Energy and Mobility Management of a Ground Robot to Increase Operational Capacity

by

John A. Broderick

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2015

Doctoral Committee:

Professor Dawn M. Tilbury, Co-Chair
Associate Professor Ella M. Atkins, Co-Chair
Assistant Professor Necmiye Ozay
Professor Huei Peng

Dedication

For Breanna, thanks for your love and support

Acknowledgments

I would like to thank the Automotive Research Center at the University of Michigan and US Army TARDEC for financial support for during my doctoral work. Completion of this work would not have been possible without this support.

I am grateful to Dawn Tilbury and Ella Atkins for their leadership and guidance during my doctoral work. I depended on their knowledge and expertise to guide me and push me to be a better student.

Additional thanks go to the Quad research members: Matt Castanier, Rob Karlsen and Herb Dobbs for their perspective to guide us in our choice of research topics and Doug MacKenzie and Jon Rice for their industry perspective on the challenges of robotics and power systems. I would like to thank Jack Hartner and Dan Maslach for their help in setting up the fuel cell power system experiments. Additional thanks go to John Dannenhoffer, Rob Isenberg, and Anshul Shambav for their help in conducting slipping robot experiments. Many thanks to the many other students in Dawn and Ella's research labs; their comments and critiques refined my presentation skills and provided insights into my research.

Most importantly, I am grateful for my family and their support: my parents for providing a great example of learning and teaching, Cora and Arthur for their smiles at the end of the day, and Breanna for her strength, love, and perseverance through the last four and a half years.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Energy and Mobility Management of a Ground Robot to Increase Operational Capacity

by

John A. Broderick

Co-Chairs: Dawn M. Tilbury, Ella M. Atkins

The operational capacity of an unmanned ground vehicle (UGV) is limited by many different challenges. Limited energy storage and loss of wheel traction can interrupt completion of the robot's mission. To prevent these disruptions to robot operation, three different problems are considered: trajectory planning for an area coverage mission with energy considerations, efficient power management for a hybrid power system, and traction control to prevent wheel slip.

An area coverage mission consists of driving the robot within sensor range of every point in a region. To plan operation, the energy usage characteristics of the robot are required. A method to determine motor efficiency and energy usage is presented. To balance the mission goals and the energy required for operation, a novel cost function, weighting motor torques, area covered and motor efficiency, is used to plan the trajectory. This trajectory is constrained to follow a coverage path planned using existing techniques. We show how the cost function can be used to tradeoff between energy usage and time required to complete the mission.

To increase onboard energy storage, we propose a hybrid power system for a UGV. This combination of power sources requires additional control algorithms to determine which sources should be used throughout the mission. Our control algorithm is based on forming a model of the hybrid power system with power demands from a particular mission. To make the control optimization problem tractable, the model is simplified by using averaged dynamics. Using this model, power management is optimized to limit energy losses. Simulation and Experimental results with a battery/fuel cell power system are presented and show 5% decrease in energy usage compared to a baseline control strategy.

Both lateral and longitudinal wheel slip must be controlled to prevent loss of traction. Using a slipping UGV model, based on automotive friction models, we develop two control algorithms: a novel switching controller and a sliding mode controller. The switching controller considers both lateral and longitudinal wheel slip and completes turning maneuvers. The sliding mode controller, based on automotive techniques, only controls longitudinal wheel slip, possibly losing traction while turning.

CHAPTER 1

Introduction

Unmanned ground vehicles (UGVs) can operate in situations that are hazardous to humans. Operating either autonomously or under remote human control, ground robots can effectively accomplish the desired mission while limiting potential dangers to humans. In the 2011 nuclear disaster at the Fukushima power plant, UGVs were used to assess the internal condition of the power plant, allowing the human operators to stay in a safer area [1]. In military settings, ground robots can greatly decrease the soldier's workload by performing basic functions such as transporting equipment and extending threat detection range. By partnering with robots, the soldiers are able to remain alert and focused on their mission and possible threats.

Most existing robotic systems that operate in unknown environments are remotely operated by humans. Teleoperation allows the robot to accomplish their mission using the human operators superior problem solving and reasoning capabilities to adapt to unanticipated challenges. Many current research efforts are directed at developing systems that can operate more autonomously in unstructured environments, such as self-driving cars [2, 3]. As these systems become more sophisticated and intelligent, the number of possible applications will increase.

In addition to increasing capabilities, there are many operational challenges for autonomous robots that must be overcome; these challenges limit the operational capacity of the UGV. We define the operational capacity of the UGV as the ability to accomplish the mission using the autonomous and semi-autonomous capabilities of the robot. The capabilities of the system are limited by the current state of technology while the capacity of the system is limited by the reliability of the hardware to perform the programmed functions.

Breakdowns that completely incapacitate the robot are one challenge that limits the operational capacity. These breakdowns are not rare; the mean time between failure for ground robotic systems has been reported as between 6 and 20 hours [4]. When a system is operating in a remote location and breaks down, not only does the user lose the robot for mission support, but also must spend time and effort in possibly dangerous areas to recover the unit. While reliability will continue to increase with improved designs and materials, there are many unknown factors that can affect and potentially incapacitate the robot.

Some breakdowns are caused by misuse of the robot, such as overloading a mechanical structure or attempting a mission that requires operation beyond robot design specifications, while others are caused by low-quality parts that break under normal operation, due to manufacturing defects or untested programming bugs. Murphy describes in detail numerous example of robot usage in disaster events [1]. In almost half of the events where robots were used, the robots failed. While only a handful of the robots were permanently lost, each failure slowed rescue operations.

One particular limiting factor for UGV operation is battery life [5]. While there is ongoing research to improve UGV battery life, energy management is critical to maximize a UGV's ability to accomplish its mission. If a battery is depleted, there is little chance for recovery of a robot stranded in a remote location; locating a robot can be a time consuming task if its location is unknown. As such, it is imperative to manage energy resources to allow for mission completion and take action if the mission cannot be completed.

Other challenges that limit operational capacity involve interactions between the robot and the environment. Loss of traction, one such challenge, is caused by wheel slip not being regulated and entering a low traction state. Once traction is lost, the robot could collide with another object due to limited mobility or, if the ground is soft, dig itself into the ground and become immobilized; both cases further limit the operational capacity of the robot. Control systems are needed to track and prevent these situations before the robot is catastrophically incapacitated.

1.1 Problem Statement and Approach

To address the energy and mobility limitations for UGV operational capacity, we focus on the following three problems: limited energy for an area coverage mission, efficient control for a hybrid power system, and prevention of loss of traction.

An area coverage mission requires the UGV to drive within sensor range of every point within the region of interest. This time-intensive mission places high demands on onboard energy stores to cover even small areas. To increase the effectiveness of the UGV, we extend previous path planners to include trajectory generation to maximize energy efficiency. To plan operation, energy usage of the robot at different speeds and on different terrains is required. We propose a method for determining energy efficiency of the robot at different speeds and incorporate these results into the cost function used for planning the velocity profile. By operating efficiently, the UGV is able to decrease the energy required to complete the mission, limiting the chance for battery depletion.

Hybrid power sources allow for increased onboard energy stores while requiring additional control to select which power sources are operating. Each individual component has internal constraints that must be maintained, such as state of charge for a battery or startup/shutdown sequences. Utilizing models of individual power components and knowledge of the mission, we can

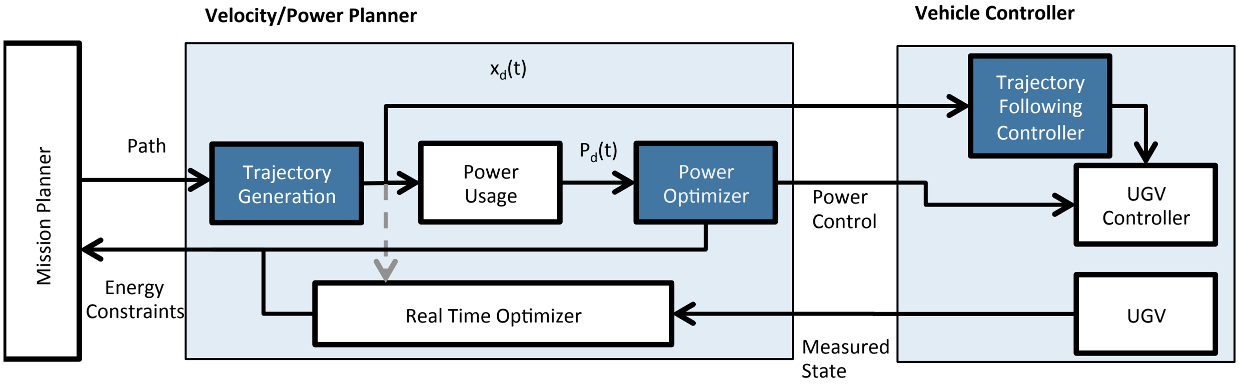


Figure 1.1: General architecture of the proposed energy and mobility management, shaded blocks denote components studied in this work

optimize selection of the power sources to maximize the energy efficiency and mission duration of a UGV. Optimization of hybrid systems is a very time-intensive process due to the discrete switches possible in the continuous dynamics. By making assumptions about the average power use and dynamics, we are able to simplify the problem to a tractable solution and optimally schedule power source usage.

