bar{x} = \begin{pmatrix} \tilde{x} \\ \xi \end{pmatrix}, \quad \mathrm{C} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix},$$

and the system can now be written

$$\dot{\bar{x}} = \mathcal{A}\bar{x} + \mathcal{B}(\tilde{u} + u^*), \tag{3.11}$$
$$\mathcal{A} = \begin{bmatrix} \mathrm{A} & 0_{6\times 2} \\ \mathrm{C} & 0_{2\times 2} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} \mathrm{B} \\ 0_{2\times 2} \end{bmatrix}.$$

Linear Quadratic Regulator

A state feedback gain matrix can be found using the linear quadratic regulator (LQR) method. The weighting matrices are determined by Bryson's method [53]. For the horizontal equilibrium where $\alpha^* = \theta^* = \pi/2$

$$Q_C = \mathrm{diag}\left(\frac{1}{(k)^2}\ \frac{1}{(k)^2}\ \frac{1}{(k)^2}\ \frac{1}{(\pi)^2}\ \frac{1}{(l)^2}\ \frac{1}{(l)^2}\ \frac{1}{(4/5)^2}\ \frac{1}{(7/5)^2}\right),$$
$$R_C = \mathrm{diag}\left(\frac{1}{(2/5)^2}\ \frac{1}{(1/4)^2}\right), \quad N_C = 0_{7\times 2},$$

where $k = 1/8$ and $l = 7/2$.

Discretization

Since the control will be implemented with digital electronics, we must discretize the system; we choose a sample time of $h = 0.01s$. We convert our continuous-time system from (3.11) using the matrix exponential

$$\begin{aligned} \bar{x}_{k+1} &= F\bar{x}_k + G(\tilde{u}_k + u^*), \\ F &= \Phi(h), \quad \Phi(\tau) = \mathbf{e}^{\mathcal{A}\tau}, \\ G &= \Theta(h), \quad \Theta(\tau) = \int_0^\tau \mathbf{e}^{\mathcal{A}\eta}\mathcal{B}d\eta. \end{aligned}$$

The continuous time weighting matrices are also transformed (given that $N_C = 0$):

$$\begin{aligned} Q_D &= \int_0^h \Phi^T(\tau)Q_C\Phi(\tau)d\tau, \\ R_D &= \int_0^h \Theta^T(\tau)Q_C\Theta(\tau) + R_C d\tau, \\ N_D &= \int_0^h \Phi^T(\tau)Q_C\Theta(\tau)d\tau. \end{aligned}$$

A discrete state feedback matrix K is found using the discrete-time LQR method $\tilde{u} = K\bar{x}$. The final control law is of the form $u = K\bar{x} + u^* + u_F$. As noted, the motor model (2.35) is valid for a bounded control input $\in [-1, 1]$, so each element of u is saturated at unity magnitude.

Trajectory Generation and Gain Scheduling

To drive up a step edge, we plan a trajectory that satisfies equilibrium (3.3) at every point. This approach has the added benefit that the trajectory can simply be reversed to descend a step edge, while maintaining equilibrium. We vary w from zero to the length of the tread assembly and choose α^* to be constant, this results in a range of values for $\phi^* \in [\phi_S, \phi_F]$. The value of θ^* is given by (3.7).

The dynamics (3.9) change considerably across this range, so we choose $n_G = 5$ values of ϕ^* evenly distributed $\in [\phi_S, \phi_F]$, use our constant α^*, solve for θ^*, and find a discrete-time state feedback matrix K at each point. The weighting matrices Q_C and R_C are adjusted at each point to keep the norm of K constant. Five different positions along a step edge climbing equilibrium manifold are shown

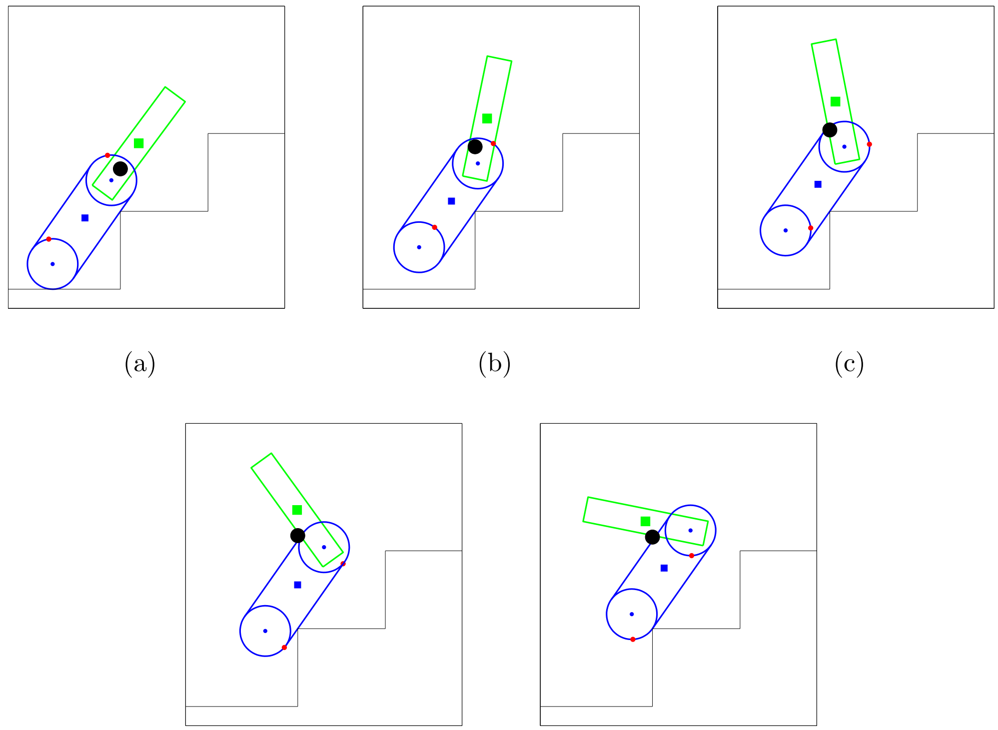


Figure 3.4: Stair climbing trajectory with centers of mass of chassis and tread assemblies as squares, and overall center of mass as a circle.

in Fig. 3.4. We construct a $8 \times 2 \times n_G$ lookup table containing all of the state feedback matrices and linearly interpolate between the entries of the table depending on the reference value ϕ^*.

Simulation

We created a simulation in order to validate the model and controller. The full nonlinear, continuous time dynamics of the system (2.42) are used to represent the plant. The controller is implemented in discrete time as per above. The choice of which constraints S_β to apply at time step t_i is determined by logical functions of the state vector $\bar{x}_i$, the control input u_i, and the previous constraint condition. Starting at rest, both the treads and the chassis are assumed to be in stiction, $S_\beta = S_{TC}$. If and when there is sufficient torque (from combined contributions of $F(q, \dot{q})$ and u) to break the stiction between the treads and the tread assemblies and/or the chassis and the tread assemblies, then a different S_β is used. If the relative velocity between the treads and the tread assemblies or the chassis and tread assemblies changes sign (passes through zero), then the system reenters stiction. Entering stiction is treated as an inelastic collision, where the angular velocity of the involved bodies is set equal by conservation of momentum. Simulation results are shown in section 5.5.3.

3.2.4 Switchbot

While the controls for Switchblade nano followed the exact same dynamics and procedure for deriving control algorithms for V and C-balancing, just with different parameter values, Switchbot requires separate, though analogous, control design.

Equilibrium Conditions

With seven degrees of freedom, the equilibrium manifold of the Switchbot dynamics (2.61) is large and complex. In order to decrease the dimensions of this space and find reasonable expressions for q^* and u^* for which $\ddot{q} = 0_{7\times1}$, we make a number of assumptions:

1. $\dot{q}^* = 0_{7\times1}$.
2. Both pulleys are in contact with the ground.
3. Neither pulley slips against the ground.
4. $|\gamma_R^*| = |\gamma_L^*|$ and $|\alpha_R^*| = |\alpha_L^*|$.
5. $\operatorname{sgn}(\gamma_R^*)\operatorname{sgn}(\gamma_L^*) = \operatorname{sgn}(\alpha_R^*)\operatorname{sgn}(\alpha_L^*)$.
6. $L_K \sin\alpha_R^* + L_H \sin\gamma_R^* = L_H \sin\gamma_L^* + L_K \sin\alpha_L^*$ (pulleys are collocated).

These assumptions imply that the overall center of mass must be directly above the pulleys. Note that some of these assumptions may be relaxed to find alternate trajectories along the equilibrium manifold. From a static analysis of Fig. 2.7, or equivalently derived from setting (2.61) equal to zero, we can derive an expression for the elements of u^*, given the above assumptions

$$
\begin{aligned}
u_1^* =& 0, \\
u_2^* =& 0, \\
u_3^* =& \{-g[m_L L_L + (m_B/2 + m_U) L_K] \sin \alpha_R\}/\sigma_K, \\
u_4^* =& \{-g[m_L L_L + (m_B/2 + m_U) L_K] \sin \alpha_L\}/\sigma_K, \\
u_5^* =& \{\sigma_K u_3^* - g[m_U L_U + (m_B/2) L_H] \sin \gamma_R\}/\sigma_H, \\
u_6^* =& \{\sigma_K u_4^* - g[m_U L_U + (m_B/2) L_H] \sin \gamma_L\}/\sigma_H.
\end{aligned}
$$

Trajectory Generation

We seek to create two particular motion trajectories. The first enables the robot to transition from a statically stable kneeling position (with the lower legs parallel to and in contact with the ground, $\alpha_R = \alpha_L = \pi/2$, Fig. 3.5a) to an upright balancing stance, Fig. 3.5b. In this maneuver, we choose to keep the right and left legs synchronized ($\phi_R = \phi_L$, $\alpha_R = \alpha_L$, $\gamma_R = \gamma_L$), so we can simply use the four degree of freedom dynamic model developed in the first half of Sec. 2.4.4. As with the maneuver for Switchblade to transition into V-balancing mode (Fig. 3.1), the center of mass is positioned over the far end of the lower leg (tread assembly). We choose $\alpha^* \in [\pi/2, 0.25]$ and $\theta^* \in [-\pi/2, 0]$ and then solve for γ^* which maintains the center of mass directly over the pulleys by

$$
\gamma^* = \arcsin\left(-\frac{[L_L m_L + L_K(m_U + m_B)] \sin \alpha^* + L_B m_B \sin \theta^*}{L_U m_U + L_H m_B}\right). \tag{3.12}
$$

Similar to the Switchblade C-balancing controller (Sec. 3.2.2), we choose $n_G = 6$ evenly spaced points along the trajectory, linearize and discretize the system and solve for a discrete time state feedback matrix K with LQR at each point. To schedule gains, we linearly interpolate between adjacent entries in the lookup table depending on the reference command x^*. The final control law has the form $u = K_{LUT}(x - x^*) + u^*$.

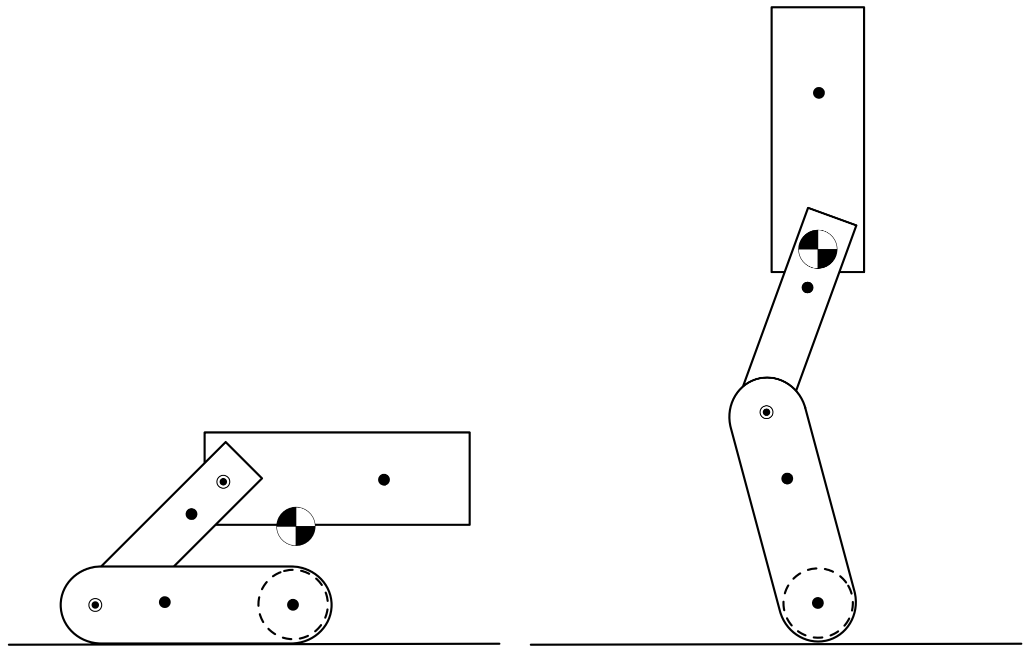


Figure 3.5: Switchbot uprighting trajectory with centers of mass of chassis and leg segments as dots, and overall center of mass as a roundel.

The second trajectory enables the the robot to pivot its legs alternately, while stationary or moving horizontally, to, in essence, "moonwalk," see Fig. 3.6. For this trajectory, in addition to the assumptions above, we choose $\theta^* = 0$ and use the seven degree of freedom dynamic model since the right and left legs will have different (though opposite, by assumption 4 above) orientations. We choose $\alpha_R^* = -\alpha_L^* \in [-0.25, 0.25]$ and can reuse (3.12) to find the magnitude of γ^* since the pulleys are collocated and the leg positions are mirrored, and setting $\operatorname{sgn}(\gamma_R^*) = -\operatorname{sgn}(\alpha_R^*)$ and $\operatorname{sgn}(\gamma_L^*) = -\operatorname{sgn}(\alpha_L^*)$. To calculate the reference position of the left pulley (ϕ_L) relative to the position of the right pulley (ϕ_R), assuming no-slip, we can simplify (2.57) by plugging in $\alpha_L = -\alpha_R$ and $\gamma_L = -\gamma_R$

$$\phi_L^* = \phi_R^* + \frac{2}{r}(L_K \sin \alpha_R^* + L_H \sin \gamma_R^*). \tag{3.13}$$

To create a controller, we split the system into two identical halves with the same dynamics (using the four degree of freedom model with half the mass and motors).

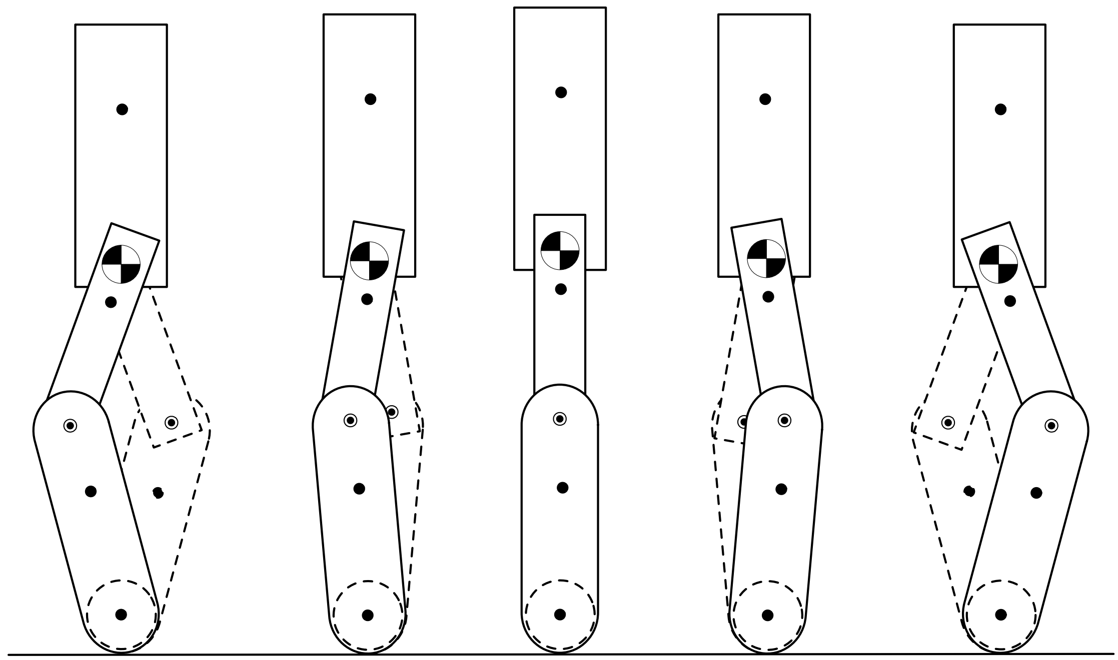


Figure 3.6: Switchbot moonwalking trajectory with centers of mass of chassis and leg segments as dots, and overall center of mass as a roundel.

We next choose $n_G = 5$ points evenly spaced in $\alpha_R \in [0.25, -0.25]$, calculate the rest of the equilibrium state x^* with (3.12) and keeping $\theta^* = 0$, linearize and discretize the halved system and solve for a discrete time state feedback matrix K with LQR at each point which is used for both halves. The result is a decentralized system where the motors in the right leg (u_1, u_3, and u_5) are actuated based on the states of ϕ_R, α_R, γ_R, and θ and the motors in the left leg (u_2, u_4, and u_6) are actuated based on the states of ϕ_L, α_L, γ_L, and θ. This is more tractable to implement in an embedded controller than a full MIMO controller. The robot may be seen performing both trajectories in Supplemental File 3: Switchbot video.

3.3 SkySweeper

We formulated a finite state machine controller to implement all maneuvers listed in Sec. 1.4.4. Each state has three actions: the positions of the two clamps and the control input u to the motor. To define the transitions between the states,

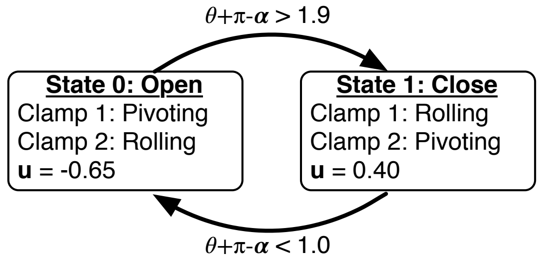


Figure 3.7: SkySweeper inchworm maneuver finite state machine.

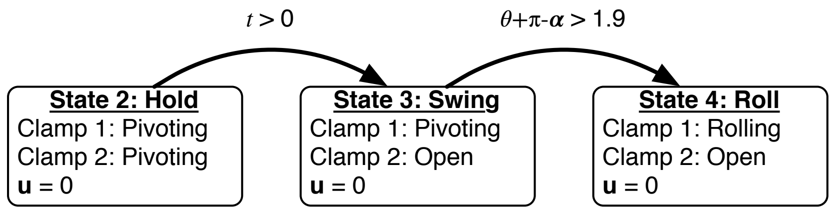


Figure 3.8: SkySweeper swing & roll maneuver finite state machine.

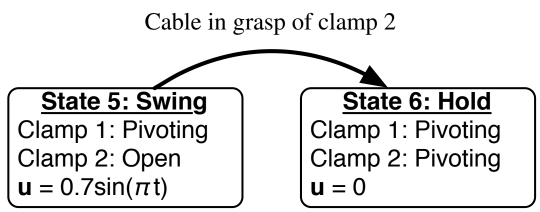


Figure 3.9: SkySweeper swing-up maneuver finite state machine.

we limit ourselves to logical expressions with simulated measurements of sensors that are possible on the prototype: the separation angle between the two links, $\theta + \pi - \alpha$, the spring deflection, $\alpha - \gamma$, if a cable is within the grasp of either clamp, and time, see Sec. 5.6.2. State machines for the four different maneuvers are illustrated in Fig.s 3.7-3.10. The state machines can be practically implemented as a switch structure in most programming languages. The exact values used in the logical expressions and the control input to the motor were determined in simulation.

We created a simulation environment to test the finite state machine controller from above with the dynamic equations of motion from Sec. 2.5. We were able to test different parameter values virtually before constructing a prototype.

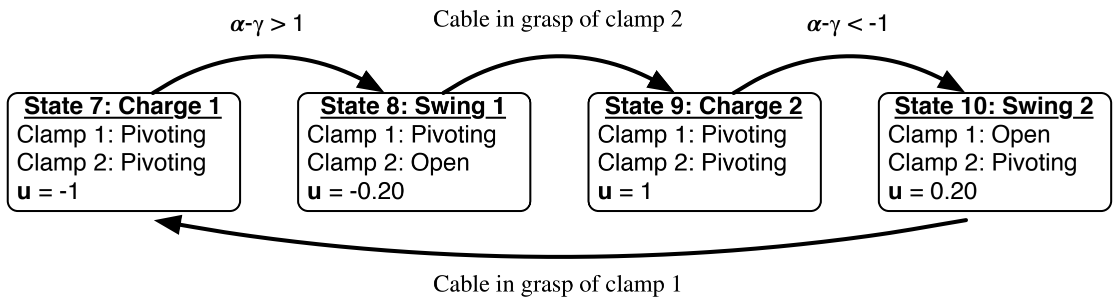


Figure 3.10: SkySweeper backflip maneuver finite state machine.