Loss of traction due to wheel slip reduces the operational capacity of the UGV by limiting turning and acceleration. Both lateral and longitudinal wheel slip must be controlled to prevent loss of traction. Using a model of the slipping UGV, based on automotive wheel/ground friction models, we develop two control algorithms: a novel switching controller and a sliding mode controller. The switching controller considers both lateral and longitudinal wheel slip while the sliding mode controller, based on similar automotive techniques, only controls longitudinal wheel slip.

Figure 1.1 shows how these three problems fit in the overall control architecture for a UGV. This control architecture is decomposed into three main blocks: the Mission Planner, the Velocity/Power Planner and the Vehicle Controller. Each component represents an abstract layer of decision making, from the high level goals in the mission planner to low level control algorithms on board the robot. In the case of a teleoperation mission, the Mission Planner and Velocity/Power Planner are replaced by a human operator interfacing directly with the vehicle controller.

The Mission Planner translates the high level objective into a set of tasks to accomplish the mission. Based on operator input and a knowledge of the operating conditions, the Mission Planner produces a traversal path for the robot. For some missions, this might include additional commands for managing peripheral devices such as sensors or manipulator arms. For the area coverage problem, the mission planner produces a series of way points to define robot operation and is based on existing path planning techniques.

The Velocity/Power Planner interprets the path provided by the mission planner and produces a working plan for the robot to actually accomplish. For the area coverage mission, the Trajectory Generation plans the desired trajectory $x_d(t)$ of the UGV over time for the planned path. Using the simulated power usage $P_d(t)$ for the mission, the Power Optimizer schedules operation for the hybrid power components over the entire mission. For simplicity, we assume a known environment that allows for all plans to be executed in accordance with their *a priori* design. While not studied in this dissertation, the Real Time Optimizer can provide on-line updates by prompting adjustment of the mission plan based on additional knowledge gained during operation.

The Vehicle Controller in Figure 1.1 drives the UGV along the desired trajectory $x_d(t)$, manages the power system based on the power commands, and interfaces with hardware elements. Vehicle state is fed back to update the mission as needed. The Trajectory Following Controller implements the wheel slip controllers to prevent loss of traction while following the commanded trajectory $x_d(t)$.

1.2 Contributions and Innovations

This work develops novel control techniques for three robot decision-making elements in Figure 1.1: (1) trajectory generation for a coverage mission (in the Trajectory Generation), (2) power optimization for a hybrid power structure (in the Power Optimizer) and (3) trajectory following control to limit the amount of slip between the UGV and the terrain (in the Vehicle Controller). To develop these three control techniques, this dissertation provides the following **contributions**:

1. **Characterization of Packbot energy usage on different terrains at different velocities**
   Understanding energy usage of a ground robot is fundamental to estimating UGV range and planning energy-efficient operations. In Chapter 3, we present field data from operating a Packbot on different terrains and at different speeds, a comparison of the energy usage for these tests, and specify an energy-efficient operational profile for the robot. This data, also presented in [6, 7], provides information about energy usage and efficiency that can be used for planning.

2. **Tradeoffs of energy considerations in completing an area coverage mission**
   Using the results of contribution 1, Chapter 4 presents an energy-conscious trajectory planning algorithm. Cost function weights are varied to show tradeoffs between mission performance, energy usage, and motor inefficiencies. This work extends previous area coverage planners by providing an optimal velocity profile for existing path planners. These tradeoffs, also presented in [8, 9], can be used to tailor UGV operation under a known set of mission conditions.

3. **Intelligent management of a hybrid power system**

   Long-duration missions require multiple power sources to efficiently complete the mission. In Chapter 5, we demonstrate one method for intelligently managing multiple power sources over the course of a coverage mission to maximize energy efficiency while meeting system constraints, such as battery temperature. Using this controller leads to 5% less energy usage in an area coverage mission, compared to a conservative baseline controller. Initial results were published in [10]. This controller is also extended to prevent thermal shutdown of the battery and tradeoff between energy loss and elevated battery temperature.

4. **Experimental validation of power models and controller with a hardware-in-the-loop configuration**

   To validate the hybrid power controller in contribution 3, we present experimental results in Chapter 5 from testing a fuel cell/battery hybrid power system. By comparing battery performance under averaged and time-varying power demands, we show that the averaging assumption used in the power management strategy is valid. Comparing the energy efficiency of the developed strategy to the baseline controller shows the same 5% improvement in energy usage shown in simulation.

5. **Control of slipping UGV**

   In Chapter 6, we develop a model of a differential-drive ground robot, using established tire-ground dynamics from the automotive literature. Both longitudinal and lateral wheel slip had not been considered previously for this type of robot. This model is used in an optimization control scheme to limit wheel slip and better follow driver-input velocity commands. This controller is compared to a sliding mode controller adapted from the automotive literature. This slipping UGV model and optimization controller were originally presented in [11].

To achieve these contributions, we build on the established literature by providing the following **innovations**:

1. **Standardized method for investigating unmanned ground robot energy usage**

   There currently is no standard method for determining unmanned ground robot energy usage. We propose a method for determining energy usage for ground robots that can be applied to any robot and only requires a simple control structure for running the tests. This method consists of driving the robot at a constant velocity while recording power usage. By comparing energy usage at different velocities on the same terrain, the most efficient velocity can be determined. This method is described in Chapter 3.

2. **Cost function for planning velocity to trade off energy usage and mission completion**

   Building on previous path planning for area coverage missions, we developed a cost function that incorporates energy considerations and mission objectives. By posing the trajectory generation as an optimization problem, we are able to incorporate energy needs for the coverage mission. This cost function incorporates the results from the Packbot energy study and is solved using existing optimization methods. The cost function and the method of solving is presented in Chapter 4.

3. **Framework for optimization over multiple onboard power systems using hybrid automaton models**
   To optimize power usage for systems with long transients, we present a hybrid model framework. In contrast to existing hybrid power controllers, this framework can control systems with discrete dynamics (e.g. on/off) and long transients (e.g. extended startup time). This framework is simplified to a hybrid automaton used to optimize power profiles over the mission for a given cost function by assuming average power demand and dynamics within one state of operation. This simplification speeds up the optimization process and is validated in simulation and hardware. Chapter 5 presents the hybrid systems framework for multiple power sources.

4. **Switching controller and cost function for tradeoff between UGV slip and deviation from user command**
   Using the slipping UGV model, we present a novel switching controller to limit wheel slip and maintain wheel traction. When slipping is detected, a cost function is used to optimize control of the UGV to limit wheel slip and more closely follow the user commands. This cost function uses both the final position and velocity to better meet the desired trajectory. Using this controller allows for consideration of both lateral and longitudinal wheel slip in the same controller. Chapter 6 presents the slipping UGV controller.

1.3 Dissertation Outline

In Chapter 2, related work for each of the three focuses of this research is discussed. Chapter 3 characterizes energy usage for a ground robot and applies this method to a Packbot. Chapter 4 details trajectory optimization for a coverage mission. Using standard optimization techniques reviewed in Chapter 2, a cost function for planning ground robot velocity profiles is developed. Tradeoffs between time, energy and mission completion are examined in different scenarios.

Chapter 5 develops a framework for modeling hybrid power systems using hybrid system models. This framework is applied to a fuel cell/battery hybrid system to manage energy efficiency and thermal battery constraints. Hardware tests, using a simulated robot load, are conducted to validate the models used in optimization and measure the energy efficiency of the power system compared to a baseline controller. The power management framework is extended to include varying the mission power demand to increase efficiency.