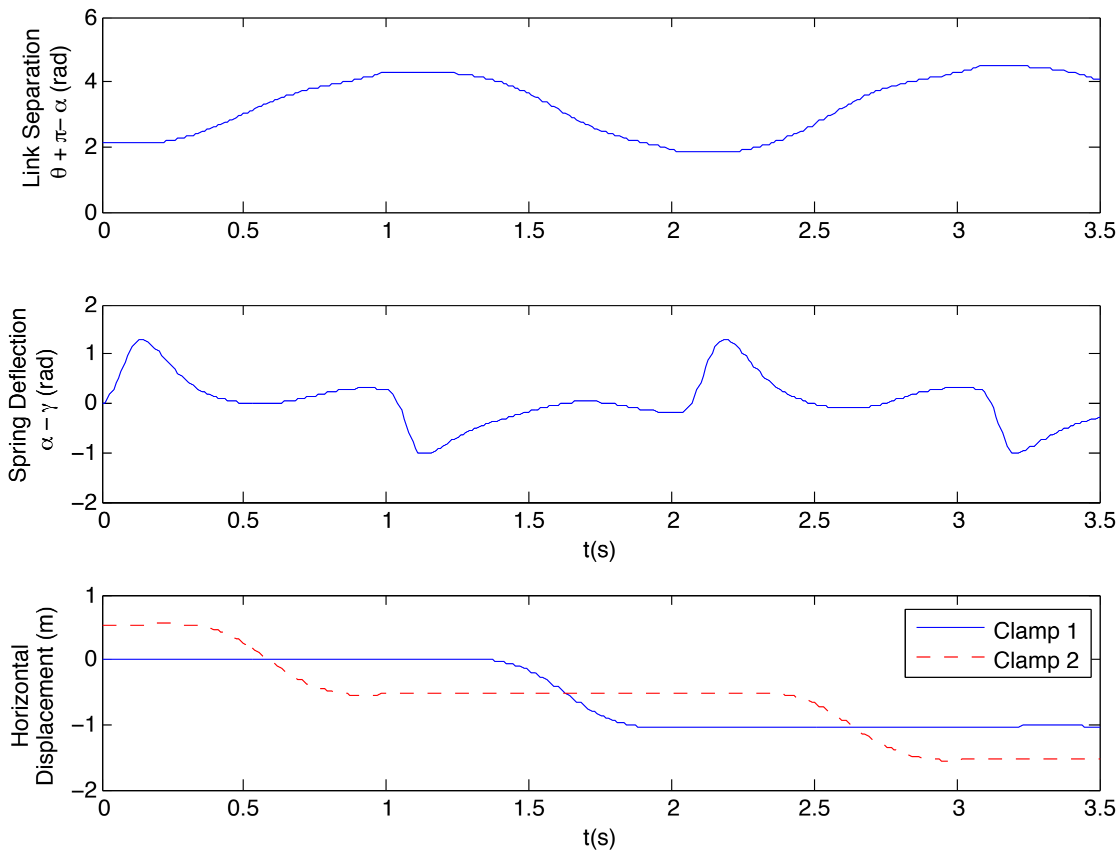


Figure 3.11: Simulation results for backflip maneuver.

The set of n second-order differential equations (2.67) can easily be written as $2n$ first-order equations and then marched forward in time from a prescribed initial condition as a switched system dependent on the clamp configuration. In inchworm mode, the robot alternates which clamp is in pivoting or rolling position. This is treated as an inelastic collision where 85% of the momentum is conserved.

Additionally, a delay of $50ms$ is imposed when switching clamping configurations to account for the time it takes the clamps to physically change positions.

All four maneuvers were simulated successfully, animations are included in Supplemental File 4: SkySweeper video (using parameter values from the prototype). Example code to produce similar animations from MATLAB/Simulink simulations is included in App. C. Results for the backflip maneuver are shown in Fig. 3.11. The maximum dynamic load on the clamp during swinging, from (2.68), is $6.75N$ or $1.48g$.

3.4 Acknowledgements

Part of this chapter concerning Switchblade has appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning SkySweeper has appeared in print: N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2339-2344, 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the perching maneuver has been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Estimation

On the physical systems we are limited in what states can be observed directly with the onboard sensors. No external sensors (e.g. motion capture systems) are used. Thus we design estimators to recreate the full state vector x. The robots are instrumented with combinations of MEMS accelerometers and gyroscopes, quadrature encoders (optical or magnetic), and rotary potentiometers.

4.1 Inertial sensors

A MEMS gyroscope can measure the rotational velocity of a robot chassis providing a direct measurement of $\dot{\theta}$. We may apply a first-order digital low pass filter with a cutoff frequency of 60 rad/s to remove high frequency noise.

The MEMS accelerometer outputs acceleration magnitudes in two orthogonal axes; by taking the arctangent of the two values, the gravity vector can be calculated, under the assumption that the body accelerations are small. Adding a first-order discrete low pass filter attenuates noise as well as disturbances from body accelerations. This method yields an acceptable estimate of θ only at low frequencies.

Another method to estimate θ is to numerically integrate the output of the gyroscope. Integration error and thermal bias build over time, but can be eliminated using a first-order digital high pass filter, thus creating an estimate of θ that is valid only at high frequencies. We create a complementary filter by choosing the accelerometer low pass filter constant μ_{ALP} and integrated gyroscope high pass filter constant μ_{GHP} such that they sum to unity, which allows the two measurements to be simply summed for a single estimate of θ valid across a wide frequency range [56].

$$\mu_{GHP} = \frac{1/\omega_c}{1/\omega_c + h}, \quad \mu_{ALP} = \frac{h}{1/\omega_c + h}$$

We choose the crossover frequency $\omega_c = 0.5$ rad/s to trade-off sensor errors such as body accelerations and integration error. Pseudocode to implement this complementary filter on a variety of embedded microcontrollers is below.

```
// time constant to mix accelerometer and gyroscope
omega_c = 0.5;

// loop period, in seconds
h = 0.010;

// integrated gyroscope high pass filter constant
mu_GHP = (1/omega_c) / (1/omega_c + h);

// accelerometer low pass filter constant
mu_ALP = h / (1/omega_c + h);

// get angle using arctangent of two accelerometer axes
acc = atan2(accX,accY);
// apply low pass filter
accLP = accLPold + mu_ALP*(acc - accLPold);

// numerically integrate gyroscope
gyroInt = gyroIntHPold + h*gyro;
// apply high pass filter to integrated gyroscope
gyroIntHP = mu_GHP*(gyroInt);

// sum components from accelerometer and integrated gyroscope
theta = accLP + gyroIntHP;

// set current values as old
```

```
accLPold = accLP;
gyroIntHPold = gyroIntHP;
```

4.2 Quadrature Encoders

The optical encoders precisely measure relative rotation, for example, between the tread sprockets and tread assemblies and between the chassis and the tread assemblies in the Switchblade system. Measuring the relative velocity with the encoders is more difficult, particularly at low velocity. In order to normalize encoder velocity, we define

$$\mathcal{M} = \frac{2\omega h \, \mathrm{CPR}}{\pi} \tag{4.1}$$

where ω the rotational velocity (in radians per second) of the encoder disc, h is the sample time, and CPR is number of encoder ticks per revolution, $\mathcal{M}$ is then the number of quadrature encoder ticks per sample time (which is four times more than the number of rising or falling edges of one channel only).

4.2.1 Real Time Encoder Velocity

Counting the number of encoder ticks in a single time step h has unacceptable discretization noise when the speed is low because there are few to no encoder ticks per time step, $\mathcal{M} \leq \mathcal{O}(10)$. A low pass or moving average filter may be applied to smooth the data, but it inherently adds delay, decreasing the bandwidth of the system which can cause the controller to fail.

An alternative method is to use a high speed clock (e.g. a field programmable gate array, FPGA) to measure the time between subsequent encoder ticks. Using quadrature encoding and counting both rising and falling edges of both channels of the encoder can increase the resolution by a factor of four (as well as determine the direction of rotation), but this common method can actually introduce significant noise to the velocity measurement. In practice, the output of the A and B channels of the encoder are not ideal square waves (see Fig. 4.1), the high time is not equal to the low time for a constant speed, $A_{RF} \neq A_{FR}$ and

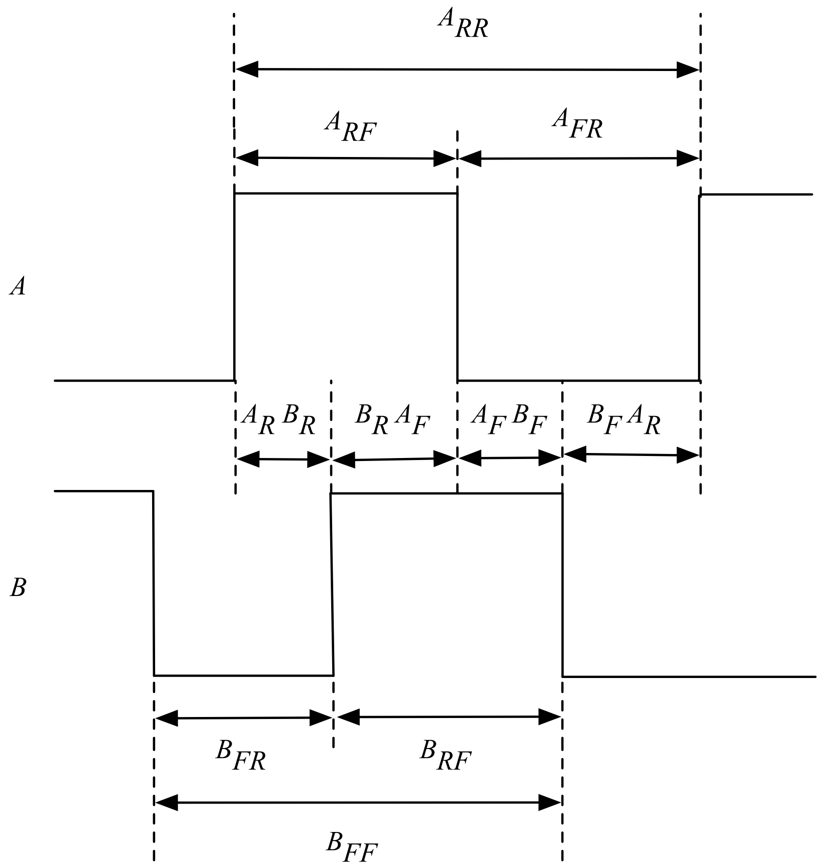


Figure 4.1: Illustration of imperfection in quadrature encoder signals.

$B_{RF} \neq B_{FR}$. Further, the phase offset between the A and B channels is not exactly 90°, and varies between rising and falling edges ($A_R B_R \neq B_R A_F$, $B_R A_F \neq A_F B_F$, etc.). What is more consistent is the 360° period from like edge to like edge (A_{RR}, B_{FF}, etc.), assuming the encoder is mounted concentrically with the shaft. By measuring the four different periods independently instead of only measuring the sub-periods between subsequent edges of any type, we maintain the factor of four increase in resolution (and direction information) without introducing error from the asymmetry of the signals [57].

When a change of direction occurs, one channel of the encoder will output two opposite edges before the other channel outputs a single edge. In this case, it is impossible to measure the instantaneous speed because it is uncertain how far the encoder rotated before changing direction. However, we know that the average velocity between the two edges is zero because the net displacement over the time interval since the last edge is zero, so we assume a measured speed of zero.

When the speed is great enough that at least two encoder edges pass in a single time step ($\mathcal{M} \geq 2$), it is possible to average the two (or more) time periods to smooth the velocity estimate. The smoothing effect is proportional to the speed. This is important to not only remove sensor noise from imperfections in the encoder disc, but also to smooth process noise. The disturbances caused by, e.g., the sprocket teeth engaging and disengaging with the treads and other factors can induce significant noise if only measuring one encoder period.

By measuring the amount of time between encoder pulses, it is not possible to measure a speed of zero. Given the current time and the time since each of the last edges, it is possible to bound the actual speed by calculating the maximum speed that would not have yet produced an encoder edge ($\mathcal{M} < 1$). In practice, after the estimate decays sufficiently, it becomes negligible. Measuring the period of an encoder pulse is not an instantaneous measurement of speed, in fact, it only yields the average speed over the interval of the last encoder pulse cycle. As speed decreases, this measurement delays increases, decreasing the bandwidth of the system. Using a higher resolution encoder will improve performance, so long as the high speed clock is still significantly faster ($\mathcal{O}(100)$) than the encoder pulse rate at the maximum shaft rotational velocity to avoid introducing discretization error. Encoder resolution is often a function of cost and size. Note that placing the encoder directly on the motor shaft, instead of the gearbox output shaft, greatly increases the resolution of the rotation of the output shaft (by a factor of Γ, the gearbox ratio) without increasing the cost or size of the encoder, at the expense of not being able to measure the backlash at the gearbox output shaft.

4.2.2 Post Processed Encoder Velocity

In the DC motor dynamometer, we use a high resolution (2500 counts per revolution) encoder, but cannot measure the time between edges precisely enough with the National Instruments myDAQ. However, we do not require a real time estimate of the velocity because the data may be processed after the test is complete. Some signal processing and filtering must be done before the data can be used for parameter identification. In particular, velocity and acceleration have to be estimated carefully from discrete measurements of position. Any noise in the original data can be exacerbated by numerical differentiation. The high resolution of the encoder disc and quadrature counting minimize differentiation error. We first apply some smoothing to the discrete position data and then use the second-order central difference approximation of the smoothed data to obtain velocity data:

$$\begin{aligned}\hat{\theta}_k &= \frac{1}{6}\theta_{k-1} + \frac{2}{3}\theta_k + \frac{1}{6}\theta_{k+1}, \\ \omega_k &= (\hat{\theta}_{k+1} - \hat{\theta}_{k-1})/2h,\end{aligned}$$

where h is the sample time, 0.01 s. Combining the above two equations gives:

$$\omega_k = (-\theta_{k-2} - 4\theta_{k-1} + 4\theta_{k+1} + \theta_{k+2})/12h. \tag{4.2}$$

This formula yields an acceptable estimate of velocity, but the discretization-induced noise is too high to immediately reapply (4.2) to estimate acceleration. We first apply a 21 time step wide Gaussian filter to smooth out the noise. The window is chosen to be wide enough to smooth the high frequency discretization and differentiation noise but not so wide as to eliminate the low frequency signal of interest. Note that the data is post-processed such that the filter is centered at the current time step and no phase delay is added.

4.3 Potentiometers

Potentiometers (variable resistors) offer three distinct advantages over quadrature encoders.

1. They are generally cheaper, not just in the cost of the components, but they also have a cheaper assembly cost because there are fewer parts and wires.

2. Potentiometers give an absolute position measurement and do not require calibration or reset each time the robot is turned on.

3. They require less hardware and software to measure position. Only a single signal to an analog to digital converter (ADC) is necessary to get a position measurement. A quadrature encoder requires two digital lines to be sampled constantly (or have interrupt routines) and a state variable is incremented.

However, there are two key disadvantages.

1. Potentiometers have a limited range of measurement. This can be aided by adding a second potentiometer 180° out of phase with the first, but this adds cost and complexity, reducing the above advantages.

2. Due to the noise generally present in analog to digital converters, it is difficult to estimate the relative velocity in real time.

For joint angles that do not move quickly (e.g. hips and knees in Switchbot) or for control systems that do not require velocity feedback (e.g. SkySweeper) this is an acceptable trade off.

4.4 Sensor Fusion

Given the estimates of $\{\theta, \dot{\theta}\}$ from the MEMS sensors and the encoder and/or potentiometer measurements, additional joint angles can be computed by simple addition. Fig. 4.2 illustrates the sensor fusion of an accelerometer, gyroscope, and encoders for a mobile inverted pendulum. For the purposes of simulation, we mimic the output of the sensors by manipulating the state vector. For the encoders, we simply quantize the simulation state given the resolution of the encoder. For the MEMS gyroscope and accelerometer, we calculate what the sensor output would be given the current position, velocity, and accelerations, add white noise (the variance of which is based on the manufacturer's specifications), and apply the complementary filter as described above.

4.4.1 A Note on Kalman Filtering

Given how useful model-based control (Ch. 3) is, one might expect model-based estimation (e.g. Kalman filter) to be as broadly used in these robotic sys-

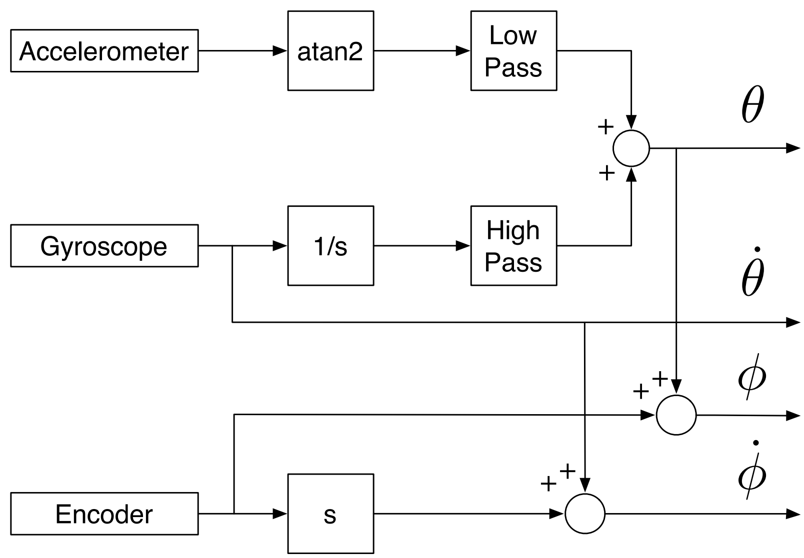


Figure 4.2: Block diagram of complementary filter with accelerometer, gyroscope, and encoders.

tems. However, there are a number of disadvantages to using model-based estimation on low-cost real-world robotic systems. Model-based estimation is prone to estimation error induced by modeling errors (such as friction or backlash) and sensor noise that is not white. The optimality condition of the Kalman filter does not hold for nonlinear systems with colored process and sensor noise. Dynamical systems that experience significant changes in dynamics (such as the complete rotation of the chassis in Switchblade C-balancing, Sec. 3.2.2) require lookup tables as a function of the reference state, similar to the model-based control strategy, adding to the memory and computational demands on the embedded microcontroller. Systems with sudden changes to the dynamic constraints (such as the different clamping conditions in SkySweeper, Sec. 2.5, or entering and exiting stiction while perching, Sec. 2.4.3) require not just additional lookup tables, but also switching conditions to determine which set of lookup tables to use. Estimation error can quickly become unstable in these systems in the presence of modeling error. The extended Kalman filter can help, but requires significantly more computational power from the embedded system and is often infeasible for low-cost microcontrollers. Alternatively, the computationally simple complementary filter described in Sec. 4.1 can be broadly applied across platforms, with different dynamics, sensors, and actuators with little to no tuning.

4.5 DC Motor Parameter Identification

To identify the parameters of the motor and driver system, RAPID employs a least-squares algorithm similar to [46]. We define N as the total number of data points taken during a trial and let $f(n)$ denote the value of f at time nh, where h is the sample time of 0.01 s. By the above construction, the model can be written in matrix form:

$$A(n)\boldsymbol{x} = \boldsymbol{b}(n), \tag{4.3}$$

where $\boldsymbol{x} \in \mathbb{R}^{5\times 1}$ is the vector of model parameters to be identified. From the set of system equations (2.7) and (2.10), we write:

$$A(n) = \begin{bmatrix} i(n) & \omega(n) & 0 & 0 & 0 \\ 0 & -i(n) & \frac{d\omega}{dt}(n) & \omega(n) & \mathrm{sgn}(\omega(n)) \end{bmatrix}$$

$$\boldsymbol{x} = [\ r \quad k \quad J_S \quad b \quad c_S\ ]^T, \quad \boldsymbol{b}(n) = \begin{bmatrix} \overline{V}(n) \\ 0 \end{bmatrix}$$

We further define:

$$R_A \triangleq \sum_{n=1}^{N} A(n)^T A(n), \quad R_{A\boldsymbol{b}} \triangleq \sum_{n=1}^{N} A(n)^T \boldsymbol{b}(n).$$

If we premultiply both sides by $A(n)^T$ and sum over n, we can rewrite (4.3) as $R_A\boldsymbol{x} = R_{A\boldsymbol{b}}$, solving for $\boldsymbol{x}$ gives $\boldsymbol{x} = R_A^{-1}R_{A\boldsymbol{b}}$, so long as R_A is invertible. Whether or not R_A is invertible depends on if the PWM input signal to the motor is sufficiently exciting. For our trials, the percent duty cycle of the PWM signal is given by a sinusoid sweep discretized at 100 Hz. The sinusoid sweep ramps linearly up, then down, in both amplitude (maximum: 6V) and frequency (from 0.125 Hz to 1.0 Hz). The carrier frequency of the PWM signal does not change, only the percent duty cycle.

We can use different metrics to evaluate how well the identified parameters fit the observed data to the chosen model. Mean Squared Error (MSE) is the mean of the square of the difference between the observed and estimated data using the identified parameters. The MSE can be calculated separately for the voltage balance (2.7) and torque balance (2.10). The coefficient of determination R^2 can be calculated:

$$R^2 = 1 - \frac{\sum_{n=1}^{N} \epsilon(n)^T \epsilon(n)}{\sum_{n=1}^{N} \boldsymbol{b}(n)^T \boldsymbol{b}(n)}$$

where $\epsilon(n)$ is the residual of the fit, $\epsilon(n) \triangleq \boldsymbol{b}(n) - \hat{\boldsymbol{b}}(n) = \boldsymbol{b}(n) - A(n)\boldsymbol{x}$.

4.6 Acknowledgements

Part of this chapter concerning Switchblade has appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, “Switchblade: an agile treaded rover,” *Proc. IEEE/RSJ Int’l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the DC motor dynamometer has appeared in print: N. Morozovsky, R. Moroto, T. Bewley, “RAPID: an inexpensive open source dynamometer for robotics applications,” *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the perching maneuver has been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Prototypes

5.1 Manufacturing Processes

Over the course of the design, manufacture, and assembly of dozens of robotic prototypes, a variety of manufacturing processes were used, each with its own advantages and disadvantages. Traditional machining, with a mill or lathe, can be used for high precision parts in a variety of materials (metals and plastics), but is time-consuming. The process can be automated somewhat with computer numerically controlled (CNC) hardware, but CNC mills and lathes are expensive tools which aren't always readily available. CNC laser cutting, which like traditional machining is a subtractive process, is fast, though it can only cut out two dimensional parts. A large sheet of material can be cut into several parts sequentially and automatically. The adoption of tab and slot assembly structures, pioneered by Switchblade (see Sec. 5.5) revolutionized how robots were designed and built in the Coordinated Robotics Lab (e.g. iCycle, iceCube v.2). Acrylic and delrin are two common materials used with a laser cutter (other thin materials, wood, poster board, etc. may also be used), delrin is less brittle than acrylic and is easier to machine for operations such as tapping threads in a hole. 3D printing, also known as additive manufacturing, has long been a means of rapid prototyping, but only recently (c.2010) has the market been disrupted with consumer level machines ($\mathcal{O}(\$1000)$). The acquisition of one, and then two, 3D printers in the lab revolutionized the manufacturing process of robotic prototypes. Switchblade nano (Sec. 5.5.4) was the first prototype made from parts created on the in-house, consumer grade (single extruder) finite deposition modeling (FDM) 3D printer. eyeFling (Sec. 5.4.2), SkySweeper (Sec. 5.6), and Switchbot (Sec. 5.5.5) are also made from 3D printed parts, as are nearly all of the other modern prototype robots in the lab. The dyno (Sec. 5.3) and myMIP (Sec. 5.4.1) are both examples of hybrid assemblies which use 3D printed and laser-cut parts together, where the use of the laser cutter saved significant manufacturing time. Other than slow printing speed, the main limitation of consumer-grade 3D printers is material selection. The two most common materials are acrylonitrile butadiene styrene (ABS) plastic and polylactic acid (PLA) plastic. ABS is less brittle, but requires a heated print area and is more prone to warping from thermal stress, making PLA the more user-friendly option with a higher successful print rate. Another critical factor for printing parts with an FDM machine is the part orientation. The nature of layer-by-layer FDM printing inherently creates anisotropic parts. Components with stress concentrations need to be oriented to minimize the chance of layer delamination.

5.2 Electronics

Just as the proliferation of low-cost 3D printing shifted the paradigm of robot prototype manufacturing, the explosion of low-cost embedded electronics has enabled these robots to be programmed with capable microprocessors and a variety of sensors. The variety of different embedded controllers are programmed with different languages and environments.

- Arduino AVR microprocessors, programmed in a user-friendly c-like language (used in SkySweeper and myMIP).
- National Instruments myDAQ USB data acquisition devices which can be programmatically sampled in the LabVIEW graphical programming language (used in the DC motor dynamometer).
- National Instruments RIO (reconfigurable input/output) series (single board

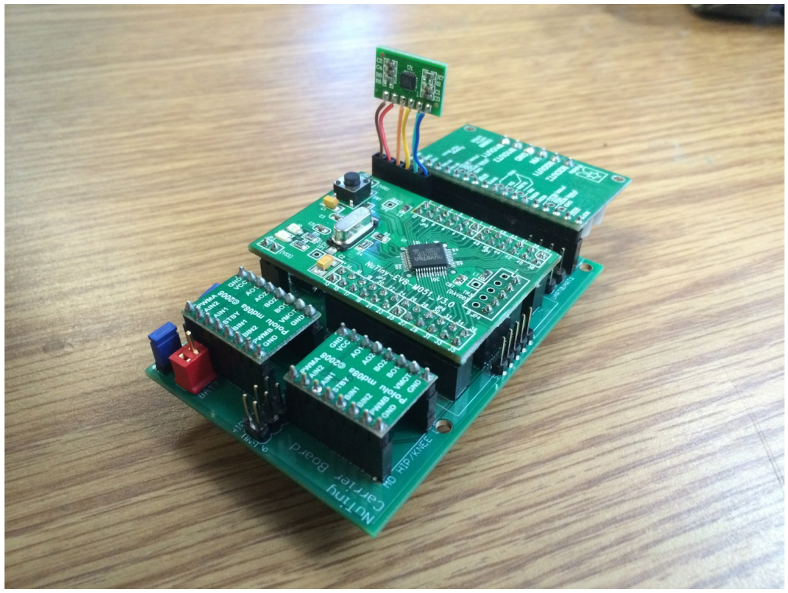


Figure 5.1: Switchbot development electronics with low-cost ARM processor.

RIO, compactRIO, myRIO) which feature a field programmable gate array (FPGA) as well as a traditional microprocessor with a real-time operating system (RTOS), either a Freescale PowerPC processor or a Xilinx Zynq System-on-Chip (SoC) which are both programmed in the LabVIEW graphical programming language, which also has textual programming modules (used in Switchblade, Switchblade nano, and Switchbot).

- bStem board developed by Brain Corporation, powered by the Qualcomm Snapdragon SoC, which can be programmed quickly in Python (used in eyeFling).
- 32-bit ARM Cortex M0 processor, available in a low-cost ($<$ \$1) package, which is programmed in c (used in the tether-less version of Switchbot).

Microelectromechanical system (MEMS) sensors, namely accelerometers, gyroscopes, and magnetometers have drastically reduced in size and price, largely due to their proliferation in smartphones. A variety of online retailers sell sensor and motor driver modules on breakout circuit boards that are easy to connect to microcontrollers. Once the electrical modules are selected, it is straightforward to design a custom circuit board which holds and appropriately connects the modules along with other peripherals (switches, fuses, voltage regulators, communication modules). This method eliminates the "rat's nest" of wires common in early robotic prototypes and also decreases size and increases reliability. The electronics package used in Switchbot, comprised of an ARM Cortex M0 processor, three dual motor driver modules, a six axis inertial measurement unit, switches, a voltage regulator, and connectors for motors, encoders, and potentiometers, is shown in Fig. 5.1.

Lithium polymer batteries have been used extensively for their excellent energy density and capacity to deliver high amounts of current. In situations where charging cannot be tightly controlled, more stable battery chemistries (e.g. nickel metal hydride, NiMH) may be preferable.

5.3 DC Motor Dynamometer

The design of RAPID was driven by two main factors: first, being able to adapt to a wide variety of motors; and second, minimizing cost and complexity. For ease of use, we impose the requirement that it take less than 15 minutes to change the motor under test.

5.3.1 Mechanical System

The core function of a dynamometer is to exert a known load on the motor under test. To minimize cost and retain simplicity, the load in RAPID is provided by an inertial disc instead of an active or passive braking system. Friction in the system is an additional load on the motor.

Depending on the power of the motor under test, a larger or smaller inertial load is required. In order to accommodate a range of motor specifications, the inertial disc has eight threaded radially symmetric holes, each capable of holding a single hex head bolt and up to three optional nuts. The removable nuts and bolts can be added in symmetric pairs to the disc to provide 35 different discrete inertial loads between $3.37 \cdot 10^{-5}$ kg·m^2 and $1.31 \cdot 10^{-3}$ kg·m^2 (see Sec. 2.1).

The inertial disc is rigidly mounted with a clamping hub to an aluminum shaft, which is supported by ball bearings on either end. One end of the shaft is

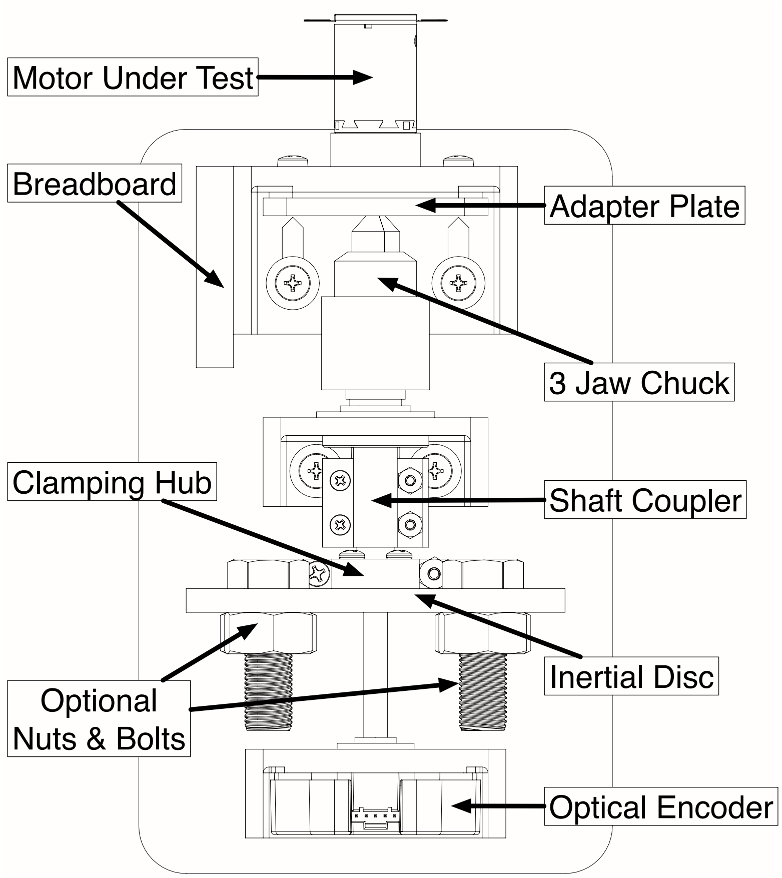


Figure 5.2: Labeled overhead drawing of RAPID assembly.

coupled to the rear of a Jacobs three jaw drill chuck, Fig. 5.2. The jaws of the chuck are used to hold the output shaft of the motor under test. Most motor output shafts possess a round cross-section, possibly with a flattened section or a keyway, nominally used to prevent the shaft from slipping. With the drill chuck, RAPID can hold most round shaft cross-section shapes so long as they possess three points of contact 120° apart that are equidistant from the center of the shaft's rotation. The chuck can accommodate shaft diameters ranging from 1.00 mm to 6.35 mm (1/4 in). If the output shaft does not meet these requirements, an adapter can easily be made with a 3D printer that presses onto the output shaft and has a circular protrusion onto which the chuck can grasp. Since the weight of the inertial disc is supported by bearings, there is minimal radial load on the motor shaft.

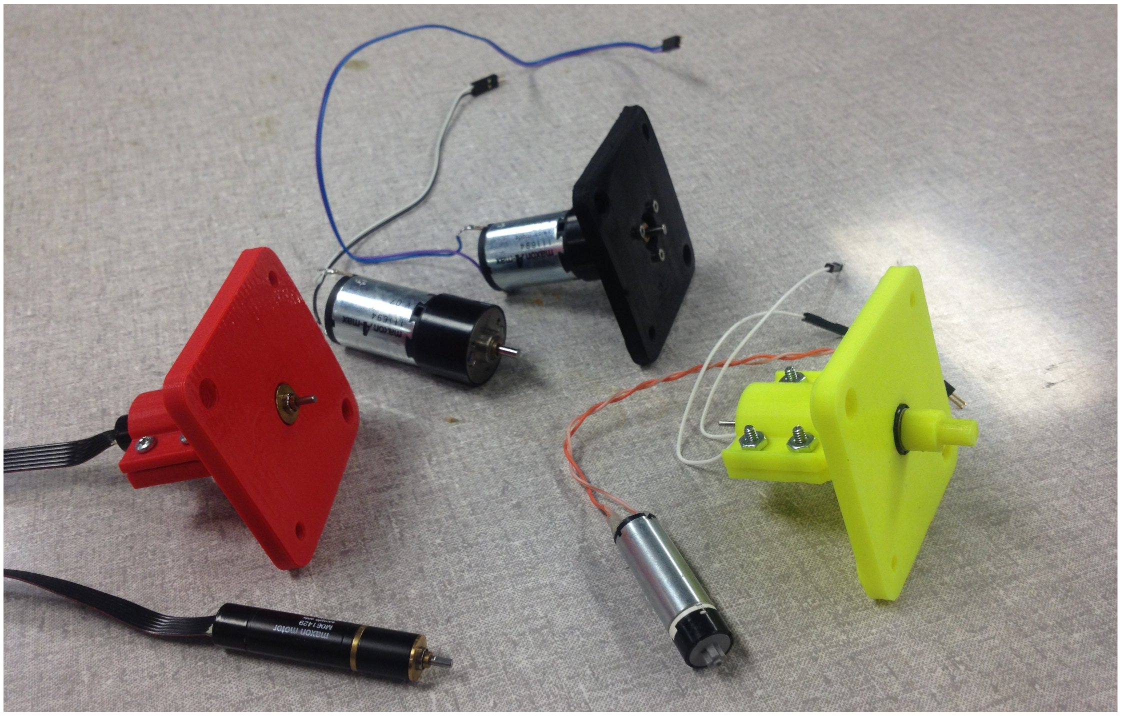


Figure 5.3: Adapter plates for three different motors. Face-mounted screws (center), clamp for round motor housing only (left), and with added shaft adapter (right).

Different motors may have different mounting features. RAPID is designed to be modular and adaptable. The motor under test mounts to an adapter plate with features specific to the motor, such as face-mounted threaded holes or flat features on the motor body, or the adapter plate may clamp around the entire motor body, Fig. 5.3. These adapter plates can be manufactured on a 3D printer, laser-cutting machine, or other machinery depending on the motor mounting features. A set of modifiable drawings of adapter plates with different common mounting features is available at the aforementioned web site and in Supplemental File 5: DC motor dynamometer files.

5.3.2 Electrical System

The electrical system is responsible for delivering power to the motor and measuring several physical values. The sensors, motor driver, and data acquisition

Table 5.1: Motor Dynamometer Electronics

Use	Make & Model
Position Sensor	US Digital E6, 2500 CPR
Voltage Sensor	2:3 resistive voltage divider circuit
Current Sensor	Allegro ACS712 ELC-05B-T Hall Effect
Motor Driver	Toshiba TB6612FNG
Data Acquisition	National Instruments myDAQ

device used are listed in Table 5.1, a wiring diagram is shown in Fig. 5.4. Data is sampled and logged at 100Hz on all sensors. Two National Instruments myDAQ devices are used to interface the sensors and motor driver over USB with a host computer running LabVIEW software. The two myDAQ devices need not be dedicated to RAPID and can be used for a variety of applications such as a software multimeter or oscilloscope. Other data acquisition devices could also be used by simply adapting the software. The dedicated electronics are all inexpensive.

An optical quadrature encoder is mounted on the end of the inertial disc shaft opposite to the motor under test. By counting the rising and falling edges and comparing the phase delay of the two channels of the encoder, the rotational position can be measured with a precision of four times the resolution of the encoder disc with the counter circuit on the myDAQ [58]. The encoder disc has 2500 counts per revolution, so the resolution with quadrature is $6.28 \cdot 10^{-4}$ radians. The counter circuit can read a maximum frequency of 1 MHz [59], which corresponds to a maximum rotational speed of 628 rad/s.

RAPID is designed to be compatible with different motor drivers. The motor driver used in the initial testing is listed in Table 5.1 and is connected to an external power supply. A pulse width modulated (PWM) 5 V square wave is generated by the timer circuit on the myDAQ [60]. The carrier frequency is 32 kHz and the percent duty cycle is changed at a rate of 100 Hz. The motor driver amplifies the 5 V PWM input signal to the full voltage of the external power

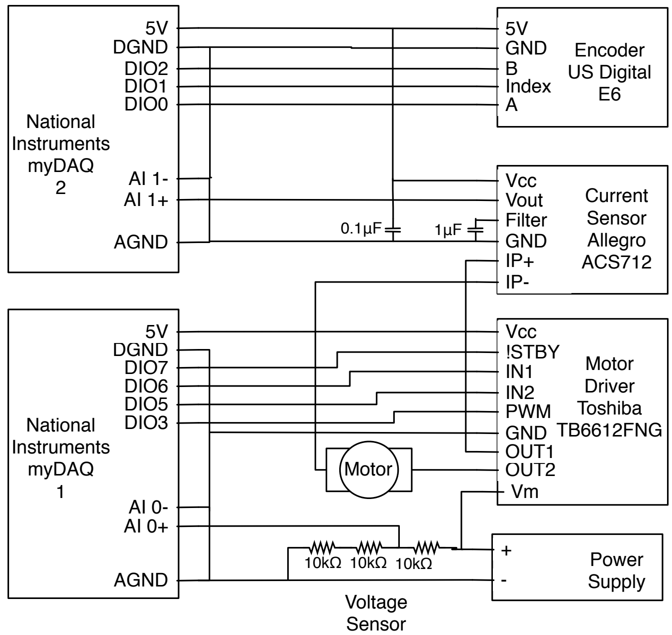


Figure 5.4: Wiring diagram for DC motor dynamometer with National Instruments myDAQ devices.

supply. The signed duty cycle of the PWM signal (including direction), $u \in [-1, 1]$, multiplied by the voltage of the power supply, V_S, is the average voltage across the motor terminals, $\overline{V} = V_S u$. We can not directly measure the voltage across the motor terminals because the PWM carrier frequency is significantly higher than the sampling frequency. Instead, we measure the voltage output of the external power supply, which may be higher than the maximum measurable voltage of the Analog to Digital Converter (ADC), 10 V. We use a simple voltage divider circuit made from off-the-shelf carbon film resistors to step down the voltage. The input of the ADC is connected to an intermediate node of the circuit. The actual voltage can be recovered by multiplying by the ratio of the resistors in the voltage divider circuit. Both the signed PWM duty cycle and power supply voltage are logged, so the average voltage can be calculated by multiplying these two values together.

A current sensor is placed in series between the motor and motor driver. The bandwidth of the sensor itself is 80 kHz, which is more than twice the PWM carrier frequency of the motor driver. However, we are only sampling the sensor at 100 Hz. We implemented a passive first-order low-pass filter with $\omega_c = 93$ Hz in

Table 5.2: Results from Maxon A-max 22 motors

Parameter		Motor				
Measure		**1**	**2**	**3**	**4**	**5**
r	Ω	4.054	3.718	3.745	3.749	3.991
k	$N \cdot m/A$	0.363	0.362	0.357	0.359	0.365
J_E	$kg \cdot m^2$	$1.7{\cdot}10^{-3}$	$1.7{\cdot}10^{-3}$	$1.7{\cdot}10^{-3}$	$1.7{\cdot}10^{-3}$	$1.7{\cdot}10^{-3}$
b	$N \cdot m/s$	$8.7{\cdot}10^{-4}$	$9.9{\cdot}10^{-4}$	$1.1{\cdot}10^{-3}$	$8.4{\cdot}10^{-4}$	$9.5{\cdot}10^{-4}$
c_M	$N \cdot m$	$3.7{\cdot}10^{-3}$	$4.9{\cdot}10^{-3}$	$6.2{\cdot}10^{-3}$	$5.1{\cdot}10^{-3}$	$5.5{\cdot}10^{-3}$
MSE	V	0.101	0.092	0.099	0.087	0.096
	$N \cdot m$	$2.5{\cdot}10^{-4}$	$2.9{\cdot}10^{-4}$	$2.9{\cdot}10^{-4}$	$2.8{\cdot}10^{-4}$	$2.5{\cdot}10^{-4}$
	R^2	0.994	0.994	0.994	0.994	0.993

hardware on the output of the sensor to eliminate high frequency noise from the PWM signal. We limit the maximum frequency of the PWM duty cycle change to well below 50 Hz to ensure there is no aliasing.

5.3.3 Experimental Validation

We tested five of the same high-quality motor, the Maxon A-max 22 (5W, 6V) with a 64:1 spur gearbox [61, 62]. The specifications from this manufacturer are quite detailed and we can thus compare the identified parameters to the specifications to determine the accuracy of RAPID. Comparing the results of the five motors indicates the precision.