To control UGV slip, a dynamical model is developed in Chapter 6. This model, used in conjunction with a cost function including slip and user commands, forms the basis of a controller to limit wheel slip and more closely follow user inputs. Conclusions and possible extensions to this work are presented in Chapter 7.

CHAPTER 2

Background

In this chapter, we review the literature in ground robot modeling, energy use and trajectory planning/optimization that will be referenced throughout the remainder of this dissertation. Section 2.1 discusses UGV control architectures similar to the general architecture (Figure 1.1) used in this dissertation. A basic UGV model is presented in Section 2.2. Sections 2.3 to 2.6 outline related work in optimization and control and provide context for the contributions and innovations of this dissertation. Sections 2.7 and 2.8 outline optimization and modeling techniques used in this dissertation.

2.1 Related System Architectures

The general ground robot planning and control architecture for this dissertation, previously presented in Figure 1.1, is similar to architectures proposed by other researchers. These architectures decompose the complex challenges of operating in an unstructured environment into manageable subproblems. Bonasso et al. present one of the first descriptions of three-tiered robot architecture to control robot operation [12]. The three tiers are labeled Reactive Skills, Sequencing, and Deliberation. The three modules in Figure 1.1, Vehicle Controller, Velocity/Path Planner, and Mission Planner, correspond to these three tiers and perform the subset of activities relating the the topics of this dissertation. Another example of this three-tiered architecture is presented by Bensalem et al. [13].

Thrun describes in detail a three-tiered architecture used in the DARPA Grand Challenge [2]. In this case, the task-level planner takes inputs from the human operator, the execution layer (strictly a path planner in our architecture) senses the surroundings and plans its trajectory to avoid unsafe zones and achieve operator commands. Lower-level controllers directly interface with the vehicle to send commands and process sensor data.

A slightly different architecture is presented by Volpe et al. [14]. In their model, the planning and executive levels are combined into a single decision layer and the functional layer has

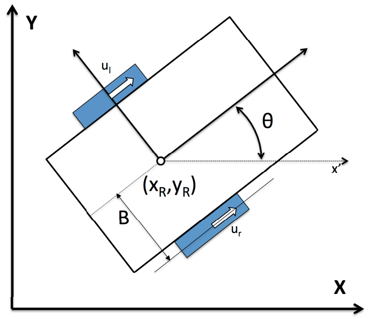


Figure 2.1: Free body diagram for UGV model

additional capabilities. The user or planner can determine if the decision layer should focus on larger mission planning tasks, utilizing additional intelligence built into the functional layer, or on smaller, execution tasks with more specific commands to the functional layer.

The main difference between these architectures and the architecture used in this dissertation is our focus on vehicle energy and mobility management and control necessitating simplification of many of the aspects of interacting with the environment. We assume that the UGV knows exactly how it needs to operate (i.e. path is obstacle free) and doesn't have to react to the environment in which it is operating. This allows us to focus on UGV energy resources and mobility characteristics.

2.2 UGV Model

This dissertation utilizes a simple model for a differential drive ground vehicle. Here we present a basic vehicle model that assumes no wheel slippage; this model is extended to include wheel slip in Chapter 6. System state $\mathbf{x}$ can be described by the robot's location x_R and y_R, forward velocity v, heading θ, and angular velocity ω. Mathematically, this is described as $\mathbf{x} = [x_R, y_R, v, \theta, \omega]^T$. Taking our inputs as u_l and u_r, the forces exerted by the left and right wheels respectively, system dynamics can be described as (adapted from [15])

$$\dot{\mathbf{x}} = f(\mathbf{x}) = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \frac{u_r+u_l}{M} - bv \\ \omega \\ \frac{B(u_r-u_l)}{I_z} - b_r\omega \end{bmatrix}. \tag{2.1}$$

where M is the vehicle mass, b is the linear drag constant, B is the length from the center of the vehicle to the wheel, I_z is the moment of inertia, and b_r is the turning drag constant. We assume the magnitude of each of input $u_{r,l}$ is bounded by u^*_{max} and the velocity is bounded by v_{max}. This model is used in Chapters 3 through 5 for the UGV controller in hardware-based experiments, to derive the optimal area coverage trajectories, and to determine UGV power usage through simulation.

2.3 UGV Energy Usage

Chapters 3, 4, and 5 deal with energy usage of ground robots. Previous research has studied ground robot energy usage in a variety of settings. Mei, Lu, Ju and Lee present a case study comparing energy usage and coverage [16]. They consider an omnidirectional robot covering an open area with no obstacles and using three different coverage schemes: scan lines, spiral, and square spiral. Simulating the robot at different velocities, they calculated the efficiency of coverage paths. By comparing efficiencies at different velocities, the most efficient velocity is determined for the particular robot. Mei et al. also present an experimentally-derived power model of a ground robot at different velocities [17]. This work adds to their earlier power analysis by looking at the other primary battery limitation: total energy storage. Boice et al. present a comparison of power used by the Packbot while traveling over different terrains at different speeds [18]. Using motor current measurements, they conclude that traveling faster draws more current, but no analysis is made for energy use over a distance.

Power use requirements for tracked vehicles is given in [20, 21], including losses due to track slip caused by turning. The power required to drive the vehicle at a constant velocity is modeled as linear in the absolute velocity of track speed. In Chapter 3, the energy characterization methodology examines energy use over a distance, rather than power draw at each point in time. In [21], the authors present experimental results that show similar trends with decreasing total energy usage at higher velocities, but their paths are much shorter and focus on analyzing energy lost due to turning. All of these energy usage studies provide specific examples of power consumption; however, there is no general methodology or benchmarks for determining energy usage for a ground robot. In Chapter 3, we present a vehicle- and mission-independent methodology for characterizing energy usage. The results from applying this methodology to an iRobot Packbot is applied to an area coverage mission in Chapter 4 to increase energy efficiency.

An in-depth, well-to-wheel analysis of an electric vehicle is presented in [22]. Their analysis is based on detailed models of the different energy pathways available in automobiles. While the results of [22] could easily be extended to small ground robots, their research does not deal with the variable terrains that a ground robot encounters. Without a detailed knowledge of the wheel-ground interactions, results of a similar analysis for ground robots could not produce precision necessary for an accurate analysis.

For automotive applications, fuel consumption at different speeds has been studied and modeled [23–25]. Since fuel has traditionally been the main energy source for automobiles, measuring fuel consumption is analogous to the energy usage calculations in this dissertation. In all three papers, a minimum fuel consumption per distance traveled is found in the 60-80 km/hr range, depending on specific vehicle configurations. Since there is a large difference in scale between the vehicles studied in these papers and the ground robots studied in this dissertation, it is necessary to repeat the analysis for ground robots, a study described in Chapter 3. In particular, aerodynamic forces become very large at high speeds for automobiles [26]; for small ground vehicles the low traversal speeds and low drag profile result in low aerodynamic drag.

2.3.1 Projecting Energy Usage

Estimation of ground robot range based on battery energy critically depends on ability to predict future energy needs. Saha, Goebel, Poll and Christophersen present work on estimating range based on historical data and anticipated operation conditions [27]. Bayesian regression and state estimation are used to model battery dynamics and predict the remaining mission life. Sadrpour, Jin and Ulsoy present an algorithm to estimate the energy required to complete the mission based on knowledge of future mission goals and terrain [28]. Prior knowledge about future mission operations can be used to better estimate when the mission can be completed before depletion of the battery

Ceraolo and Pede estimate the remaining range for an electric car based on battery dynamics and estimated future current draw [29]. Zhang, Wang, Kobayashi, and Shirai present a range calculation for electric vehicles based on nine different criteria, including current location, battery state of charge, and driving style [30]. Data from Chapter 3 (or similar tests for different robots) can be used to augment range estimation methods.

For autonomous robots, Berenz, Tanaka, and Suzuki present a method for determining when to return to a recharging station [31]. The authors use a risk/reward analysis to determine the best time to return to the recharging station. One critical part of the algorithm is estimating the remaining range. The energy characterization, described in Chapter 3, can form the basis of range estimation.

In addition to estimating stored energy availability in batteries, the energy consumption of a ground robot can be used to help plan future paths based on energy usage. Ooi and Schindelhauer derive a path planner for a mobile communication network based on minimizing energy consumption [32]. Using the energy characterization method presented in Chapter 3, this method can be augmented for better prediction of energy costs. Similar methods can be used to better optimize other path planners, consistent with ideas presented in [33].