The expected torque from this motor is on the order of 10^{-2} N·m, we thus add four bolts to the inertial disc to increase the total inertia. Average results from twenty trials of each motor are shown in Table 5.2. A representative plot of the measured and estimated voltage is shown in Fig. 5.5 and a plot of both sides of (2.10) with the estimated parameters is shown in Fig. 5.6.

The specifications list a k value of 0.377 N·m/A (accounting for the gear-

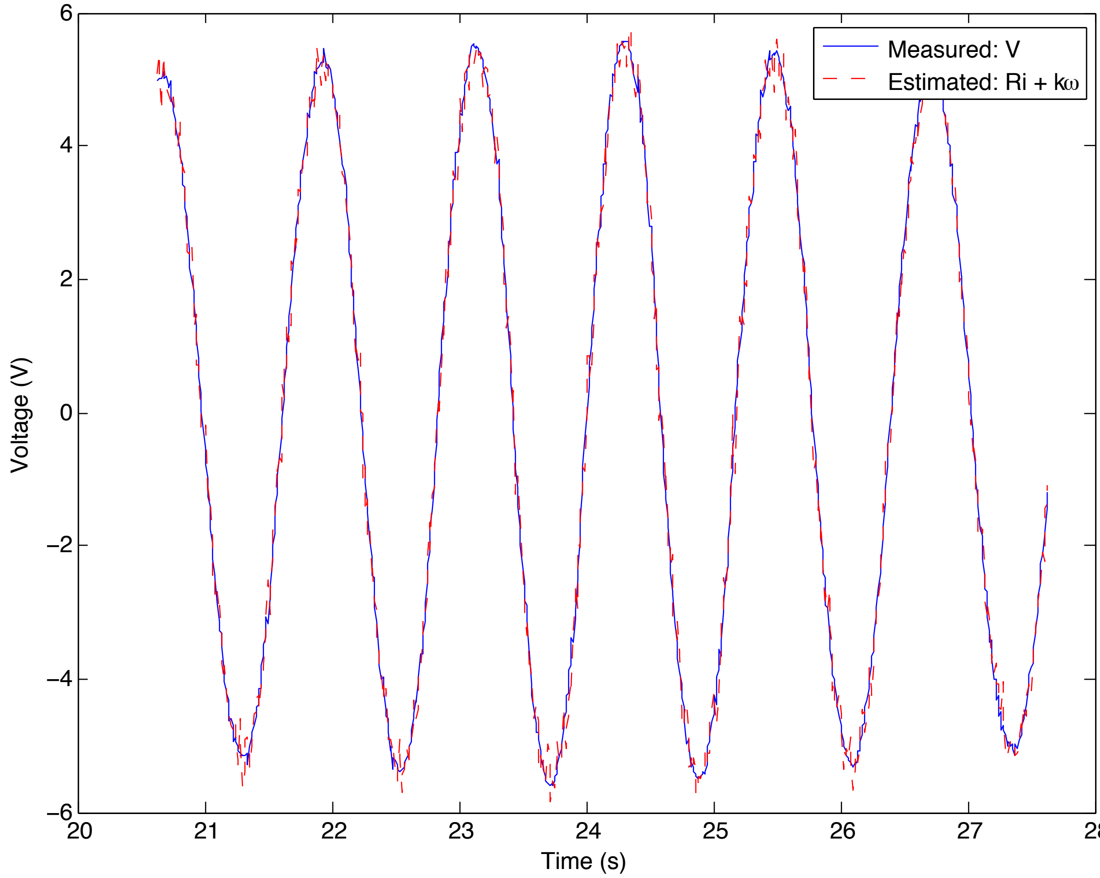


Figure 5.5: Measured and estimated voltage [see (2.7)].

box); the value estimated by RAPID for each of the motors is within 5% of this value. Given that we can expect a manufacturing tolerance of 1-3%, the accuracy of the tests is acceptable. The efficiency of the motor driver may also account for the deviation from the specification, particularly since the values are consistently lower than the specification. The motor driver also affects the estimation of the resistance of the armature wire. The motor driver H bridge has an "on" resistance on the order of 1 Ω, in addition to the resistance in the wire and breadboard. The motor specification is 1.76 Ω, which is less than the identified resistance. The estimated effective inertia of the motor and gearbox is within 2% of the specifications. Comparing parameter estimates from motor to motor shows acceptable precision as well. We see the most motor-to-motor variation in the friction parameters. Slight differences in alignment while mounting the motor in RAPID may account

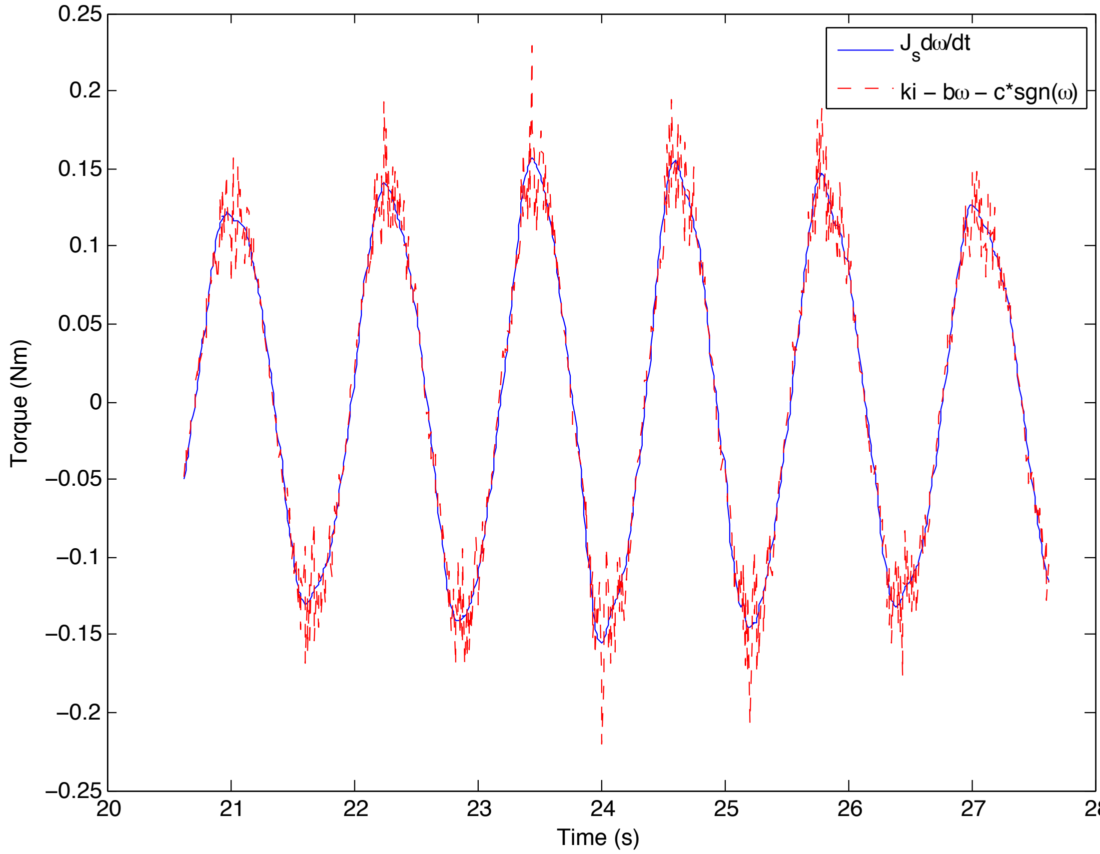


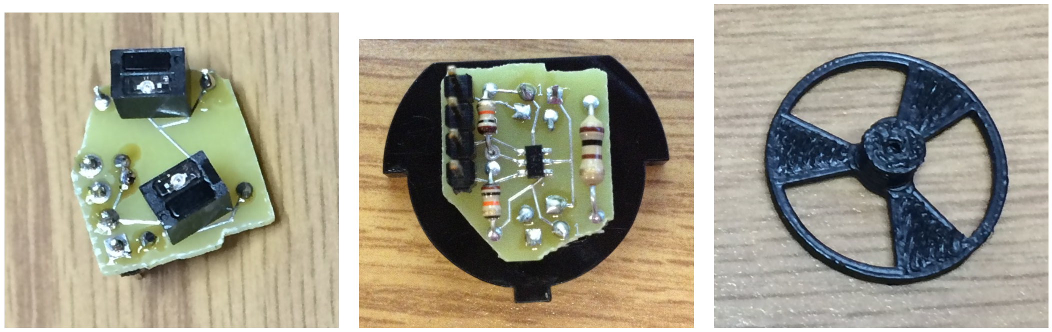
Figure 5.6: Torque balance with estimated parameters [see (2.10)].

for this. MSE values are within 2% of full scale and R^2 values are close to unity and compare favorably with other motor parameter identification results in [46], [63].

5.4 Mobile Inverted Pendulum

5.4.1 myMIP

The design of myMIP was optimized for low-cost and ease of assembly to facilitate student ownership at a price comparable to a textbook. Several off the shelf components were used: motors, wheels, microcontroller (Arduino nano), accelerometer, gyroscope, and motor driver. A 3D printed motor mount holds the two motors coaxially (in opposite orientations) and provides mounting points for



(a) Front. (b) Back. (c) Encoder Disc.

Figure 5.7: myMIP custom quadrature encoder.

the laser cut parts which hold the electronics. The cost of off the shelf encoders was too high and so I developed custom quadrature encoders comprised of a custom circuit board and a 3D printed encoder disc mounted on the back of the motors (before the gear reduction), Fig. 5.7. The circuit board has two infrared LED and phototransistor pairs on one side (Fig. 5.7a), and current limiting and pull-up resistors, a Schmitt trigger with hysteresis to debounce the signal, and a connector on the other side (Fig. 5.7b). The LED and phototransistor pairs can detect the transitions between the gaps and solid portions of the encoder disc (Fig. 5.7c). The LED and phototransistor pairs are placed 210° apart (90° + 120° for the 3 count per revolution disc) such that the two channels are out of phase for quadrature decoding. The width of the gap of the encoder disc was tuned to give equal high and low times.

5.4.2 eyeFling

The eyeFling prototype (Fig. 1.2b) was designed to be completely 3D printed, with off the shelf screws and motors. Similar to the earlier iFling prototype, eye-Fling can pick up standard 40mm ping pong balls by running them over, at which point the central prow of the body forces them to either side, where they are pinched between the body and the wheel. Further rotation of the wheel lifts the balls up into the body and past a one-way spring check valve. Ping pong balls can be dispensed one at a time (limited by the actuation of a miniature servo motor

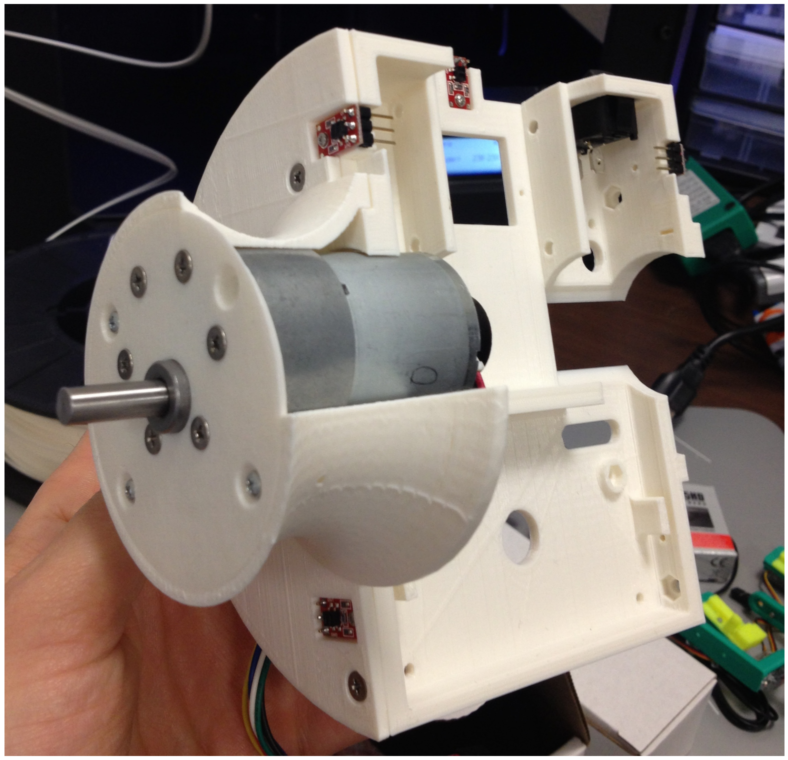


Figure 5.8: Partial eyeFling body assembly with one motor mounted.

with a gate) along the rail and thrown, lacrosse style. Unlike the earlier iFling, the parts of eyeFling were designed to be printed on a consumer-grade, single extruder 3D printer. The parts were designed to be secured with machine screws and captive nuts, for secure connections that can be assembled and disassembled multiple times, Fig. 5.8. Detection sensors (infrared LED and phototransistor pairs) are placed at several points on the body to keep track of ping pong balls entering and exiting the body. The battery is mounted high up on the rail in order to raise the center of mass for reasonable handling with inverted pendulum dynamics (approximately one wheel radius above the center of rotation, at the top of the wheel). The electronics (bStem controller board plus adCord motor control board) are mounted on the rear of the body behind a magnetic, removable cover. In the second revision, access holes were added to the side of the robot to allow the USB, HDMI, and power ports to be directly accessed without removing the bStem as well as a quick release tab to swap batteries.

5.5 Switchblade

5.5.1 Mechanical Design

The linchpin of this design is the hip joint that pivotally connects each tread assembly to the chassis of the robot. A novel, patent-pending two degree of freedom joint is introduced to connect the chassis to each tread assembly (Fig. 5.9). The joint combines the axis of rotation of the tread assembly with respect to the chassis and the axis of rotation of the sprocket driving the treads into one stainless steel shaft. The shaft rides in ball bearings in both the chassis and tread assembly. The end of the shaft inside the chassis is directly coupled to the output shaft of the planetary gearbox motor. The other end of the shaft passes through, and spins freely relative to, a large spur gear which is rigidly mounted to the tread assembly and the shaft is finally rigidly coupled to the tread drive sprocket. A second stainless steel shaft is mounted inside the chassis parallel to the first. A pinion gear, mounted on the end of the second shaft that extends from the chassis, drives the large spur gear, and hence the rotation of the tread assembly with respect to the chassis. The other end of the second shaft is directly coupled to the output shaft of a smaller planetary gearbox motor with a much higher gear reduction. With both motors mounted in the chassis, this joint independently transmits two coaxial torques: one to rotate the sprocket driving the treads and a second to rotate the tread assembly with respect to the chassis. Optical encoders (360 counts per revolution, CPR) are mounted within the chassis coaxially on both motor shafts. An additional high resolution (5000 CPR) optical encoder is mounted on the outboard side of the tread assembly to measure the relative rotation of the tread shaft. The higher resolution encoder enables better low-speed estimation, see section 4.2.1 for a detailed discussion. The actuation of the two degrees of freedom on each hip joint enable the robot to perform its unique suite of maneuvers. A set of passive, un-actuated wheels is mounted on the end of the chassis opposite the main drive axles. The wheels prevent the chassis from dragging on the ground when the tread assemblies are rotated higher than the chassis.

The diameter of the tread sprocket was chosen to give over 25 mm of ground

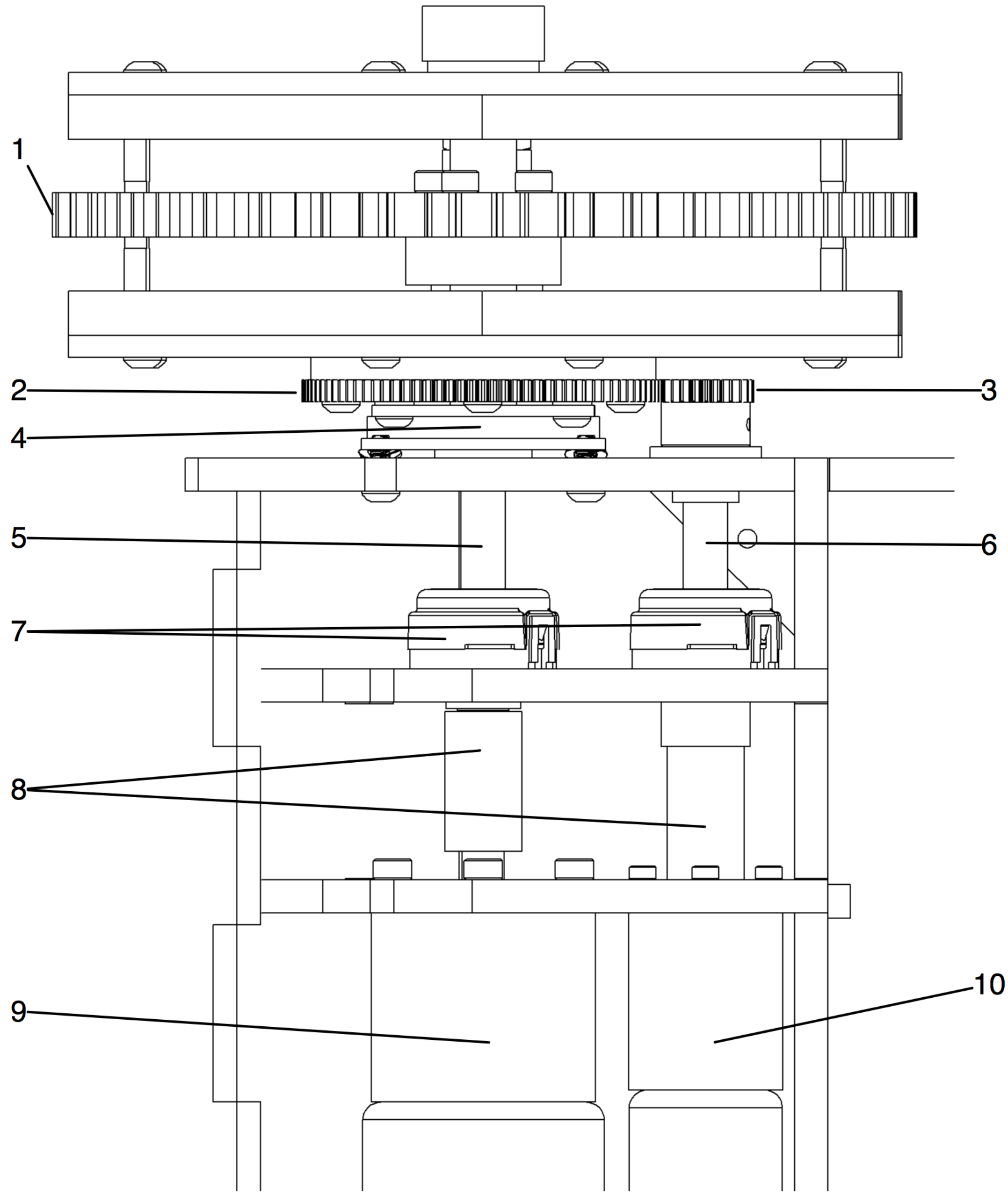


Figure 5.9: 2 Degree of freedom hip joint of the v.2 Switchblade: (1) tread sprocket, (2) spur gear, (3) pinion gear, (4) optional slip ring, (5) main drive axle, (6) rotation axle, (7) optical encoders, (8) shaft couplers, (9) tread motor, and (10) rotation motor.

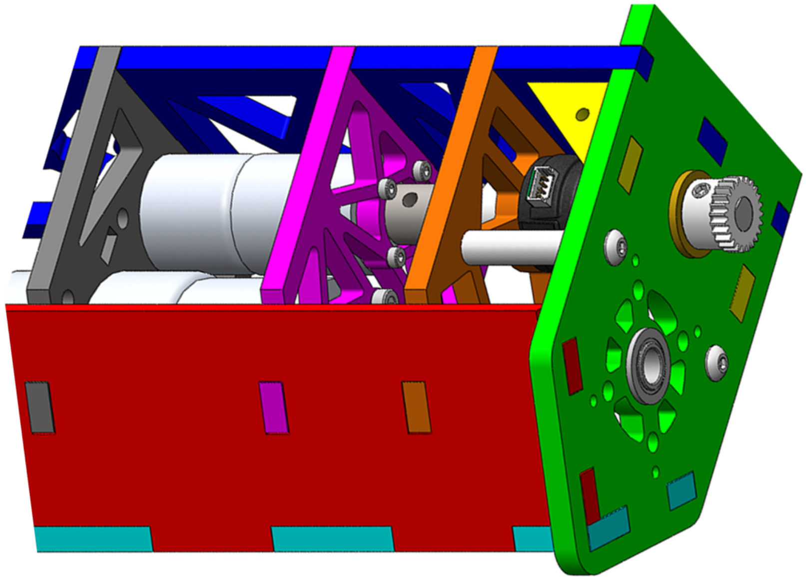


Figure 5.10: View of the hip joint. Tab-and-slot construction simplifies assembly and reduces the number of screws required.

clearance for the chassis when in a horizontal configuration, in conjunction with the motor and gearbox choice to have sufficient torque to lift the weight of the robot, and a top speed in excess of 2.5 m/s (6 body lengths per second). The traction between the treads and various ground surfaces is balanced between the need to grip while accelerating and the need to slip while skid-steering. It is critical to maintain traction while balancing upright, where a tread slipping may cause the robot to fall. The off-the-shelf treads (manufactured by VEX Robotics) are made of acetal and have a spray rubber coating to increase traction. The treads are continuously supported underneath, any large normal force on the treads will be transferred to the structure of the tread assembly (also acetal). The power loss incurred by the smooth sliding contact between the treads and tread assemblies is included in the dynamic model (2.33). The position of the rear idler sprocket is adjustable to appropriately tension the treads.

Great pains were taken to minimize the part count, particularly the custom part count, and to reduce the number of machining operations per custom part. Off-the-shelf parts were used wherever possible to reduce manufacturing time. All but two of the custom parts are laser-cut from sheets of acetal, with thread-tapping of some holes (thus avoiding the need for nuts) and press-fitting bearings being the only secondary machining operations. The remaining two custom parts are formed from stainless steel rod stock with simple operations on a lathe and milling machine. The symmetry of the design reduces the unique part count and many parts are orientation-agnostic, simplifying assembly. The 26 unique custom parts account for 73 pieces used in the assembly of each robot, with most parts being used in multiple places.

The design process included careful consideration of assembly time. The parts of the superstructure are quickly assembled with a series of interlocking tabs and slots (Fig. 5.10), thereby minimizing the number of mechanical fasteners needed and saving cost, weight, and assembly time. A team of five undergraduates working under our direction constructed 13 Switchblade robots from part manufacturing to final assembly in 10 weeks.

Many system parameters (masses, lengths, etc.) were directly measured or taken from the 3D CAD model. Other parameters (stall torques, back EMF damping coefficients, rotational inertias, coefficients of friction) were determined empirically with an *in situ* characterization. We used a least squares algorithm similar to what is presented in [50].

Embedded Electronics

The Switchblade robot is built around the National Instruments sbRIO 9602 embedded controller. This board has both an FPGA and a PowerPC processor, which gives flexibility in handling both low-level high-speed tasks (such as reading optical encoders) and more complex control algorithms (as described in section 3.2) in real time, and is programmed using the LabVIEW graphical programming language, including the Control Design & Simulation and Robotics modules. Built-in ethernet coupled with a wireless ethernet adapter enables real-time wireless communication, debugging, and deployment of software. An AF-1501 frame grabber module from moviMED allows for onboard image processing. A 16-bit analog to digital converter reads the output of the analog sensors and monitors the battery voltage.

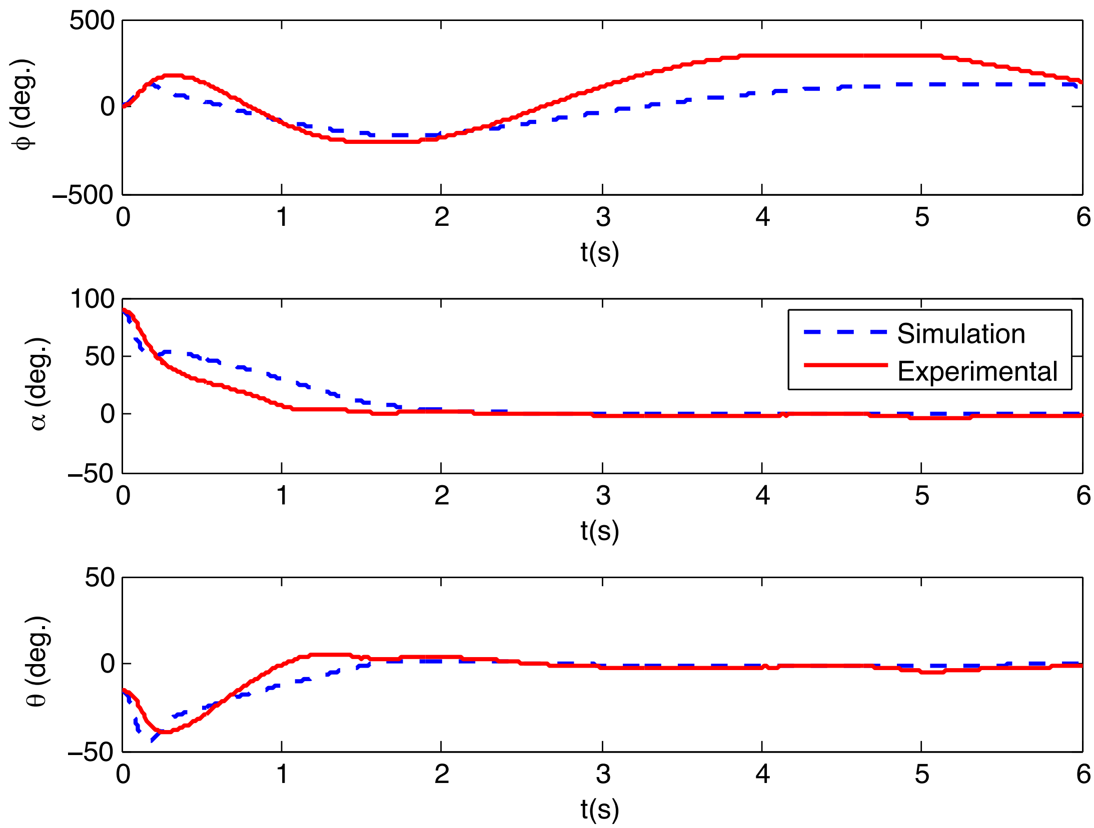


Figure 5.11: Comparison of simulation results and experimental results for the Switchblade uprighting maneuver.

5.5.2 V-Balancing Experimental Results

Both the simulation and experimental results may be seen in Fig. 5.11 for the uprighting maneuver as described in Sec. 3.2.1, plotting the critical angles as labeled in Fig. 2.3. The plot shows a close, though not perfect, correlation between simulation and experiment. The discrepancies may be explained by modeling simplifications and errors; in particular, the motor parameters are not particularly well characterized, and bias error in the measurement of the accelerometer could affect the drift of ϕ. Notably, the robot actually uprights faster in the experiment than in simulation. Supplemental File 1: Switchblade video shows the robot performing the uprighting maneuver as in Fig. 3.1.

Table 5.3: Results from horizontal stability tests.

	a_S	RMS $(\phi-\alpha)-(\phi^*-\alpha^*)$	RMS $(\dot{\phi}-\dot{\alpha})$	RMS u_1
Experimental	0	0.044	0.352	0.144
	0.06	0.028	0.169	0.167
Simulation	0	0.053	0.466	0.159
	0.06	0.038	0.326	0.161

5.5.3 Perching Experimental Results

The increased difficulty of the perching maneuver necessitated two physical changes to the Switchblade prototype. First, the tread motors were switched to maxon motors, with much smoother actuation due to a continuously woven rotor winding. Second, high resolution (5000 CPR) encoders were added to the outside of the tread assemblies to measure the relative rotation of the tread sprockets with respect to the tread assemblies with higher resolution than the 360 CPR encoders mounted in the chassis.

We first look at the performance of the system when stabilizing about a single equilibrium point, where $\alpha^* = \theta^* = \pi/2$. We use a narrow beam ($\rho = 0.011$m) as a step edge surrogate for testing. We can compare the balancing behavior with and without the friction compensator (3.10) by choosing $a_S = 0$ or $a_S = 0.06$. In both cases, $b_S = b_T = 0.04$ and $a_T = 0$ (since $\alpha^* = \theta^*$ the friction between the chassis and tread assemblies does not need compensation). Due to backlash in the treads and uncertainty in the velocity estimate, the $\dot{\phi}$ gain is decreased 40% to eliminate chatter.

As can clearly be seen in Fig. 5.12 and Table 5.3, the controller with the friction compensator has significantly reduced variance from the reference command $(\phi^* - \alpha^*)$ and the magnitude of the tread velocity $(\dot{\phi} - \dot{\alpha})$ peaks is greatly reduced without a significant increase in the control effort. The flat sections in the plot of

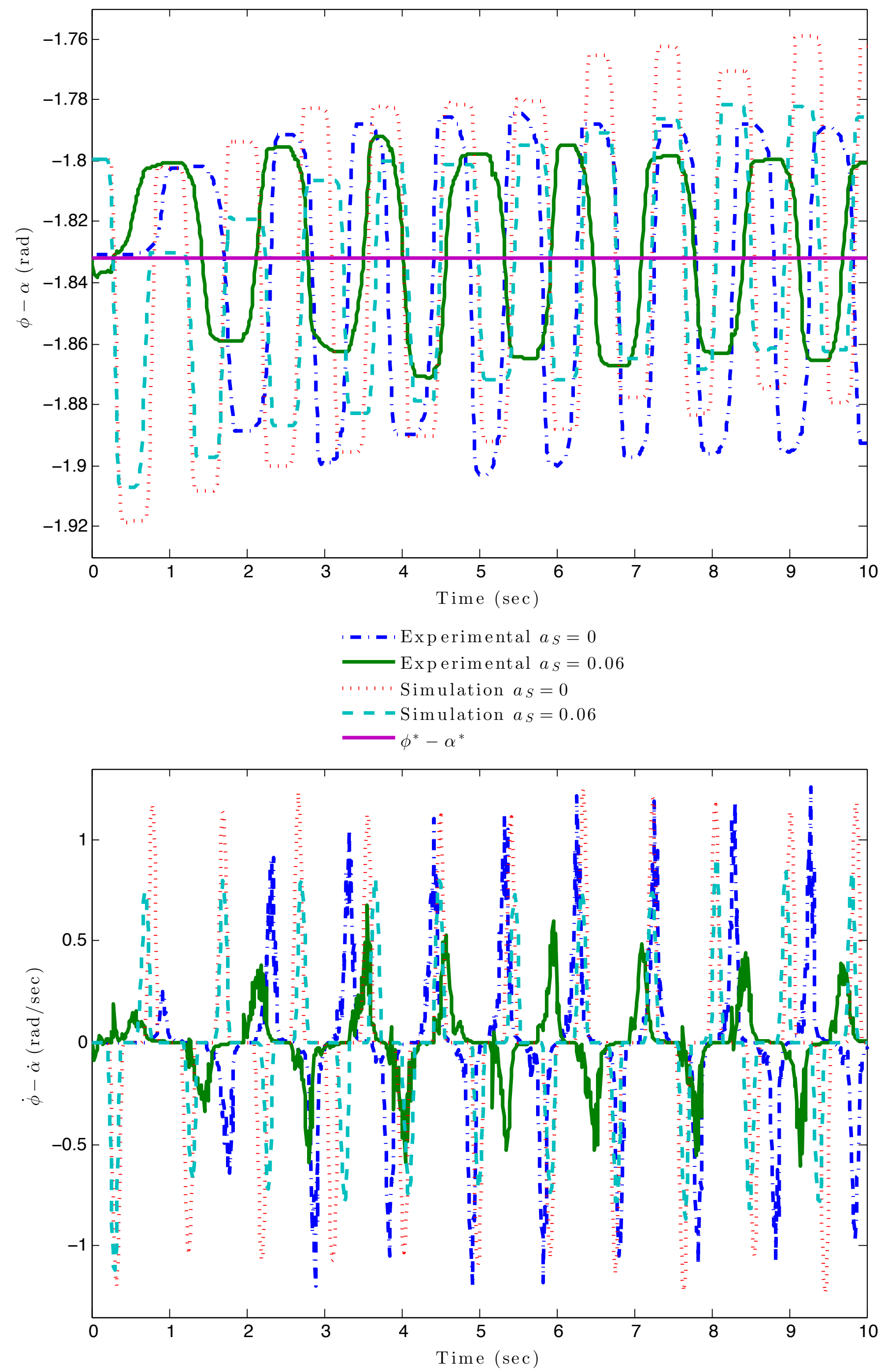


Figure 5.12: Experimental and simulation results for stabilizing at $\alpha^* = \theta^* = \pi/2$ with and without friction compensation.

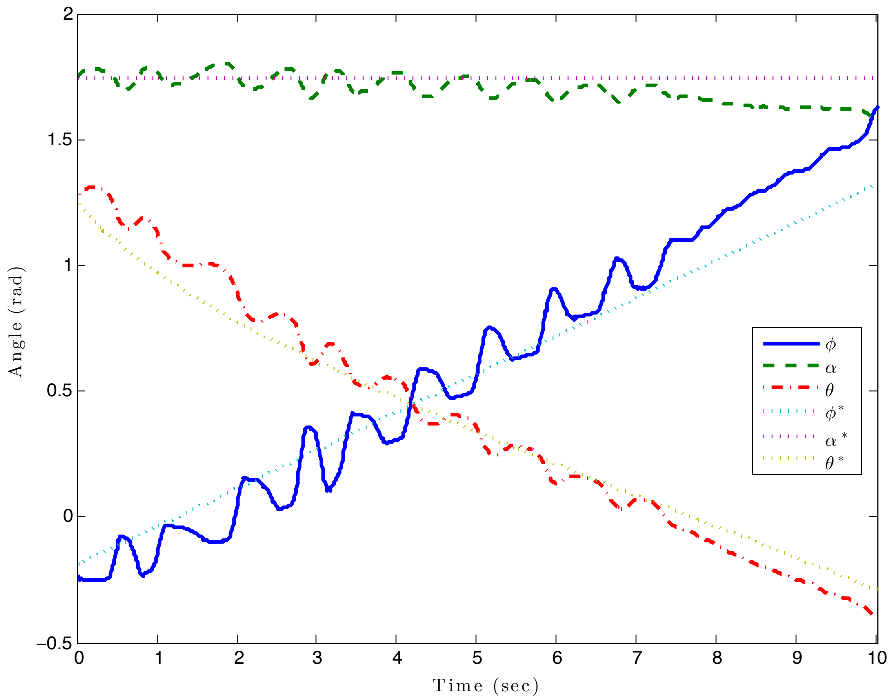


Figure 5.13: Experimental results for traversing with $\alpha^* = 190°$.

$\phi - \alpha$ correspond to when the treads are in stiction. Please refer to Supplemental File 1: Switchblade video to see horizontal stability tests with ($a_S = 0.06$) and without ($a_S = 0$) the friction compensator.

Given the mass distribution of the robot, climbing the edge of a standard step, while kinematically possible, see Fig. 3.4, is not practically feasible. The distance between the center of mass of the chassis and the tread assembly pivot point (L_C in Fig. 2.5) is less than half the length of the tread assemblies. This means that in order to shift the total center of mass from one end of the tread assemblies to the other, the inclination angle must be steep ($\alpha^* = 150°$). By (3.5), a significant amount of the torque available from the tread motor is used by the feedforward term and not available for error regulation (in one direction). This problem is exacerbated as the height of the center of mass increases, because more torque is required to recover from a given disturbance. The amount of stiction and friction between the treads and the tread assemblies further limits the control authority since the friction limits the amount of torque that can be used for correction, and there is zero control authority when in stiction. Worse, the normal force on the treads increases with α (2.33) which increases the friction and stiction. With the current prototype build, only a limited traverse is possible. Logged data from a 10 cm traverse at an inclination of 10 degrees from horizontal ($\alpha^* = 100°$) over 10 seconds is shown in Fig. 5.13, the maneuver is also shown in Supplemental File 1: Switchblade video. The oscillatory behavior is due to the treads entering and exiting stiction. The steady state error is due to sensor and parameter errors.

5.5.4 Switchblade nano

The goal of this prototype was to shrink the Switchblade design as much as feasible as an investigation into possible commercialization. As a surrogate for designing injection molded plastic parts, 3D printing was extensively used. The motors, sensors (accelerometer, gyroscope, and encoders), and battery were all mounted in the chassis, but the processing electronics were off board and connected via a tether. This sped up development by avoiding the need to design and fabricate a custom circuit board small enough to fit inside the chassis and allowed the use of the same sbRIO 9602 control board as used in the full-sized Switchblade, which enabled maximum code reuse.

Symmetry

All parts (except for the battery clip) are designed to be used twice symmetrically in the robot. The motivation was to minimize the part count so fewer molds have to be made. One chassis half part is used as the top and a second chassis half part is used as the bottom. The left and right tread assemblies are identical, one is attached to the top chassis half and the other is attached to the bottom chassis half. Each chassis half is designed with mounting holes for the electronics and battery holder, but in practice, only one chassis half will actually have the electronics and battery holder mounted, see Fig. 5.14. There is only space for one chassis half to have the electronics and battery holder mounted.

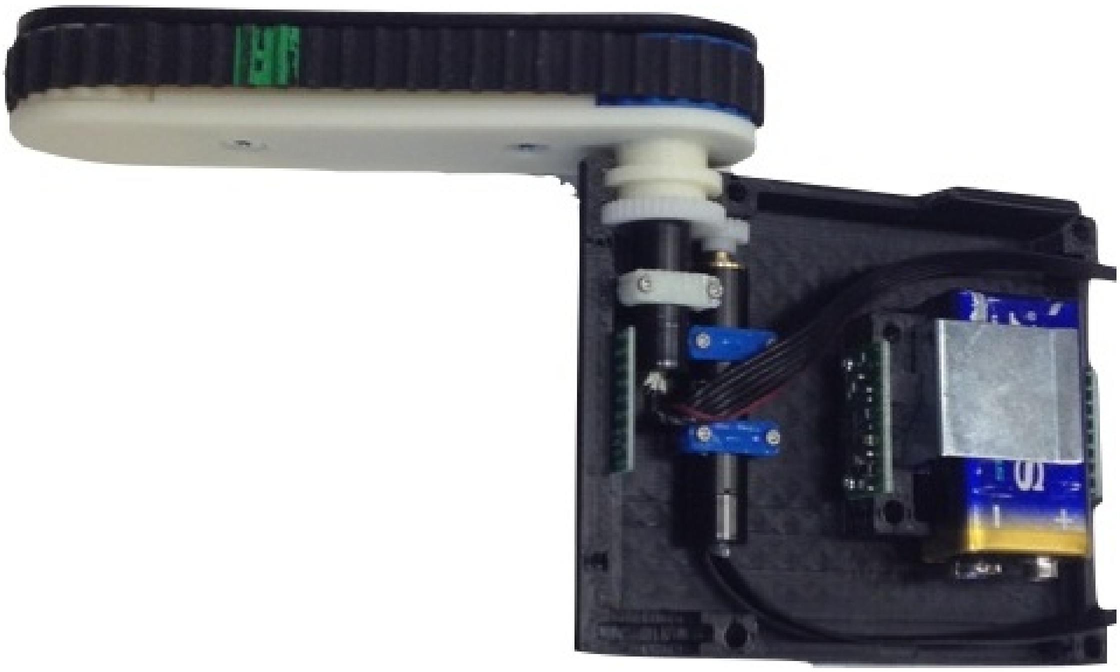


Figure 5.14: Switchblade nano symmetric half assembly with electronics installed.

3D Printing

The 3D printed part drawings are all toleranced to work with the lab's 3D printers. This means:

- Parts are oversized by 0.030 inches in the direction of printing.
- Hole diameters are oversized by 0.020 inches.
- The bottom printed face is chamfered by 0.020 inches.

These features would have to be removed or changed for an alternate manufacturing process.

Hip Joint

The most critical mechanical component of the robot is the hip joint, where the tread assembly connects to the chassis. The hip joint was redesigned from the ground up for the new size and with plastic parts instead of steel. The tread assembly can rotate continuously with respect to the chassis. All four motors in

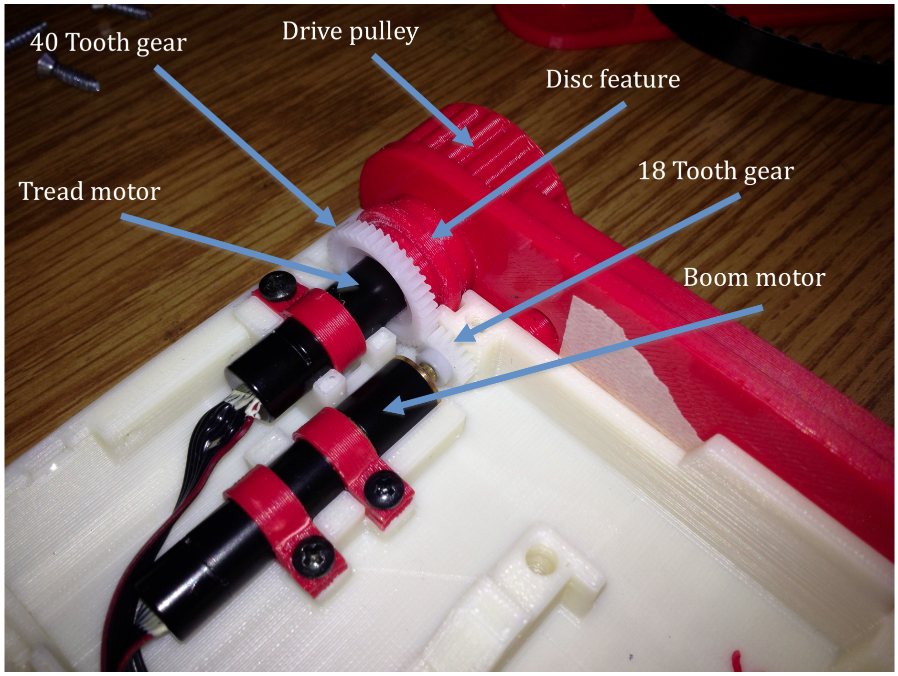


Figure 5.15: Switchblade nano hip joint.

the robot are mounted in the chassis. One motor on each side (left/right) drives the treads forward and back. One motor on each side (left/right) rotates the entire tread assembly with respect to the chassis. The pulley that drives the treads has the same axis of rotation as the entire tread assembly. All components are labeled in Fig. 5.15. The 40 tooth gear is rigidly connected to the tread assembly. The tread motor (which drives the pulley which drives the treads) passes through the 40 tooth gear and rotates freely inside the 40 tooth gear. The drive pulley is pressed on to the tread motor shaft. The 40 tooth gear is driven by the 18 tooth gear. The 18 tooth gear is pressed on to the output shaft of the boom motor. When the boom motor turns, the entire tread assembly rotates with respect to the chassis. The tread assembly has a disc feature which sits in a circular slot in each half of the chassis. This disc feature constrains the tread assembly to the chassis except for one rotational degree of freedom.