2.3.2 Track/Wheel-Ground Interaction

A major component of ground robot energy usage is energy lost in the wheel-ground or track-ground interactions. While we do not directly measure the wheel-ground forces when characterizing energy use, some results from terramechanics are summarized here to provide context for UGV energy usage results. The theory of terramechanics dates back to the initial work by Bekkar [34]. These principles have been extended by Wong to form the basic foundation of terramechanics [26]. For a detailed analysis of many aspects of skid steering and tracked vehicles, see [15]. From an empirical perspective, we are concerned with two main aspects: tractive effort and rolling resistance.

The tractive effort is the force that the track applies to the ground. For tractive effort, the friction coefficient is based on slip between the track and the ground [26]. While tractive effort and slip are necessary components of understanding UGV mobility, we are interested in energy analysis. In order to estimate the tractive force in our tests, we would need to experimentally determine the slip-friction curve and develop a slip estimator to use during the tests. Research in slip detection is ongoing and not a focus of this dissertation.

Rolling resistance, another focus of terramechanics, is a source of mechanical energy loss for ground vehicles. Track-ground (and wheel-ground) interactions have been well studied in the literature. Wong presents the basic theory of track-ground interaction [26]. For both wheels and tracks, one of the primary sources of energy loss is through compaction of the terrain. The pressure required to compact the soil is key in the derivation. Wong treats this pressure solely as a function of the depth of the track (wheel) sinkage.

Grahn extended the analysis for wheel-ground interaction by including a dependence for the pressure on compaction velocity [35]. The author found that the rolling resistance decreased as the wheel velocity increased from 0 km/h to 16.6 km/h. Pope found a 9.3% decrease in rolling resistance as wheel velocity increased from 0.036 to 0.274 km/h [36]. Crenshaw investigated the rolling resistance in high speed tests and found three regions of interest: decrease in rut depth and rolling resistance with increasing velocity at low (0-37 km/h) and high (above 93 km/h) speeds, and rut depth and resistance increases with velocity at medium (37-93 km/h) speeds [37]. Shmulevich, Mussel, and Wolf summarize the previous results and create a simulation model to predict the effect of velocity on rigid-wheel performance for off-road terrain [38]. Their model showed a decrease in motion resistance ratio for driven, rigid wheels. There were no extensions of the work to track-ground interactions, though we expect similar trends to hold for tracked vehicles. While our energy characterization methodology does not measure energy loss due to ground interactions and we cannot directly compare results to the above research, we expect to observe similar trends in energy usage at different speeds.

In addition to energy loss due to compaction, tracked vehicles encounter energy loss due to skid steering. Guo and Peng present results for energy loss due to steering as a function of turning radius [39]. Using a power loss model for turning with tracked robots, the authors show that at smaller turning radii, there is larger power loss and one track switches from propulsion to braking. Choosing a sufficiently large turning radius limits the difference in torques between the motors and results in limited additional power usage compared to straight line travel. While their tests were performed on sand, we would expect similar results for the terrains used in this research. Their work justifies our use of a rounded box test path with a sufficiently large turning radius to approximate straight-line travel.

2.4 Area Coverage

Chapter 4 deals with optimal completion of an area coverage mission. This mission consists of moving the robot within a set distance of every point in the region of interest. Several different path generation algorithms have been proposed in the literature to solve this problem [40]. Choset describes the Boustrophedon Decomposition (BD) method for partitioning the search region into simple areas that can be completely covered by simple back and forth paths [41]. An adjacency matrix is formed by all the regions and a route is planned from cell to cell.

Gabriely and Rimon approach the problem in a slightly different way [42]. They decompose the search area into square cells twice the size of the search dimension. An adjacency graph is formed over all cells and a spanning tree is computed, possibly a minimal tree based on an assignment of weights between cells. The robot traverses around the spanning tree and covers the entire grid. This path does not guarantee complete coverage because grid cells that are partially occupied with obstacles are not visited. This limitation is removed in [43]. These methods are extended to teams of robots in [44] and [45], respectively. While we do not consider multi-robot systems in this dissertation, the results presented in Chapter 4 can be extended to multiple robots.

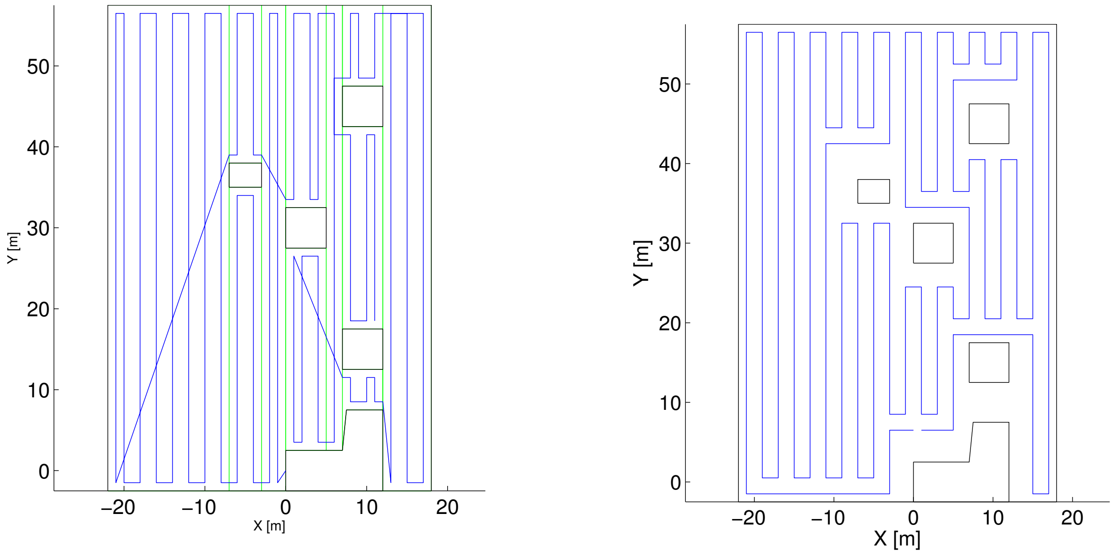
All these methods exclusively look at generating a coverage path and do not attempt to analyze the best way to traverse coverage paths from an energy perspective. Huang presents an extension of the BD algorithm to find the fewest turns in the path, assumed to be the largest energy expense for the coverage operation [46]. This method, however, does not take into account any other energy considerations for UGVs, such as acceleration along the path or motor efficiencies. Chapter 4 extends all of the previous work by taking the defined path and planning mission operation to increase energy efficiency.

A similar problem in coverage for space interferometry is presented by [47]. The authors derive conditions for optimality of fully-actuated spacecraft traveling over Riemannian manifolds. The main result depends on the inclusion of a velocity term in the cost function, since low velocity is required for interferometry. In robotic area coverage missions, the velocity is unpenalized because the same restrictions do not apply. Chapter 3 proposes a cost function incorporating energy considerations and area covered and derives optimal traversal for a ground robot completing an area coverage mission.

Tokekar, Karnad, and Isler report similar tradeoffs for general UGV missions [48]. In their work, the cost function looks exclusively at maximizing energy efficiency and is based on a first-principles motor cost term, compared to the experimental efficiency curve used in Chapter 4.

Figure 2.2 shows an example region for area coverage. Paths for both path planning algorithms are shown. To create the BD path, the region is divided into cells by sweeping a line through the area. When a new obstacle is encountered, one cell is closed and two are opened. When the end of an obstacle is reached, two cells are closed and one is opened. In Figure 2.2a, the green lines represent the divisions into cells. Each cell is covered by simple back and forth movements parallel to the line used to create the cells. These segments are $2R$ apart, where R is the sensor coverage radius from the UGV. To determine the order to visit the cells, we chose a traveling salesman path that minimized the amount of redundant coverage.

To create the Spanning Tree path, the region is divided into cells that are 4R wide. Each cell that is free of obstacles forms a node in a graph. From that graph, a spanning tree is formed and the coverage path circles the spanning tree. We constructed several spanning trees and selected the one that contained the fewest number of turns. Figure 2.2b shows the spanning tree coverage path for an example area.