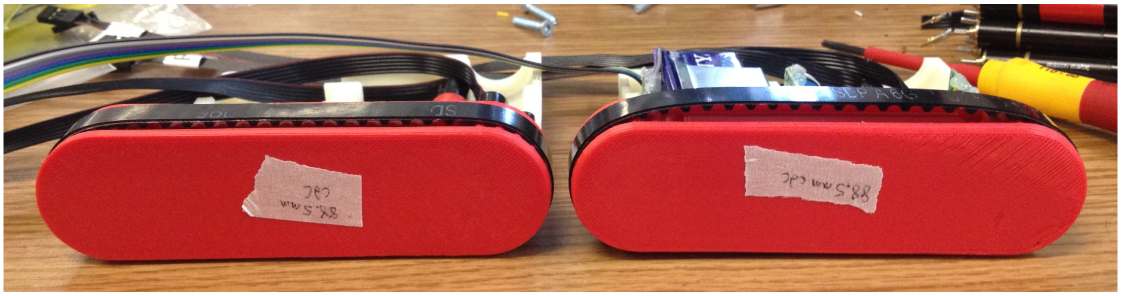


Figure 5.16: Switchblade nano belt tension.

Belt Tension

The belt tension is an important parameter. If the belt tension is too tight, there will be too much rolling resistance and the motor will not be able to turn the treads quickly. If the belt tension is too loose, the robot may not be able to balance on the far end of the treads (C-balancing) because of the slack in the belt. In the extreme case, the driving pulley may slip, also causing the robot to fall.

In the current prototype, there is no active tensioning mechanism, as that would increase cost. Instead, the exact center-to-center distance of the two pulleys was adjusted to properly fit the belts on hand. If the manufacturing tolerance in the length of the belt is small enough, then this strategy can be used.

In Fig. 5.16 there is an example of a belt that is slightly too tight (left) and a belt that is slightly too loose (right). Both left and right belts should have the same tension, otherwise the tighter belt will turn less because of the increased rolling resistance and the robot will appear to pivot about the tighter belt.

Another important consideration is the stiffness of the belt. If the belt is difficult to deform, then even if the belt tension is low, the motor will not be able to turn the belt quickly. The belt must be supple and easily bent.

Tread Inner Plate Assembly

In the current prototype, there are four individual pieces which could be combined into one injection-molded plastic part. The four pieces are distinct in this prototype due to limitations in our lab's 3D printer. Fig. 5.17 below shows these four pieces assembled and glued together.

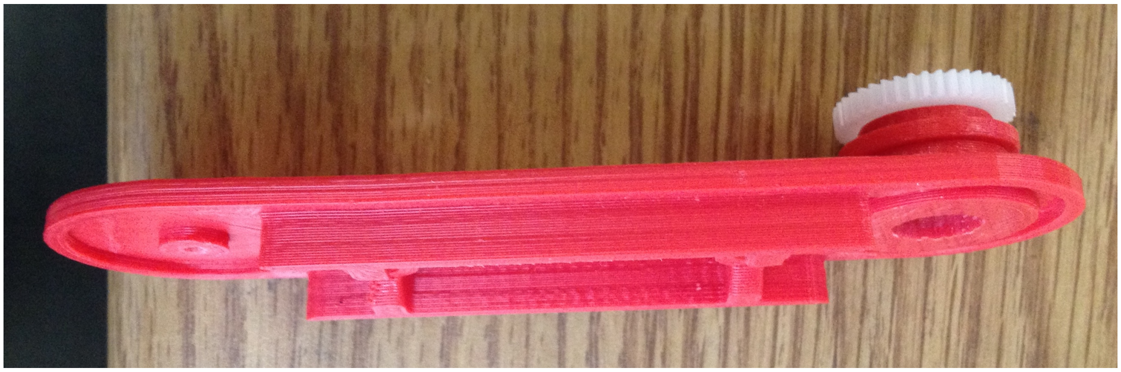


Figure 5.17: Switchblade nano tread inner plate assembly.

Performance

The Switchblade nano prototype was able to reproduce several of the primary maneuvers of the full-sized prototype.

- Horizontal, skid steer driving
- Overcoming thresholds
- V-balancing, including transitioning to and from horizontal mode
- C-balancing, although the tether physically prevented transitioning to and from horizontal mode

Supplemental File 2: Switchblade nano video contains demos of the maneuvers and a teardown. Perching was not attempted with the Switchblade nano prototype.

5.5.5 Switchbot

The goal of the Switchbot project was to put the treaded inverted pendulum technology from Switchblade into a more anthropomorphic form factor. Adding a link between the tread assembly and the main chassis (to create a two segment, human-like "leg") necessitated a complete redesign. It was deemed infeasible to power the treads with a motor located in the chassis and a drivetrain going through both an actuated hip joint and an actuated knee joint. The tread motor was then placed inside the tread assembly (Fig. 5.18), with a 2:1 bevel gear on the output

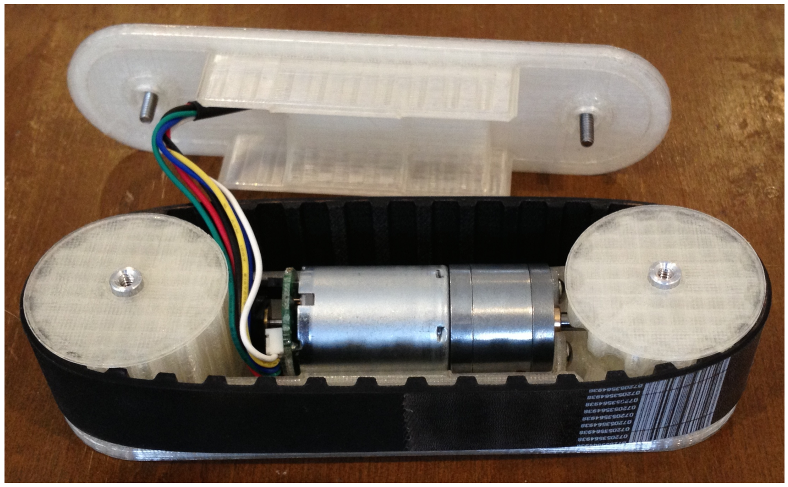
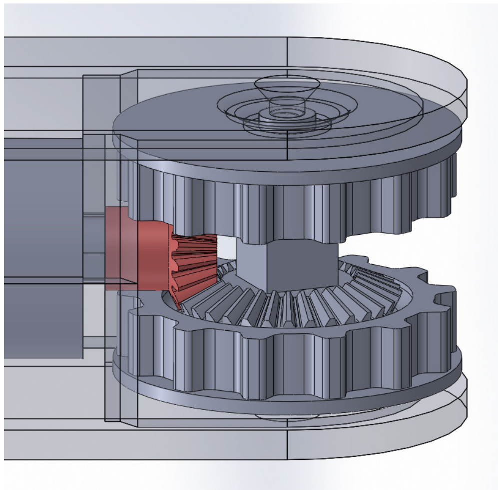


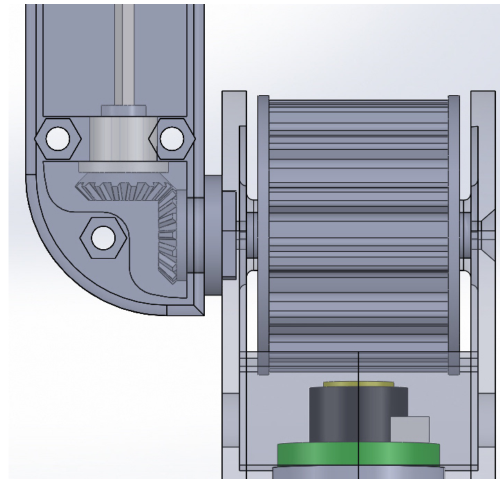
Figure 5.18: Internal view of the Switchbot prototype tread assembly.

of the gearbox to turn the drive pulley (Fig. 5.19a). The wires for the motor (and quadrature encoder) were routed external to the body. Since a more human design was intended, support for continuous rotation without tangling wires (with slip rings) was not necessary. For rapid prototyping purposes, all components are either 3D printed or available off the shelf.

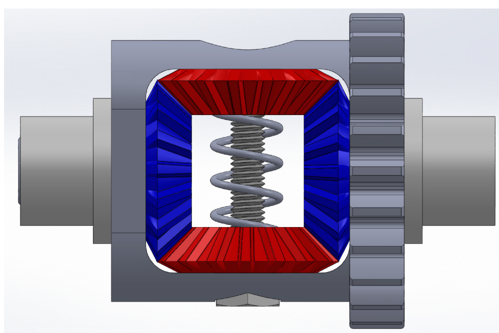
One of the primary design considerations was minimizing cost, and therefore motor count. The first round of prototypes utilized four motors: one each for the right and left treads, one for both hips, and one for both knees. Instead of rigidly fixing the right upper leg to the left upper leg, and the right lower leg to the left lower leg, I designed and built a novel, 3D printed, spring clutch limited slip differential (SCLSD), see Fig. 5.19c. The spring in the differential presses the two planet gears against the carrier, increasing the static and kinetic friction that resist relative rotation between the right and left driven bevel gears. When the motor drives the carrier, both right and left bevel gears will nominally turn the same amount. The differential prevents damage to the robot if the user were to grab the legs and twist them in opposite directions. Both the hip and knee motors are located in the upper body with an SCLSD each to drive the hip and knee joints, Fig. 5.19d. Individual rotary potentiometers on each joint measure the relative angle (in blue in Fig. 5.19d), so any difference in angle between the right and left sides can be directly measured; the robot knows if the legs are skewed



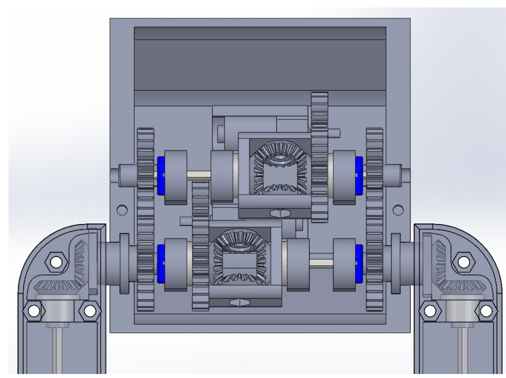
(a) Tread motor and pulley interface.



(b) Knee joint.



(c) Spring clutch limited slip differential (SCLSD).



(d) Chassis with SCLSDs and hip joints.

Figure 5.19: 3D drawings of Switchbot prototype.

and balancing is not possible. A two degree of freedom hip joint, similar to other Switchblade prototypes, allows the a driveshaft carrying the knee motor torque to pass through the hip joint independent of the rotation of the hip joint itself. Bevel gears on either end of a drive shaft in each upper leg transmits power from the hip joints (Fig. 5.19d) to the knee joints (Fig. 5.19b).

The National Instruments myRIO embedded controller was used as part of a pre-release beta program for National Instruments. The myRIO has a much smaller form factor than the earlier sbRIO (used in Switchblade) with comparable processing power and was a natural choice for this project to be able to reuse code

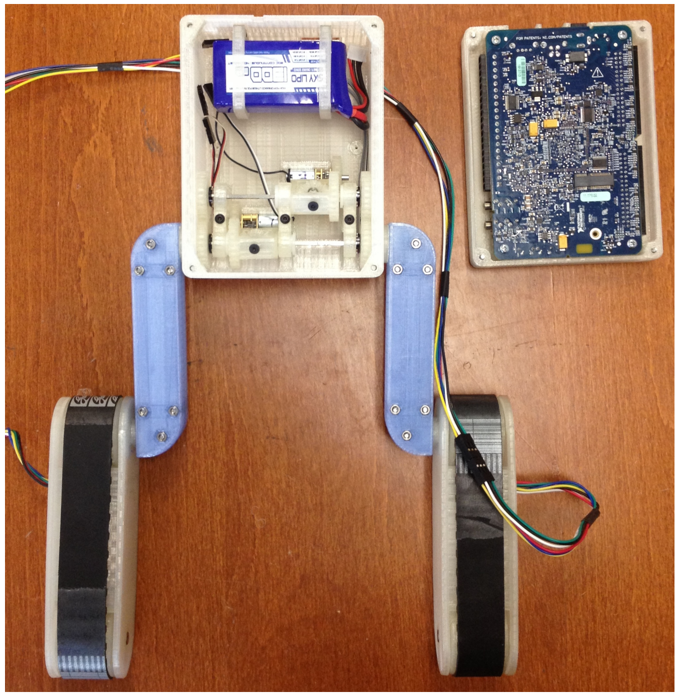


Figure 5.20: Internal view of the four motor Switchbot prototype with onboard electronics.

from the Switchblade and Switchblade nano projects. Though smaller than the sbRIO, the myRIO still proved to be the largest component in the first prototype (upper right of Fig. 5.20). A 1300 mAh 3 cell (11.1V nominal) lithium polymer battery was used to power the motors and electronics. A Bluetooth module enabled wireless communication with a computer for teleoperation. A simple custom printed circuit board (as mentioned in Sec. 5.2) serves to connect the myRIO to the motor drivers, encoders, potentiometers, and a gyroscope (the myRIO itself includes a three-axis accelerometer onboard).

While this prototype could successfully balance and be remotely driven, the weight of the myRIO prevented the robot from squatting and standing up straight with the small form factor hip and knee motors. The chassis was also unrealistically wide. It was decided to move the myRIO off board and connect it to the onboard motors and sensors over a tether. At this point, a more commercial, consumer-friendly shell was available and the components of the chassis were significantly compacted, Fig. 5.21. With the myRIO and battery off board,

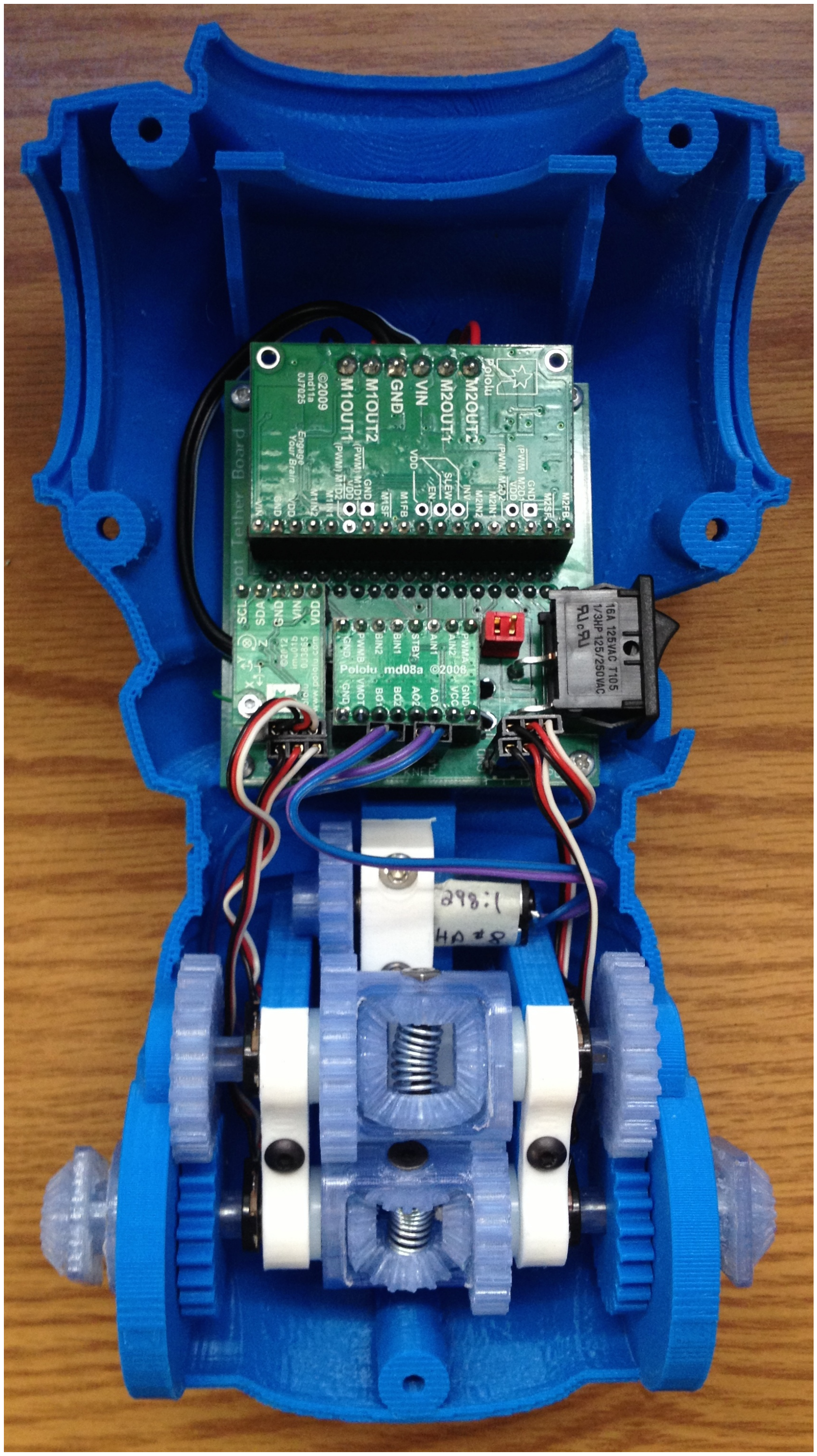


Figure 5.21: Internal view of the four motor Switchbot prototype in commercial shell.

the weight was significantly reduced, but the limitations of the drivetrain were beginning to become apparent. Given the small size of the SCLSD (in order to fit in the chassis) and the limited strength of the printed material, the amount of spring force that can be applied to increase the friction to resist relative rotation was minimal. In practice, it took little force, acting at the significant distance of the legs, to separate the left and right legs. While balancing upright, differences in friction on each side would cause separation of the legs. The complexity and added assembly cost of the SCLSD made it less attractive. Furthermore, by using separate motors for the left and right sides, the motors could be both larger in size and lower in torque (meaning that plastic gearboxes were viable instead of metal gearboxes), both of which are factors that decrease cost. The overall cost impact of switching to the six motor design (Fig. 5.22) may be negligible. Left and right hip motors in the chassis drive the hip joints directly, with potentiometers measuring the relative angle between the chassis and each upper leg. Motors were embedded in the upper legs to actuate the knee joint. An additional single stage gear reduction between the knee motor and knee joint serves two purposes. First, with a gear ratio less than one, it multiplies the torque capacity of the knee motor, which is critical for being able to transition from kneeling to standing. Second, a potentiometer can be placed coaxial to the knee joint, instead of mounting to the backside of the motor gearbox, which would increase the width of the upper leg.

The six motor design had much more robust performance, and after increasing the torque and gear ratio of the knee motors, it could lift its own weight to stand up straight, and transition from a kneeling pose to a standing (balancing) pose and vice versa, see Supplemental File 3: Switchbot video. Finally, the Switchbot prototype has been converted back to onboard control electronics using a low-cost ARM processor. The controller code has been converted from LabVIEW into c.