(a) Path (blue) generated by BD algorithm with cells marked (green)

(b) Path generated by Spanning Tree algorithm

Figure 2.2: Example Boustrophedon and Spanning Tree paths

2.5 Hybrid Power Sources

Hybrid power sources are a major area of interest in the automotive industry [49]. Most of the work is based on a combustion engine/battery hybrid. Ceraolo et al. presents a general approach to hybrid power architectures for automobiles [50]. For cars, the engine typically produces mechanical power and the battery produces electrical power. One of the key design decisions is therefore between a parallel, series, or more complex mechanical power transfer system. Optimization for these different configurations have been studied [51, 52].

For a series configuration, the generator can be run at any desired speed. Barsali et al. present one algorithm for this configuration [51]. In [51], the generator runs in an on-off mode of operation, with the generator operating at the most efficient point when on. Barsali et al. optimize the ratio of on-time to off-time and, based on a desired range of battery state of charge, the actual values of these times. In cases where the generator must be run constantly at a less efficient set point, the average battery level is held constant.

For parallel configurations, the speed of the engine cannot be chosen due to mechanical coupling with the wheels. Lin et al. present a power management strategy for a parallel hybrid vehicle [52]. In this strategy, the power split between the battery and the engine is obtained using an optimal control technique to maximize fuel economy and minimize emissions while maintaining the state of charge of the battery.

Rodatz et al. present a fuel cell/supercapacitor hybrid vehicle [53]. In their work, their control objective is to minimize the hydrogen consumed by the fuel cell over the a given mission. Using a strategy called the equivalent consumption minimization strategy, the global optimization problem can be approximated by a local minimization problem. This local problem includes power coming from the supercapacitor as well as the fuel cell and can be applied to real scenarios where the drive cycle is unknown.

For the UGV systems that are considered in our work, all of the power components produce electrical energy directly; there is no need for a series/parallel distinction. However, the fuel cell that we consider does not work with the above mentioned optimizations due to discrete on/off nature of its operation, motivating the design of our optimization framework.

More recently, Murphey et al. presented a power management scheme for a vehicle with multiple power sources [54]. Each individual source can be turned on or off, in addition to any throttling allowed by the device. Using a machine learning algorithm, the controller can decide at each time step which power sources are the best to use. While this algorithm has the same purpose as our algorithm described in Chapter 5, there are two key differences. First, their model assumes that the power sources can be turned on and off instantaneously. Second, their optimization looks over a short time horizon and not over an entire mission.

While current UGVs are almost universally battery powered, recent research has investigated replacing or augmenting the battery system with a fuel cell. Wilhelm et al. present a UGV powered exclusively by a fuel cell [55]. Their robot was quite small, using a 10 W fuel cell, and served as a proof of concept. Joh et al. present a humanoid robot powered by a fuel cell and a battery in parallel [56]. The authors demonstrate the use of their robot, including the use of the battery to supplement the fuel cell when the power demand exceeds the capacity of the fuel cell. In this work, the fuel cell is able to vary the power output and there is no discussion of charging the battery when power demands are low.

2.6 Slip Control

Wheel slip is a fundamental challenge of wheeled locomotion. Extensive wheel slip research has been conducted in both automotive and wheeled robotic settings. There are several different models of tire/ground interaction [57]. Since we are focused on a robotic system with both longitudinal and lateral slip, we used the formulas presented in [58] to develop our friction model for slipping.

Wheel slip is a known complication for position tracking in wheeled mobile robots. In [59], the authors study a three-wheeled robot with omni directional steering. They present a dynamic model of the system and compare tracking performance of a Proportional-Integral-Derivative (PID) controller with a model-based controller, demonstrating the need for incorporating slip into control laws. Williams et al. analyze the friction model for this class of wheeled robots [60]. Wang and Low present a control design perspective on skidding and slipping for different configurations of wheeled robots [24]. Their analysis focuses on the controllability of the different configurations but only analyzes a kinematic model. We derive a dynamic model of a slipping UGV in Chapter 6 that can better model actual operation of the UGV.

Sarkar and Yun present a traction control algorithm for a robot with two drivable and steerable wheels [61]. While they present slip and traction control results for this class of robots, we are motivated by a differentially-driven robot for our research. An adaptive nonlinear controller for tracked robots with longitudinal slip is presented in [62]. This model focuses on a kinematic model for tracked vehicles, as opposed to a dynamic model. All of these papers present slip controllers for different configurations of wheels and tracks. The controller presented in Chapter 6 extends this work for a differential-drive robot.

A differentially-driven robot with lateral slip only is studied in [63]. A lookahead point is defined in front of the robot and the control objective is to drive to the lookahead point. Using this control law, they are able to decrease the lateral motion of the robot when turning. Chapter 6 broadens their work by including longitudinal slip to prevent loss of control.

Nandy et al. present a dynamic model for a differentially-driven mobile robot with slip [64]. This model forms the basis of our slipping UGV model in Chapter 6; we expand their model by using a different method of eliminating constraints [65] and a different friction model [58].

Control of wheel slip is a well studied topic for automotive applications, in the form of anti-lock brake and traction control systems. For controlling longitudinal slip, a common control method is to use sliding mode control [66–68]. Additionally, vehicle stability controllers use differential braking to achieve better turning [69]. While many of the issues are similar between automotive and robotic settings, the different configuration of wheels and the difference in scale between ground robots necessitates different control approaches. In Chapter 6, the sliding mode controllers presented in [66, 68] are extended to the slipping UGV framework and compared to the switching control strategy developed in that chapter.

For physical operation of robots, the slip and friction values need to be measured. A lateral slip and force estimator is presented in [70]. Dar and Longoria present a method for slip estimation in tracked vehicles [71]. Two different wheel slip detectors are presented in [72, 73]. These detectors are different from the previous work in that they only output if the wheel has traction or is slipping; there is no estimate of the wheel slip value used to calculate friction forces. These methods can be used with the proposed controller in Chapter 6 to determine when the controller should switch modes.

2.7 Optimal Control

Optimal control for dynamical systems is a well established field. We use optimization in Chapters 4, 5, and 6 to solve the three main challenges discussed in Chapter 1. In this work, we use three optimization algorithms. The first two methods are explained here as they are general methods. The third is described in Section 5.1.2.2 after defining relevant mathematical symbols.

The first method used in this work is the calculus of variations, an analytical derivation of the optimal solution for a system. The methods described in [74] are summarized here for completeness. For a system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$ and a cost function $J = h(\mathbf{x}_f, \mathbf{t}_f) + \int_{t_0}^{t_f} g(\mathbf{x}, \mathbf{u}, t)\,\mathrm{d}t$, the Hamiltonian $\mathscr{H}$ is defined as

$$\mathscr{H}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) = g(\mathbf{x}, \mathbf{u}, t) + \mathbf{p}^T[f(\mathbf{x}, \mathbf{u}, t)] \tag{2.2}$$

where $\mathbf{p}(t)$ are the costate variables. As derived in [74], the optimal state $\mathbf{x}^*(t)$ and costate $\mathbf{p}^*(t)$ can be found by

$$\dot{\mathbf{x}}^* = \frac{\partial \mathscr{H}}{\partial \mathbf{p}}(\mathbf{x}^*(t), \mathbf{u}^*(t), \mathbf{p}^*(t), t) \tag{2.3}$$

$$\dot{\mathbf{p}}^* = -\frac{\partial \mathscr{H}}{\partial \mathbf{x}}(\mathbf{x}^*(t), \mathbf{u}^*(t), \mathbf{p}^*(t), t) \tag{2.4}$$

$$\mathbf{0} = \frac{\partial \mathscr{H}}{\partial \mathbf{u}}(\mathbf{x}^*(t), \mathbf{u}^*(t), \mathbf{p}^*(t), t) \tag{2.5}$$

These equations can be solved by a boundary value problem solver, such as `bvp4c` in MATLAB, for a given set of initial and final state values and final time. Additional constraints exist if the final state value and time are allowed to vary. This method is used in Chapter 4 to solve the cost function for area coverage trajectory planning.

The second method used is numerically approximating the optimal solution. In this case, a gradient-based method is used to find a local minimum of a given function. In this work, we use `fmincon` in the MATLAB Optimization Toolbox. The cost function is evaluated at different operating points and the cost function is minimized by using the gradient to find the next operating point to test while maintaining system constraints. With this method, the resulting solution is not guaranteed to be globally optimal so care must be taken to ensure that good initial conditions are given to the algorithm to find an acceptable locally-optimal point. See [75] for an in depth description of these function minimization algorithms. This method is used in Chapter 6 to solve the slipping UGV optimization problem.

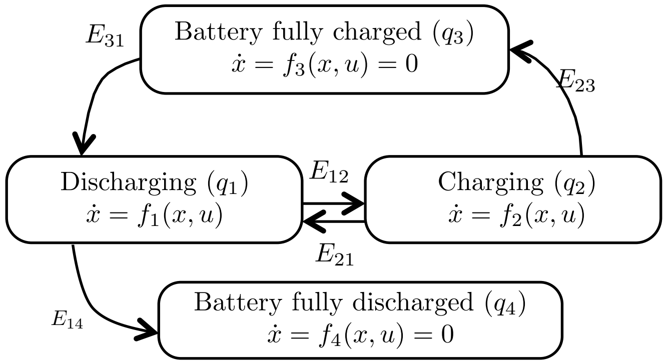


Figure 2.3: Example of a battery model with three discrete states

2.8 Hybrid Control Systems

The field of hybrid control systems, consisting of both continuous and discrete dynamics, allows system with discrete modes and continuous state dynamics to be analyzed [76, 77]. Optimization of general hybrid systems have been studied in depth [78–81]. These methods are computationally intense. In Chapter 5, we are considering full mission operation due to the long transients present in the considered fuel cell system. Initial attempts to optimize the full system proved computationally infeasible and, in this paper, we present a method to simplify the problem to a linear optimization that can be solved quickly.

In Chapter 5, we create a simpler optimization problem by using hybrid automaton models [82]. This model is composed of a set of discrete states with defined transitions between the states. In each discrete state, the continuous states evolve at a known rate. The discrete-state transitions are governed by the model inputs and the continuous state.

To use the hybrid automaton in an optimization problem, each automaton state and transition of the model is appended with a cost value. The cost of a trajectory for the system is given by the sum of costs of the taken transitions plus the sum of the costs associated with the visited states multiplied by the time spent in each of them [83]. The full mathematical description of the hybrid control system used in this dissertation is presented in Chapter 5.

Figure 2.3 shows an example hybrid system for a battery. In this model, the dynamics are divided into charging and discharging states. When the battery is charging and reaches full capacity, the discrete state transitions to the *Battery fully charged* state. Likewise, when the battery is discharging and the battery becomes fully depleted, the state transitions to the *Battery fully discharged* state. The dynamics for each state are defined as f_i and are a function of the state x and input u. These functions can be derived from first principles or fit to experimental data.

CHAPTER 3

Energy Usage Characterization for a UGV

Battery energy storage is a limiting factor for electrically-powered unmanned ground vehicles (UGVs). If the battery is completely depleted, the robot is stranded and must be retrieved by a human operator, potentially in dangerous situations. In order to prevent battery depletion, it is necessary to estimate how much energy will be used over the remainder of the mission. Locomotion is one major energy expenditure.

Current fielded robots rely on different methods of avoiding battery depletion and UGV immobilization. The most basic, for semi-autonomous or human controlled missions, is a range estimation based on current battery state of charge and estimating energy usage based on past data (see [28] for example). In this case, the UGV range estimate is relayed to the user, who can make decisions about aborting the mission. One autonomous system can return to a base station to recharge (see [31] for one example of scheduling charging based on a risk/reward analysis of running out of energy). In both of these cases, an understanding of energy usage is necessary for the methods to function.

Here, we focus on evaluating currently fielded robots. There exist a number of commercially available robots (such as the Packbot and TALON) and many more research platforms at laboratories throughout the world. Many commercial platforms are in operation in military, police, and search and rescue operations. Based on discussion with our research collaborators in the US Army, the energy characterizations presented in this chapter are needed to maximize the performance of robots that are currently in the field.

In this chapter, we present a methodology for determining energy usage by a small ground robot and apply this methodology to the Packbot as a case study. This methodology comprises Innovation 1 of this dissertation. We also present Contribution 1 of this dissertation: energy usage characterization for a Packbot. This methodology is based on empirical energy data recorded during UGV operation, not determined from a terramechanics model of the system. While this methodology is general, it is particularly helpful for characterizing and understanding commercially available robots, where internal configurations are unknown due to proprietary constraints ("black box"). This methodology characterizes energy usage by determining the following information: 1) energy usage per distance traveled at a given speed, 2) the most efficient speed on a given terrain, and 3) a terrain independent electrical-to-mechanical efficiency curve. From this data, a path planner can optimize the path based on the known terrains and mission goals to reduce energy consumption. Additionally, this methodology can be used to compare different robots for a given application or aid in designing a robot to meet certain mission goals. Our methodology consists of conducting a series of tests at a constant forward velocity; to create a sufficiently long path, we introduce a "rounded box" path.

This methodology is not based on a knowledge of the internal configuration of the robot. The methodology only requires a way to drive the robot on a specified path at a constant velocity and record the electrical and mechanical energy used during the test. In the case of the Packbot presented in this work, we did not have access to the internal details, due to the proprietary black box nature of the robot, and could only calculate the mechanical energy to within a scaling factor.

Related work for this chapter is summarized in Section 2.3. Section 3.1 presents the proposed methodology for characterizing the energy usage of a ground robot at a constant speed. Section 3.2 describes the Packbot platform used as a case study for this methodology. Results from the tests on the Packbot are presented in Section 3.3. The results in this chapter have been published in [6, 7].

3.1 Testing Methodology

As ground robots become more widespread, there is a much greater need for methods and standards to analyze robot performance. In this chapter, we are interested in one particular aspect of robot performance: energy usage while moving. Previous standards have been focused on robot speed [84], common terrains [85], and obstacles [86]. While these standards are beneficial, they do not address energy usage. Our method adds the metric *energy used for motion*.

Energy usage for straight-line motion and turning must both be characterized. The test methodology presented in this chapter is general but our tests focus on straight-line energy usage on consistent, traversable terrain. One example of predominantly straight-line motion, and the motivation behind our work, is an area coverage task. The coverage task consists of long, straight segments connected by short turning segments. The energy usage is a function of many variables; this methodology considers speed and terrain as the primary variables affecting energy use.

The methodology for determining energy usage consists of selecting a series of tests, recording the necessary data during the tests and processing the data to analyze energy usage after the tests. Figure 3.1 depicts the important aspects of our method. First, a set of test paths and trajectories are designed based on the desired velocities and terrains for which energy use will be characterized; wide turns can be introduced to the planned paths based on area constraints. Using

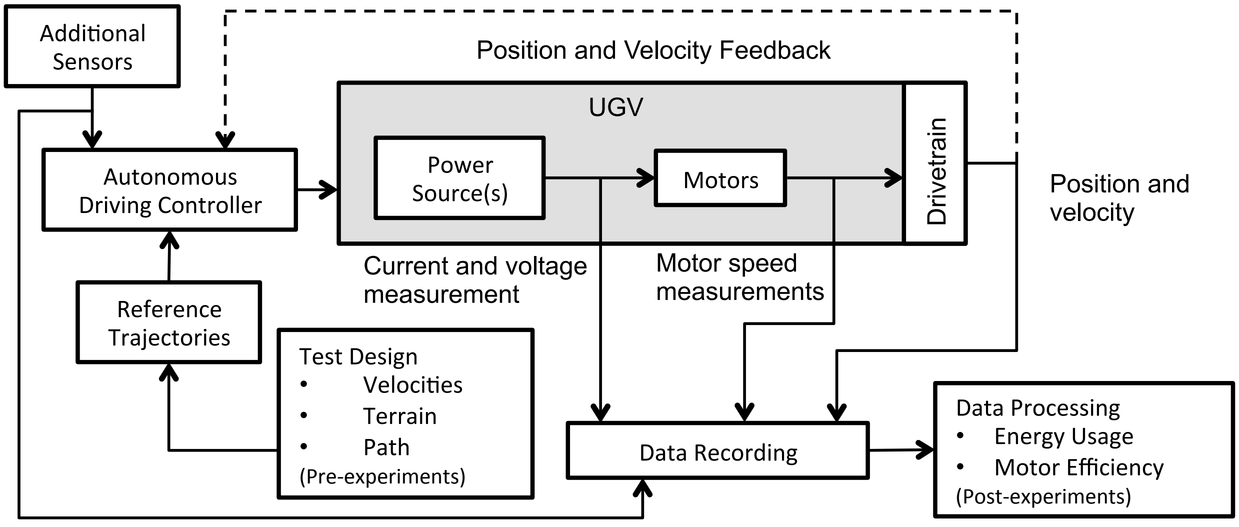


Figure 3.1: Flow of methodology

an autonomous or semi-autonomous controller, the UGV is driven based the computed trajectories. If necessary, the robot can be augmented with additional sensors, e.g. GPS for localization, to complete the feedback controller. During the tests, the electrical voltage and currents, motor speeds and distance traveled are recorded from the UGV. Additional sensors, such as GPS or inertial measurements, may also be recorded and used by an autonomous driving controller to follow the reference trajectory for each test. After all the tests are run, the recorded data is analyzed to produce the following characterizations:

1. Energy usage per distance traveled for a specific terrain

2. Most efficient speed for straight-line travel for a specific terrain

3. Terrain-independent electrical-to-mechanical efficiency curve

We chose as the metric energy usage per distance traveled, instead of instantaneous power usage, to analyze long-duration missions for ground robots. As mission duration increases, total energy stored becomes more important than instantaneous power draws since energy used per distance shows the most efficient way to travel between different points. While power can be integrated to measure energy usage, measuring energy usage per distance changes the metric from a time-based measurement to a distance based measurement.