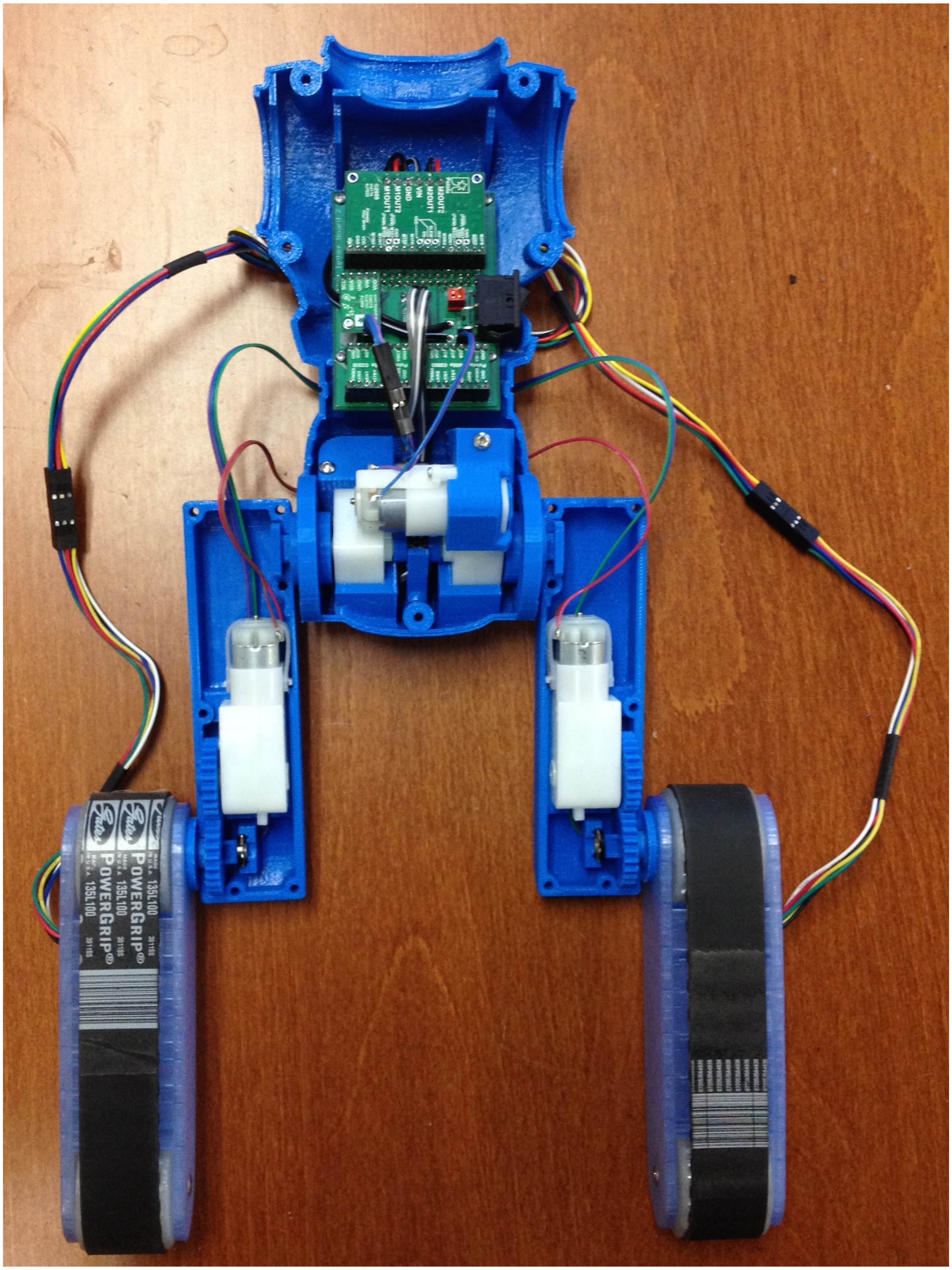


Figure 5.22: Internal view of the six motor Switchbot prototype.

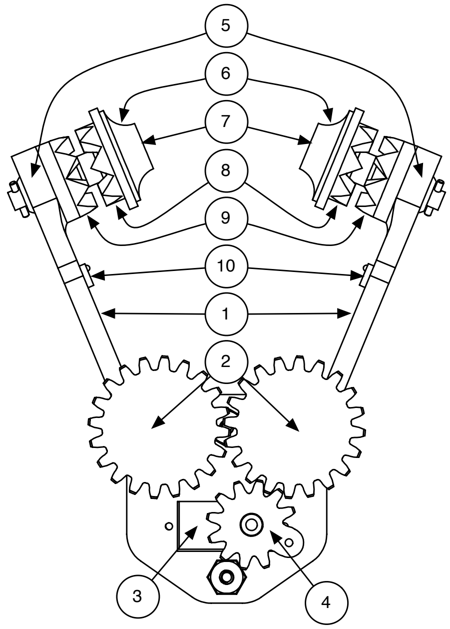


Figure 5.23: Clamp mechanism, consisting of 1) arms, 2) coupled spur gears, 3) servo motor, 4) driving spur gear, 5) swivel bearings, 6) rollers, 7) opposite pole magnets, 8) and 9) locking teeth, and 10) infrared LED and phototransistor.

5.6 SkySweeper

5.6.1 Mechanical Design

The prototype is made almost entirely of 3D-printed and off-the-shelf parts. This accelerated manufacturing and enabled more flexibility in the design than traditional machining methods.

The elbow joint pivotally connects the two links and houses an SEA with a motor and spring system connected in series. The motor is mounted on the first link such that the motor shaft is coaxial with the axis of rotation between the two links. The spring system includes two right-handed torsion springs which are mounted coaxially to the motor shaft. Opposite ends of the springs are rigidly attached to the second link. The motor shaft is rigidly coupled to an intermediate arm which engages one of the two free ends of the torsion springs, depending on the direction of rotation.

The actuated clamp mechanism is shown in detail in Fig. 5.23, it consists of two arms (1) which are coupled to rotate symmetrically with spur gears (2). A hobby-grade servo motor (3) is used to open and close the arms. A spur gear (4) connected to the output shaft of the servo meshes with the spur gear connected to one of the arms. The servo motor is geared down to increase the torque. The distal ends of the arms house swivel bearings (5), which hold the rollers (6) with a slip fit; there are four degrees of freedom: all three rotational and axial translation. Opposite pole magnets in the rollers (7) align and pull the rollers together. When the two rollers connect, they form a semicircular profile around the top half of the cable. The design is currently optimized for an $11mm$ diameter. The magnets also provide enough force to prevent the clamp from opening in the event of a power loss. When the clamp is in pivoting position, the arms are rotated to vertical and teeth on the rollers (8) engage with teeth on the arms (9), constraining relative rotation between the rollers and the arms. When the clamp is in rolling position, the arms are rotated far enough apart to disengage the teeth, but the magnets keep the rollers together. In the open position, the arms are rotated far enough to pull the magnets apart. A thin layer of silicone rubber is added to the ABS plastic roller (6) to increase friction. Square polycarbonate tubing is used to connect the clamps to the joint, wires are routed through the interior of the tube. For the constructed prototype, $m_L = 0.233kg$ and $L = 0.158m$ for a total mass of $0.466kg$ and total length of $0.632m$. Other parameters are $J_L = 0.023kgm^2$, $J_J = 0.0017kgm^2$, $k = 0.331Nm/rad$, $\sigma = 0.754Nm$, and $\zeta = 0.036Nms/rad$.

5.6.2 Electronics

An infrared LED and phototransistor pair are mounted in each clamp to detect when a cable is within grasp, see Fig. 5.23, (10). One rotary potentiometer measures the angle between the SEA shaft and the first link $(\gamma - \theta)$. The second potentiometer measures the angle between the SEA shaft and the second link, which is the same as the angle of the spring deflection $(\alpha - \gamma)$. Subtracting the two measurements from π gives the link separation angle $(\theta + \pi - \alpha)$, which is useful in the controller. The brushed DC motor in the SEA is driven with an off-the-shelf full H-bridge via a pulse width modulated signal. The finite state machine from

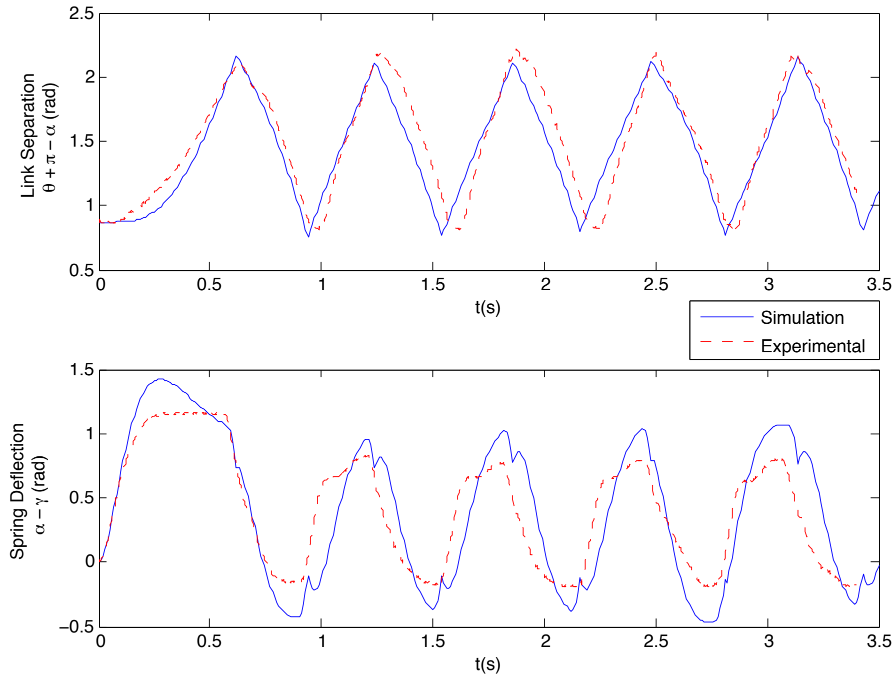


Figure 5.24: Comparison of simulation and experimental results for the inchworm maneuver.

section 3.3 is implemented on an Arduino Uno microcontroller, which measures the analog sensors and commands the actuators accordingly. We chose a 1000 mAh two cell lithium polymer battery for its low mass and high energy density.

5.6.3 Experimental Results

Data was logged from the prototype while performing the inchworm maneuver on a tensioned rope, see Sec. 1.4.4. Both the simulation and experimental results may be seen in Fig. 5.24, see also Supplemental File 4: SkySweeper video, which includes video of the inchworm, swing-up, and backflip maneuvers. The simulation results match the experimental results, although greater spring deflection is predicted in simulation. Some slipping on the rope occurs while switching clamp positions and the vibrational modes of the rope, which are excited by the movement of the robot, were not included in the simulation. These unmodeled effects contribute to the discrepancy between the plots.

5.7 Acknowledgements

Part of this chapter concerning Switchblade has appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning SkySweeper has appeared in print: N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2339-2344, 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the DC motor dynamometer has appeared in print: N. Morozovsky, R. Moroto, T. Bewley, "RAPID: an inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the perching maneuver has been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Conclusions

Several novel robotic systems have been presented in this dissertation. Though the systems are varied, they share a unified Lagrangian dynamics model and a design methodology that utilizes modern additive manufacturing and embedded electronics. The systems can be used in a wide variety of applications including border patrol, security, search & rescue, maintenance, construction, and entertainment. Furthermore, the processes followed in developing dynamical models, designing control systems, processing sensor data, and building physical prototypes is broadly applicable to the field of robotics. It is the hope of the author that the methods and tools presented in this dissertation will serve to enable the reader in his or her own robotic system development.

6.1 Robotic Systems

6.1.1 DC Motor Dynamometer

An inexpensive, open source dynamometer has been presented that can accurately and precisely fit parameters to a given electromechanical model of a motor. The motor model developed also proves to be useful in the dynamic formulation of mobile robotic systems. The integrated sensor suite measures several physical values in the system. The reconfigurable mechanical design allows for testing a wide range of motors.

With the motor dynamics identified, a model-based control algorithm may be developed for a robotic system. Future work may include *in situ* testing in a robotic system, using similar algorithms to those developed here, to characterize and track the motor parameters for motor health monitoring as well as estimating external friction parameters, such as rolling resistance over different terrain. Online parameter identification may even be incorporated into the controller [64].

6.1.2 Switchblade

Switchblade is a robotic platform that combines the treads of a tank with the balancing behavior of an inverted pendulum to reach a new level of agility. The robot can overcome obstacles on the order of its length instead of the order of its height. A two degree of freedom hip joint enables the current prototype to perform complicated maneuvers with a relatively simple internal structure and wiring. A working prototype has been created that can "stand up" on its own and maintain its balance. This platform has potential for applications in search & rescue, mine exploration, homeland security, border patrol, reconnaissance, and ordnance disposal.

We have presented a novel approach to the problem of stair climbing utilizing feedback control on this platform. Successful experimental results have been shown for a prototype traversing across a narrow beam. This method of stair climbing enables a relatively small robot (length scale on the same order as the height of a single step) to climb stairs without any external or expensive sensors and without any dedicated stair climbing hardware. As mentioned in Sec. 5.5.3, the mass distribution of the current robot is not optimized for climbing standard-sized stairs. Better performance could be achieved by decreasing the mass and shifting the center of mass of the chassis further from the tread assembly pivot point (alternatively, the tread assembly pivot point could move closer to the center of the tread assemblies). Stiction and friction could be significantly reduced by switching from discrete tread links to a timing belt and changing the design of the tread assemblies to avoid rubbing. All of these changes would act to increase the control authority of the system. In future work, the robot could perform automated parameter identification of a staircase (i.e. rise and run) from sensor data, such as from a camera or LIDAR [65] [66] [67]. The robot could then calculate a trajectory and controller to climb the staircase using the algorithms outlined in this paper. One or more robots could be used to autonomously map a multi-story building, such as in a fire-fighting or urban warfare scenario. The perching controller could be expanded to balance on uneven geometry, adaptively estimating the point of contact, possibly with additional sensors [68].

The Switchblade nano and Switchbot prototypes have both shown significant promise in delivering the maneuverability of the full-scale design in a smaller, and lower cost, form factor.

6.1.3 SkySweeper

SkySweeper is a unique cable-locomoting robot with few actuators, but many configurations, which leads to multiple modes of locomotion. The dynamics were derived for the different clamp configurations and finite state machine controllers were developed to perform different maneuvers. Both the dynamics and controller were integrated into a simulation to validate the concept. Finally, a prototype was constructed and shown to behave as predicted in simulation. Compared to existing systems, SkySweeper is smaller, less complex, cheaper, and earlier in the development process. SkySweeper is designed to locomote quickly, and as such may be better suited to applications other than inspection, such as entertainment.

In future work, the simulation environment can be used to optimize the spring stiffness, link length, and different maneuvers' control input sequences to minimize the cost of transport, defined as the power required to transport mass over distance. The dynamical model can also be expanded to include the dynamics of the rope including curvature, such as in a suspension bridge. The clamp design may be improved to better handle high dynamic loads and could be modified to enable climbing vertical pipes or rope. Cameras and other sensors for inspection applications can readily be integrated into the robot. Specific to the power line environment, energy could be harvested from the surrounding electric field, which would enable long duration deployments.

6.2 Development Tools

6.2.1 Dynamics

A systematic process for applying Lagrangian dynamics to robotic systems has been presented. This process is broadly applicable to many classes of robots, as illustrated by its application to systems as different as Switchblade and SkySweeper. Two methods have been presented to impose physical constraints on the dynamic system. Common constraints, such as backlash, stiction, and no-slip between a wheel and the ground, have been illustrated. The strength of this method is that it can be programmatically applied to robotic systems, drastically reducing the time necessary to develop dynamic equations of motion for a new system for simulation and control system design.

6.2.2 Controls

Linearization and discretization of nonlinear systems for implementation on digital control electronics has been shown to be a generally useful technique. Feed-forward controls are also used to stabilize unstable equilibrium manifolds. Integral control increases robustness to parameter and sensor error. This control design can also be achieved in a programmatic fashion for rapid prototyping. For systems with dynamics that change significantly, lookup tables for gain scheduling can be employed where subsequent entries are linearly interpolated for a smooth control law. Lookup tables can also be programmatically calculated. A method of friction compensation has been presented and validated that decreases the nonlinear effect of stiction and coulomb friction on a system.

6.2.3 Estimation

The presentation of the complementary filter (Sec. 4.1), as applied to estimating body angle with respect to gravity, while not new, is compact and broadly useful in mobile robotic systems. An advanced technique to estimate velocity from quadrature encoder signals has been presented. The method is accurate over a wide velocity range, particularly at low velocity, where other common estimation techniques fail, either with high discretization noise or phase delay. The method also smoothes process noise at higher speeds without adding phase delay. The definition of the $\mathcal{M}$ number is useful to define different velocity regimes, normalizing for encoder resolution and sampling time. The method of parameter identification developed for the DC motor dynamometer is useful not only for motor parameters, but may also be used to identify friction, rotational inertia, or other parameters of robotic systems that may be difficult to characterize otherwise.

6.2.4 Prototypes

Over the development of literally dozens of prototypes, several best practices have become apparent. The recent advent of desktop additive manufacturing (3D printing) both increases the potential complexity of rapid prototyped parts and significantly decreases the time of the design-build-test iteration cycle. The explosion of low-cost, capable embedded electronics dovetails with the proliferation of 3D printers, easing the creation of mobile robotic systems. Custom printed circuit boards can simplify wiring, ease debugging, and increase robustness.

6.3 Acknowledgements

Part of this chapter concerning Switchblade has appeared in print: N. Morozovsky, C. Schmidt-Wetekam, T. Bewley, "Switchblade: an agile treaded rover," *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2741-2746, 2011. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning SkySweeper has appeared in print: N. Morozovsky, T. Bewley, "SkySweeper: A Low DOF, Dynamic High Wire Robot" *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pp. 2339-2344, 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the DC motor dynamometer has appeared in print: N. Morozovsky, R. Moroto, T. Bewley, "RAPID: an inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1855-1860. Dec. 2013. The dissertation author was the primary investigator and author of this paper.

Part of this chapter concerning the perching maneuver has been submitted for publication. N. Morozovsky, T. Bewley. The dissertation author was the primary investigator and author of this paper.

Appendix A

Nick's Rules of Robotics

1. Never disassemble a working robot.
2. If it works the first time, you're testing it wrong.
3. When in doubt, lubricate.
4. Never underestimate the estimation problem.
5. If specs for a part are listed differently in two places, they're both wrong.
6. Glue, tape, and zip-ties are not engineering solutions (though they might work in a pinch).
7. Do not leave lithium polymer batteries charging unattended.
8. Always have a complete CAD model, including screws and fasteners, before constructing your robot.
9. Avoid using slip rings if at all possible.
10. Clamping collars are always better than set screws. If you have to use set screws (e.g. for cost reasons), use a driving flat and an appropriate thread-locking agent.
11. Always check polarity before plugging a component into a power source.

Appendix B

Lagrangian Dynamics Code

The code in this appendix will generate the equations of motion for a mobile inverted pendulum (e.g. myMIP or eyeFling) and generate the linear quadratic regulator (LQR) state feedback gains for the linearized and discretized system with and without state augmentation with integrated regulator error. The generalized coordinates and parameters are defined in Fig. 2.2. The code can directly be translated to other robotic systems by adding coordinates, etc. This code is implemented in MATLAB and makes use of the Symbolic Math Toolbox, although similar code may be written for Mathematica or Python (with the SymPy library). The code below may also be found in Supplemental File 6: eyeFlingEOM.m

```
% eyeFlingEOM.m
% derive equations of motion for a mobile inverted pendulum
% all coordinates (gamma, x, phi, theta) absolute CCW from vertical

% define symbolic variables as real numbers
% generalized coordinates and their time derivatives, and input u
syms gamma x phi theta gammad xd phid thetad gammadd xdd phidd ...
    thetadd u real
% geometrical coordinates
syms xw yw xc yc real
% system parameters, lengths, masses, inertias, motor torque, etc.
syms Lc r mw mc g Jw Jc Jm sw bw real
```

```
% define vectors of generalized coordinates
q = [gamma; x; phi; theta];              % positions
qd = [gammad; xd; phid; thetad];         % velocities
qdd = [gammadd; xdd; phidd; thetadd];    % accelerations

% define positions of centers of mass of rigid bodies in x and y
% coordinates in terms of generalized coordinates

% wheel COM location (same as center of rotation)
xw = x;
yw = r;

% chassis COM location
xc = xw - Lc*sin(theta);
yc = yw + Lc*cos(theta);

% take jacobian of x and y positions with respect to generalized
% coordinates and multiply by vector of velocities to find
% velocities of centers of mass of rigid bodies

% velocity of wheel COM
xwd = jacobian(xw,q) * qd;
ywd = jacobian(yw,q) * qd;

% velocity of chassis COM
xcd = jacobian(xc,q) * qd;
ycd = jacobian(yc,q) * qd;

% kinetic energies, using velocities of centers of mass defined
% above. 0.5*[mass*(COM velocity)^2 +
% rotational inertia*(rotational velocity)^2] for rigid bodies
% 0.5*motor inertia*(relative velocity)^2 for motor
Tw = 0.5*(mw*(xwd^2 + ywd^2) + Jw*phid^2);
Tc = 0.5*(mc*(xcd^2 + ycd^2) + Jc*thetad^2);
Tm = 0.5*(Jm*(gammad-thetad)^2);

% gravitational potential energy, use y coordinates of centers
```

```
% of mass of rigid bodies
V = mw*g*yw + mc*g*yc;

% Lagrangian
L = Tw + Tc + Tm - V;

% set up Euler-Lagrange equations
one = simplify(collect(jacobian(L,qd))); % delta L / delta qd
two = simplify(collect(jacobian(one,[q' qd'])));
three = two*[qd; qdd]; % d/dt (delta L / delta qd)
four = transpose(simplify(collect(jacobian(L,q)))); % delta L / ...
    delta q
five = three - four;

% get mass matrix, coefficients of acceleration terms
M = simplify(jacobian(five,qdd));

% everything else is F
F = simplify(five - M*qdd);

% generalized forces/torques
B = [ 2;
      0;
      0;
     -2];
% u: + torque exerted on gamma, - torque exerted on theta
nu = length(u);

% motor back EMF damping
Z = bw*(gammad-thetad);

% define constraints in form Ac(q)*qd = 0

% no-slip constraint, xd + r*phid = 0
Ans = [0 1 r 0];

% no backlash constraint, gammad + phid = 0
Awa = [1 0 -1 0];
```

```

% concatenate constraints
Ac = [Awa; Ans];

% find orthonormal basis of null space of constraint matrix
S = null(Ac);
z = sqrt(sum(S.^2));
S = [S(:,1)./z(1) S(:,2)./z(2)]; % columns form orthonormal ...
    null space basis

% SB doesn't include superfluous coordinates gamma or x
SB = S(end-1:end,:);

% substitute superfluous variables
xexp  = -r*phi;
xdexp = -r*phid;
Fc = subs(F,[x; xd; gamma; gammad],[xexp; xdexp; phi; phid]);
Zc = subs(Z,[x; xd; gamma; gammad],[xexp; xdexp; phi; phid]);

% reduced coordinate vectors
qr = [phi; theta];
qrd = [phid; thetad];
n = length(qr);

% net motor torque
tau = sw*u-Zc;

% determine equilibrium manifold
eq1 = S'*(-Fc + B*tau);

% impose zero velocities
eq2 = subs(eq1,qd,zeros(2*n,1));

% solve for expression for u^*
f1qs = eq2(1);
f1j = jacobian(f1qs,u);
ustar = simplify(-(f1qs - u*f1j)/f1j);