To characterize ground robot energy usage, we propose a series of tests, each with the robot traveling at a constant speed. Ideally, these tests would be run in a straight line to avoid any additional energy loss due to turning. Because of limited operating areas, a “rounded box” path can be used. This path consists of straight sides with sufficiently large radius turns between the sides. The sides of the box should be as long as possible given operating area constraints. In order to obtain sufficient data, the total length of the path should be a significant portion of the vehicle's range.

Turns should have a radius sufficiently larger than the robot track or wheel base to minimize the amount of lateral slip at the contact points. Using a power consumption model for turning, such as the one presented in [39], the minimum radius that meets a desired power tolerance (i.e., less than 10% increase in power draw due to turning) can be computed. While the radius will depend on terrain and robot characteristics, we found that a turn radius of about 8 times the track length was sufficient for our setup; we expect other robots to be similar.

For each of the independent variables, speed and terrain, a set of desired test points must be defined. For speed, a maximum and minimum speed should be chosen based on robot capabilities and the desired application. Intermediate speeds can be chosen based on the fineness of data desired; more tests produce more data points at the cost of additional resources required for testing. Five to eight test speeds will provide a good tradeoff between quantity of data recorded and testing duration requirements for ground robot applications where test time is constrained.

Terrain selection for the tests should be based on the required application. While there will always be variations in terrain parameters in real-world applications, a set of basic terrains can be chosen to predict general performance of the robot. This set should include at least one non-deformable terrain (e.g. asphalt) and one deformable terrain (e.g. grass or sand) for a general robot designed for outdoor off-road missions.

The ground robot should be driven at the desired speed. This can be achieved through purely autonomous operation or semi-autonomous operation, with the speed regulated by the controller and a human steering the robot. Autonomous operation requires more set up and sensors, but provides more repeatability between tests. If the robot already has autonomous capabilities, this entails programming the robot to follow the path at the desired speed. Otherwise, an additional controller, either on board the robot or on an external computer, must be designed to drive the robot on the prescribed trajectory. If necessary, a localization sensor, e.g. GPS, can be added to complete the autonomous feedback. In both cases, the controller must be well designed to limit overshoot in steering and maintain the reference trajectory in the presence of measurement noise. For semi-autonomous operation, a user interface must be provided that allows for selection of the speed.

There are two main ways of measuring energy usage: change in energy storage (e.g. battery state of charge) or integrating instantaneous power measurements over time. The battery or other power source must provide up-to-date energy storage, while the instantaneous battery power can be calculated from the battery voltage and current measurements, if a direct battery power measurement is not available. Instantaneous mechanical power can be calculated from the motor speed and torque, which is proportional to motor currents.

After the set of tests is complete, the recorded data is analyzed to compute the energy per distance traveled and the efficiency of the electrical to mechanical conversion, which includes the motor and additional power component efficiencies. While the total average energy per distance for the entire trajectory is a useful metric, breaking the tests up into shorter segments enables extraction of more information about the statistics of energy usage. The length of the shorter segments should be based on the quality of the sensors used: tests with higher quality sensors can be broken into shorter segments while still limiting sensor noise impact on results. The electrical-to-mechanical efficiency is calculated by dividing the mechanical energy output by the electrical energy input and represents losses due to the motor, power electronics, and other drivetrain components.

Alternatively, the cost of transport, defined as the energy used per meter per kilogram, can be calculated by dividing the energy per distance by the platform mass. This metric is commonly used to assess and compare bipedal robots (for example, see [87]). While this makes the total energy used dependent on the current system configuration, it also allows for estimating energy usage when the UGV has an increased mass due to additional loads (e.g. sensors, manipulators, etc).

This methodology computes an energy characterization for straight-line travel of a ground robot. It does not attempt to study complex ground-robot interactions, but creates an empirically-determined energy characterization of driving. For complex and variable terrains, these tests can serve as a basis for estimating energy usage by determining a scale factor as the robot operates. We leave a derivation of such methods as future work.

As the field of ground robots continues to develop, a standard set of proving tests can be created to compare different robots in the different settings, similar to specifications used in the automotive industry (for example, [88]). These standard tests can be devised through robotic competitions, as described by [89].

3.2 Packbot Platform and Test Setup

For our tests, we used an iRobot Packbot, shown in Figure 3.2. The University of Michigan Mobile Robotics Lab (UM MRL) provided an external interface to the proprietary Packbot interface. The UM MRL interface accepts forward velocity v and angular velocity ω commands and provides data feedback from the Packbot. This interface is run on a Gumstix processor attached to the Packbot and all communication takes place over a wireless network, using a standard router located on the robot. Both devices are powered by the Packbot battery and consume less power than the drive system when active.

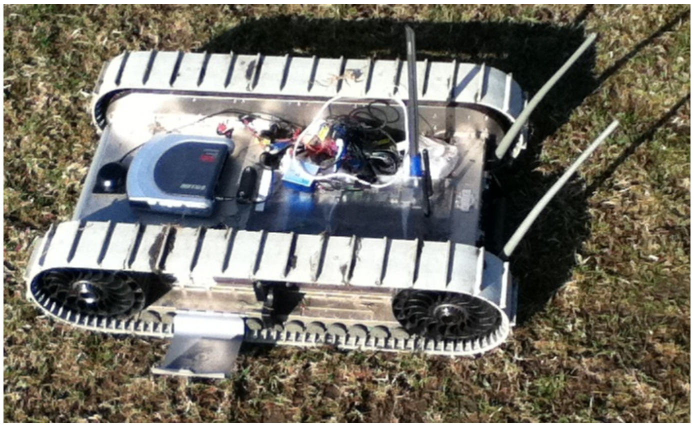


Figure 3.2: Packbot used in experiments

The UM MRL interface also provides real-time data from the Packbot, including the electrical and mechanical energy usage information discussed in the previous section. In this methodology, we use two different energy values: electrical energy input and mechanical energy output. The mechanical power is calculated from the motor current, which is proportional to motor torque and the motor shaft speed. Since the motor speeds are not provided by the interface, they can be calculated from the ground robot forward and angular velocities, as calculated by the interface from the wheel encoders, using the Packbot width $2B$ and track drive wheel radius R. The UM MRL interface thus provides

$$\begin{aligned} P_{\text{mech}} &= K_m K_r \left(I_l \omega_l + I_r \omega_r \right) \\ &= K_m K_r \left(I_l \frac{v - \omega B}{R} + I_r \frac{v + \omega B}{R} \right) \end{aligned} \tag{3.1}$$

where K_m is the motor constant and K_r is the gear ratio from the motors to the drive wheels. Due to proprietary data restrictions, K_m and K_r are unknown values. We therefore utilize the scaled mechanical power $P_{\text{mech}}/(K_m K_r) = I_l \omega_l + I_r \omega_r$. Since we are dealing with a black box model of the robot, we do not know if the motor current measurements occur directly before the motors or if there are additional electronic components between the sensor and the motors. Therefore, we treat the motor and any additional electronics together as a unit. This may result in inaccurate mechanical power calculations from (3.1); however, we will assume that any errors are small in the range that the motors are being operated.