```

```
% define intermediate variable to solve for sin(theta^*)
syms sintheta real
f2qs = eq2(2);
f2qss = subs(f2qs,sin(theta),sintheta);
f2j2 = jacobian(f2qss,sintheta);
sinthetastar = simplify(collect(-(f2qss - sintheta*f2j2)/f2j2));
sinthetastar = simplify(subs(sinthetastar,u,ustar));

% for a mobile inverted pendulum on level ground, both u^* and
% theta^* are equal to zero

%% define system parameters

mc = 1.150; % kg of chassis
mw = 2*0.064; % kg of both wheels
r = 0.141/2; % m
Lc = r; % m
Jw = 0.5*mw*r^2; %kg*m^2
Jc = mc*Lc^2; %kg*m^2
V = 11.1; % V, nominal 3 cell LiPo

% wheel motor parameters from DC motor dynamometer, Pololu 37D, ...
    18.75:1
rw = mean([3.4903 3.3826]);
kw = mean([0.1409 0.1455]);
sw = kw*V/rw;
bw = (kw^2)/rw;
Jm = 4.5894e-04; % kg*m^2

g = 9.81; % m/s^2
ts = 0.01; % sec, = 100 Hz

%% linearize to get controller

% substitute parameter values into expressions
Ss = subs(S);
SBs = subs(SB);
Fcs = subs(Fc);
```

```
Zs = subs(Zc);
StMSs = subs(S'*M*S);

% complete nonlinear dynamics
Anl = SBs*(StMSs\(-Ss'*(Fcs+B*Zs)));
Bnl = SBs*(StMSs\(Ss'*B*sw*u));

% take jacobian of nonlinear dynamics with state vector and ...
    input vector
Al = jacobian(Anl, [qr; qrd]);
Bl = jacobian(Bnl, u);

% substitute values into jacobian, can be a slow process for
% larger systems; can save execution time by writing symbolic
% expression for jacobian to a file and calling it as a
% function (programmatically)
Als = subs(Al, [qr; qrd], zeros(2*n,1));
Bls = subs(Bl, [qr; qrd], zeros(2*n,1));

% construct state space system matrices
A = [zeros(n) eye(n); double(Als)];
Bc = [zeros(n,nu); double(Bls)];

% set up integral control, xi dot = phi = C * x
C  = [1 0 0 0];
ni = size(C,1);
Ai = [A zeros(2*n,ni); C zeros(ni)];
Bi = [Bc; zeros(ni,nu)];

% define Q and R weighting matrices
Qi = 1/(pi)^2;
Q  = [1/(2*pi)^2 1/(0.125)^2 1/(6*pi)^2 1/(3.5)^2];
R  = 1/(0.5)^2;

% create diagonal matrices
Qi = diag([Q Qi]);
Q  = diag(Q);

```

```
% solve discrete algebraic riccati equation (DARE) with
% Matlab's lqrd command, which automatically converts
% continuous [A, B] to discrete time with sample time ts

[K,~,~] = lqrd(A,Bc,Q,R,ts);
K = -K    % flip sign to fit convention u = K*x, output to screen

% with integral control
[Ki,~,~] = lqrd(Ai,Bi,Qi,R,ts);
Ki = -Ki  % flip sign to fit convention u = K*x, output to screen
```

This code can be expanded to programmatically calculate all the entries of a lookup table for gain scheduling for a system with dynamics that change significantly along a trajectory of interest. The equilibrium manifold can be defined as a set of vectors of equilibrium positions and then a discrete time state feedback matrix can be found for each vector (by linearizing, etc. about that vector) with the use of a for loop. The lookup table of state feedback matrices can be stored in a three dimensional array.

Appendix C

MATLAB Plotting Code

The code in this appendix creates a sequence of PNG images animating a mobile inverted pendulum (e.g. myMIP or eyeFling) from a MATLAB/Simulink simulation. The sequence of PNG images can be converted to a movie file with Apple QuickTime Player 7 (not the most recent version). The code below may also be found in Supplemental File 7: eyeFlingPlot.m

```
% eyeFlingPlot.m
% animate eyeFling simulation

% get data from simulation
t = scope_state.time;
phiP = scope_state.signals.values(:,1);
thetaP = scope_state.signals.values(:,2);
uP = scope_u.signals.values(:,1);

% load data file with parameter values
load eyeFling
msP = P(1);
mcP = P(2);
LcP = P(3);
RadP = P(9);

% no-slip condition
```

```
xP = -RadP*phiP;

MAKE_IMG_SEQ = 1;
prefix = 'eyeFlingPlot'; % filename
path = ['animations/'];  % directory to store images
count = 0;

% circle for wheel
plot_sprocket = RadP*[cos([0:.1:2*pi]') sin([0:.1:2*pi]')];

% line for chassis
plot_chassis = [0 0; 0 LcP];

% sprocket COM location (same as joint location)
xs = xP;
ys = RadP;

% chassis COM location
xc = xs - LcP*sin(thetaP);
yc = ys + LcP*cos(thetaP);

% Total COM location
xcm = (msP*xs + mcP*xc)/(msP+mcP);
ycm = (msP*ys + mcP*yc)/(msP+mcP);

% 25 frames per second with data at 100 Hz
fpl=4;

% bound figure to fit min and max position locations
dim = [min(xs)-LcP max(xs)+LcP min(ys)-LcP max(ys)+LcP];

fig = figure('Position',[0 0 720 480]);
for i = 1:fpl:length(t)

    % rotate chassis by thetaP, then offset position by xs and ys
    chassis = plot_chassis*[cos(thetaP(i)) sin(thetaP(i)); ...
        -sin(thetaP(i)) cos(thetaP(i))];
    chassis(:,1) = chassis(:,1) + xs(i);
```

```
    chassis(:,2) = chassis(:,2) + ys;

    % rotate sprocket by phiP, then offset position by xs and ys
    sprocket = plot_sprocket*[cos(phiP(i)) sin(phiP(i)); ...
        -sin(phiP(i)) cos(phiP(i))];
    sprocket1(:,1) = sprocket(:,1) + xs(i);
    sprocket1(:,2) = sprocket(:,2) + ys;

    %Plot that stuff
    plot(chassis(:,1),chassis(:,2),'g','LineWidth',2)
    hold on; % don't overwrite current figure
    plot(sprocket1(:,1),sprocket1(:,2),'b')
    plot(sprocket1(1,1),sprocket1(1,2),'r.') % marker to ...
        observe sprocket rotation
    plot(xj(i),yj(i),'go','MarkerSize',10*abs(uP(i))+1) % plot ...
        control effort

    % plot ground
    plot([0 dim(2)],[0 0],'k')
    plot([0 dim(1)], [0 0],'k')

    % plot centers of mass
    plot(xc(i),yc(i),'go','MarkerSize',8*sqrt(mcP))
    plot(xcm(i),ycm(i),'ko','MarkerSize',8*sqrt(mcP+msP))

    hold off;
    axis 'equal'
    axis(dim)

    % title with current time
    title(['t: ',num2str(round(100*t(i))/100),' sec.']);
    drawnow
    pause(0.1);

    if MAKE_IMG_SEQ
        %save png with incremental filename
        saveas(fig,strcat(path,prefix,num2str(count),'.png'));
        count = count + 1;
```

```
    end

    if i == 1
        pause(0.5)
    end

end
```

This code can be expanded to plot more complex systems or geometric shapes (e.g. Fig. 3.4, the stair climbing animation in Supplemental File 1: Switchblade video, and the spring force meter in the simulation animations in Supplemental File 4: SkySweeper video).

Bibliography

[1] N. Morozovsky, C. Schmidt-Wetekam, and T. Bewley, "Switchblade: An agile treaded rover," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 2741–2746.

[2] C. Schmidt-Wetekam, D. Zhang, R. Hughes, and T. Bewley, "Design, optimization, and control of a new class of reconfigurable hopping rovers," *Decision and Control, 2007 46th IEEE Conference on*, pp. 5150–5155, 2007.

[3] Y. Khaligh and M. Namvar, "Adaptive control of robot manipulators including actuator dynamics and without joint torque measurement," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4639–4644.

[4] M. R. Hossain and N. Krouglicof, "Propeller dynamometer for small Unmanned Aerial Vehicle," in *Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference on*, 2010, pp. 1–5.

[5] Y. Tarte, Y. Chen, W. Ren, and K. Moore, "Fractional Horsepower Dynamometer - A General Purpose Hardware-In-The-Loop Real-Time Simulation Platform for Nonlinear Control Research and Education," in *Decision and Control, 2006 45th IEEE Conference on*, 2006, pp. 3912–3917.

[6] P. Wolm, X. Chen, J. Chase, W. Pettigrew, and C. Hann, "Analysis of a PM DC Motor Model for Application in Feedback Design for Electric Powered Mobility Vehicles," in *Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference on*, 2008, pp. 640–645.

[7] E. Steltz and R. S. Fearing, "Dynamometer Power Output Measurements of Miniature Piezoelectric Actuators," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 1, pp. 1–10, 2009.

[8] G. Totis and M. Sortino, "Development of a modular dynamometer for triaxial cutting force measurement in turning," *International Journal of Machine Tools and Manufacture*, vol. 51, no. 1, pp. 34–42, 2011.

[9] Segway Inc. [Online]. Available: http://www.segway.com

[10] F. Grasser, A. D Arrigo, S. Colombi, and A. Rufer, "JOE: a mobile, inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107–114, 2002.

[11] K. Pathak, G. Student, J. Franch, and S. Agrawal, "Velocity control of a wheeled inverted pendulum by partial feedback . . . ," *Atlantis*, 2004.

[12] T. Lauwers and G. Kantor, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," *Robotics and Automation, Proceedings IEEE International Conference on*, pp. 2884–2889, 2006.

[13] M. Kumaga and T. Ochiai, "Development of a robot balanced on a ball — Application of passive motion to transport —," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 4106–4111, 2009.

[14] T. Sugihara and Y. Nakamura, "Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control," *. . . Conference on Robotics . . .* , 2002.

[15] J. Chestnutt, J. Kuffner, and K. Nishiwaki, "Planning biped navigation strategies in complex environments," *IEEE Int. Conf. Hum. Rob.*, 2003.

[16] B. Yamauchi, "PackBot: A versatile platform for military robotics," *Proceedings of SPIE*, vol. 5422, pp. 228—237, 2004.

[17] F. Michaud, D. Letourneau, M. Arsenault, Y. Bergeron, R. Cadrin, F. Gagnon, M. Legault, M. Millette, J. Pare, M. Tremblay, P. Lapage, Y. Morin, J. Bisson, and S. Caron, "AZIMUT, a leg-track-wheel robot," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2553–2558 vol.3, 2003.

[18] J. Borenstein and M. Hansen, "The OmniTread OT-4 serpentine robot-design and performance," *Journal of Field Robotics*, 2007.

[19] Vecna Technologies Inc. [Online]. Available: http://vecnarobotics.com/solutions/bear/

[20] DARPA Robotics Challenge. [Online]. Available: http://www.theroboticschallenge.org

[21] SCHAFT Inc. [Online]. Available: http://schaft-inc.jp

[22] Boston Dynamics. Atlas - The Agile Anthropomorphic Robot. [Online]. Available: http://www.bostondynamics.com/robot_Atlas.html

[23] C. X. Chen, M. M. Trivedi, and C. R. Bidlack, "Simulation and animation of sensor-driven robots," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 5, pp. 684–704, 1994.

[24] N. Li, S. Ma, B. Li, M. Wang, and Y. Wang, “An online stair-climbing control method for a transformable tracked robot,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 923–929.

[25] S. Yu, T. Wang, Y. Wang, D. Zhi, C. Yao, X. Li, Z. Wang, Y. Luo, and Z. Wang, “A tip-over and slippage stability criterion for stair-climbing of a wheelchair robot with Variable Geometry Single Tracked Mechanism,” in *Information and Automation (ICIA), 2012 International Conference on*, 2012, pp. 88–93.

[26] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, “Reliable stair climbing in the simple hexapod 'RHex',” in *2002 IEEE International Conference on Robotics and Automation*. IEEE, 2002, pp. 2222–2227.

[27] DEKA Research and Development - Technologies and Applications - iBOT. [Online]. Available: http://www.dekaresearch.com/ibot.shtml

[28] S. D. Herbert, A. Drenner, and N. Papanikolopoulos, “Loper: A quadruped-hybrid stair climbing robot,” in *2008 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2008, pp. 799–804.

[29] S.-C. Chen, K. J. Huang, W.-H. Chen, S.-Y. Shen, C.-H. Li, and P.-C. Lin, “Quattroped: A Leg–Wheel Transformable Robot,” *Mechatronics, IEEE/ASME Transactions on*, vol. 19, no. 2, pp. 730–742, 2014.

[30] J.-s. Koh, D.-y. Lee, S.-w. Kim, and K.-j. Cho, “Deformable soft wheel robot using hybrid actuation,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3869–3870.

[31] S. A. Stoeter and N. Papanikolopoulos, “Autonomous Stair-Climbing With Miniature Jumping Robots,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 2, pp. 313–325, Apr. 2005.

[32] T. Tsuda, H. Mochiyama, and H. Fujimoto, “Quick stair-climbing using snap-through buckling of closed elastica,” *Micro-NanoMechatronics and Human Science (MHS), 2012 International Symposium on*, pp. 368–373, 2012.

[33] H. Wang, L. He, Q. Li, W. Zhang, D. Zhang, and P. Xu, “Research on a kind of leg-wheel stair-climbing wheelchair,” in *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, 2014, pp. 2101–2105.

[34] H. Hong, D. Kim, J. Kim, J. Oh, and H. S. Kim, “A locomotive strategy for a stair-climbing mobile platform based on a new contact angle estimation,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 3819–3824.

[35] B. Seo, H. Kim, M. Kim, Y. Liu, K. Jeong, and T. Seo, "Novel robotic platform for stable stair climbing by flipping locomotion," in *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, 2013, pp. 109–111.

[36] T. Takaki, T. Aoyama, and I. Ishii, "Development of inverted pendulum robot capable of climbing stairs using planetary wheel mechanism," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 5618–5624.

[37] R. C. Luo, M. Hsiao, and T.-W. Lin, "Erect wheel-legged stair climbing robot for indoor service applications," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 2731–2736.

[38] C. Schmidt-Wetekam. !Lean robot climbs stairs. [Online]. Available: http://www.youtube.com/watch?v=rUbMQzLjo2Y

[39] K. Toussaint, N. Pouliot, and S. Montambault, "Transmission line maintenance robots capable of crossing obstacles: State-of-the-art review and challenges ahead," *Journal of Field Robotics*, vol. 26, no. 5, pp. 477–499, May 2009.

[40] P. Debenest and M. Guarnieri, "Expliner — From prototype towards a practical robot for inspection of high-voltage lines," in *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*, 2010, pp. 1–6.

[41] N. Pouliot and S. Montambault, "Field-oriented developments for LineScout Technology and its deployment on large water crossing transmission lines," *Journal of Field Robotics*, vol. 29, no. 1, pp. 25–46, Dec. 2011.

[42] Z. Tingyu, Z. Guoxian, and J. Jian, "Development of a dual-arm mobile robot for high voltage power lines," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, 2007, pp. 1924–1929.

[43] M. Bühringer, J. Berchtold, M. Büchel, C. Dold, M. Bütikofer, M. Feuerstein, W. Fischer, C. Bermes, and R. Siegwart, "Inspection Of High Voltage Power Lines–A New Approach," *12th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Sep. 2009.

[44] C. Nie, X. Pacheco Corcho, and M. Spenko, "Robots on the Move: Versatility and Complexity in Mobile Robot Locomotion," *Robotics & Automation Magazine, IEEE*, vol. 20, no. 4, pp. 72–82, 2013.

[45] G. Pratt and M. Williamson, "Series elastic actuators," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, 1995, pp. 399–406.

[46] S. Saab and R. Kaed-Bey, "Parameter identification of a DC motor: an experimental approach," *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on*, vol. 2, pp. 981–984 vol.2, 2001.

[47] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[48] A. R. Cavender, "A note on dynamics," University of California, San Diego, Tech. Rep., December 2008.

[49] D. A. Wells, "A "Power Function" for the Determination of Lagrangian Generalized Forces," *Journal of Applied Physics*, vol. 16, no. 9, pp. 535–538, 1945.

[50] N. Morozovsky, R. Moroto, and T. Bewley, "RAPID: An Inexpensive Open Source Dynamometer for Robotics Applications," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1855–1860, 2013.

[51] F. E. Udwadia, "On constrained motion," *Applied Mathematics and Computation*, vol. 164, no. 2, pp. 313–320, May 2005.

[52] M. Nordin and P.-O. Gutman, "Controlling mechanical systems with backlash—a survey," *Automatica*, vol. 38, no. 10, pp. 1633–1649, Oct. 2002.

[53] J. Arthur E. Bryson and Y. C. Ho, *Applied Optimal Control*. New York, NY: Taylor & Francis Group, 1975.

[54] B. Bona and M. Indri, "Friction Compensation in Robotics: an Overview," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, 2005, pp. 4360–4367.

[55] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1996.

[56] W. T. Higgins, "A Comparison of Complementary and Kalman Filtering," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 3, pp. 321–325, 1975.

[57] R. C. Kavanagh, "Improved digital tachometer with reduced sensitivity to sensor nonideality," *Industrial Electronics, IEEE Transactions on*, vol. 47, no. 4, pp. 890–897, 2000.

[58] Angular position measurement with a quadrature encoder. [Online]. Available: https://decibel.ni.com/content/docs/DOC-17346

[59] User Guide and Specifications NI myDAQ. [Online]. Available: http://www.ni.com/pdf/manuals/373060e.pdf

[60] Pulse Width Modulation (PWM) Using NI-DAQmx and LabVIEW. [Online]. Available: http://www.ni.com/white-paper/2991/en

[61] Maxon motor ag. motor specifications, article number 110117. [Online]. Available: http://maxonmotor.com/medias/sys_master/8800973651998/12_097_EN.pdf

[62] Maxon motor ag. gearbox specifications, article number 110483. [Online]. Available: http://maxonmotor.com/medias/sys_master/8801015988254/12_230_EN.pdf

[63] J. Stephan, M. Bodson, and J. Chiasson, "Real-time estimation of the parameters and fluxes of induction motors," *Industry Applications, IEEE Transactions on*, vol. 30, no. 3, pp. 746–759, 1994.

[64] M. F. Moussa, M. Saad, and Y. G. Dessouky, "Adaptive control and one-line identification of sensorless permanent magnet DC motor," in *Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), 2010 IEEE Region 8 International Conference on*, 2010, pp. 852–857.

[65] R. C. Luo, M. Hsiao, and C.-W. Liu, "Multisensor integrated stair recognition and parameters measurement system for dynamic stair climbing robots," in *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, 2013, pp. 318–323.

[66] J. Chocoteco, R. Morales, V. Feliu, and L. Sanchez, "Trajectory Planning for a Stair-Climbing Mobility System Using Laser Distance Sensors," *Systems Journal, IEEE*, no. 99, pp. 1–13, 2014.

[67] C.-K. Tseng, I.-H. Li, Y.-H. Chien, M.-C. Chen, and W.-Y. Wang, "Autonomous stair detection and climbing systems for a tracked robot," in *System Science and Engineering (ICSSE), 2013 International Conference on*, 2013, pp. 201–204.

[68] C. Schmidt-Wetekam, N. Morozovsky, and T. Bewley, "Inertial rotation center position estimation for a perching treaded vehicle," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 4097–4102.