To filter the motor current data, we used Tikhonov regularization to reduce spikes in the measured current [90]. We imposed smoothness constraints on the current values to limit current spikes in the recorded data. Regularization minimized the following equation:

$$|I_{r,l} - I_{op} w_{r,l}|^2 + \lambda^2 |L w_{r,l}|^2, \tag{3.2}$$

where $w_{r,l}$ is the first order difference in the respective measured currents, L is the differential operator, I_{op} is the integral operator, and λ is the regularization parameter. Each motor current was smoothed individually. We selected $\lambda = 1$ to reduce motor current spikes without flattening the

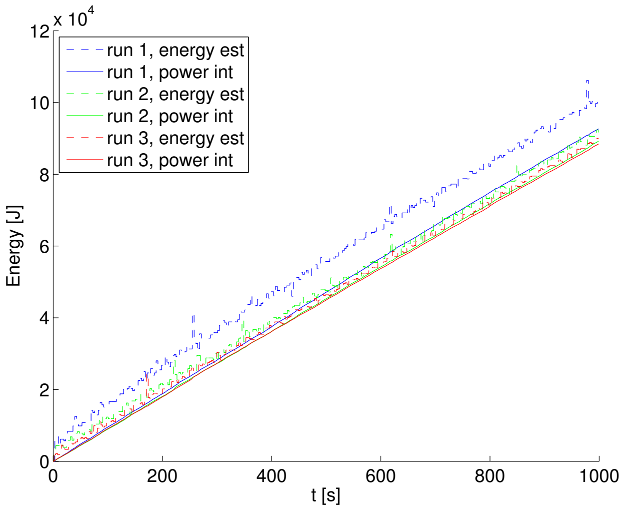


Figure 3.3: Integrating battery power produces comparable results to the battery energy estimate

data significantly. This method was chosen over a simple low pass filter to limit the amount of data lost due to filter transients.

The Packbot data allows two methods to measure electrical energy usage: 1) estimation of remaining battery life and 2) the integral of the instantaneous battery power draw. For our setup, the Packbot estimates the battery energy remaining, presumably based on the state of charge of the battery, and the battery draw. We record the battery draw and integrate over time to produce our own estimate of power used. Figure 3.3 compares these two energy calculations for three consecutive experiments on the same battery charge. As described in detail on page 29, these trials consist of the Packbot running at a constant speed for a certain distance. The energy used for the second and third trials is nearly identical and the first run ends with a similar slope between the two energy estimates. The discrepancy in the first trial is presumed to be caused by overestimating the energy remaining when the battery is mostly full. Similar trends are present in other sets of consecutive experiments. Also note that the battery energy estimate fluctuates up and down over the course of the trial. Because of these observations, we chose to use the integral of the battery power as our method to calculate the electrical energy used by the Packbot.

Additionally, the battery power draw includes power for peripheral electronics. To measure the auxiliary power, we recorded the power data while the Packbot was stationary with all electronics operating for about 6000 seconds. The average power usage over test was 30.8 Watts, with the maximum and minimum measurements 32.1 W and 30.1 W respectively. To calculate the energy used for motion, we subtracted the average power for peripheral devices from the battery data field before integrating.

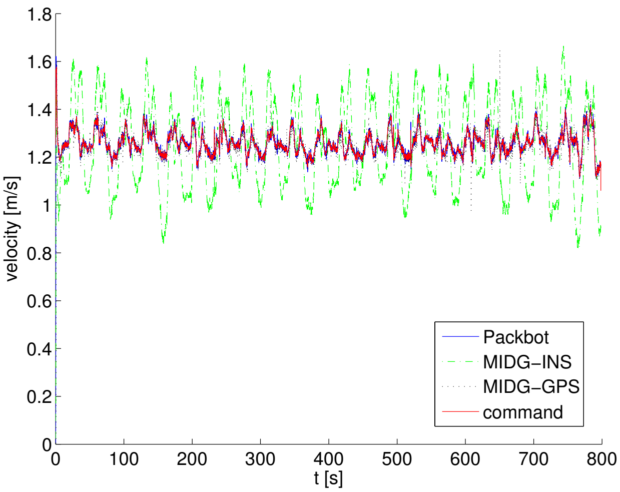


Figure 3.4: Comparing forward velocities from Packbot and MIDG, $v = 1.25\ \mathrm{m/s}$

To examine differences due to current battery state, we ran each test several times on the same battery charge. The tests were run on flat terrain. Figure 3.4 shows the velocity profile for one test on asphalt using three different velocity data fields recorded during the tests: track encoders (Packbot), Global Positioning System (GPS) and Inertial Navigation System (INS) values. The periodic variations are due to the slight slope in the surface used for testing.

For localization, the Packbot was augmented with a MIDG IIc INS [91], which includes a GPS sensor. The MIDG provides 50 Hz localization data from an Extended Kalman Filter, integrating GPS, accelerometer, gyro and magnetometer data to provide three-dimensional location and attitude information. Both the raw GPS and filtered INS outputs were recorded. We ultimately turned off the magnetometer input to the Kalman filter due to interference from the Packbot, after which the heading was estimated from GPS and the integral of angular velocity about the vertical z axis (turn rate). An additional Gumstix processor interfaces with the MIDG and relays the data over the wireless network. The MIDG and its electronics are powered by an external battery.

To drive the Packbot along a reference trajectory, we used the following nonlinear trajectory–following controller to calculate the commanded forward velocity v_c and turn rate ω_c [92]:

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e + K_x x_e \\ \omega_r + v_r \left(K_y y_e + K_\theta \sin\theta_e\right) \end{bmatrix} \tag{3.3}$$

In this controller, v_r and ω_r are the reference forward velocity and angular velocity, θ_e is the heading error, x_e is the longitudinal position error, y_e is the lateral position error, and K_x, K_y, and K_θ are control gains. This controller is proven to be stable using a Lyapunov analysis [92]. For these experiments, we use the position $(x, y)_{INS}$ and heading θ for position control.

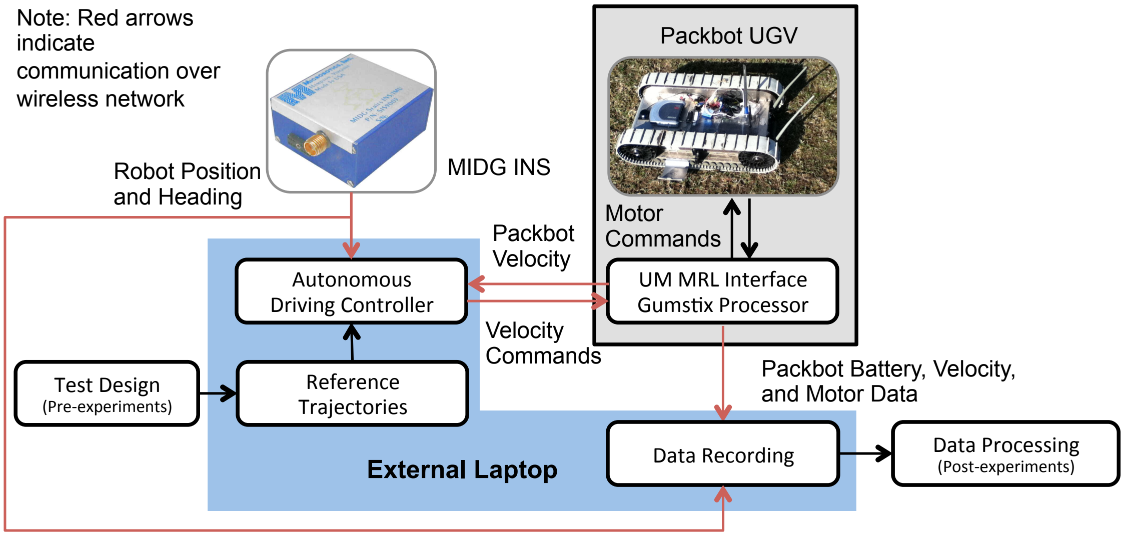

Note: Red arrows indicate communication over wireless network
MIDG INS
Packbot UGV
Robot Position and Heading
Motor Commands
Packbot Velocity
Autonomous Driving Controller
UM MRL Interface Gumstix Processor
Velocity Commands
Test Design (Pre-experiments)
Reference Trajectories
Packbot Battery, Velocity, and Motor Data
External Laptop
Data Recording
Data Processing (Post-experiments)

Figure 3.5: Schematic of data flow for the Packbot setup

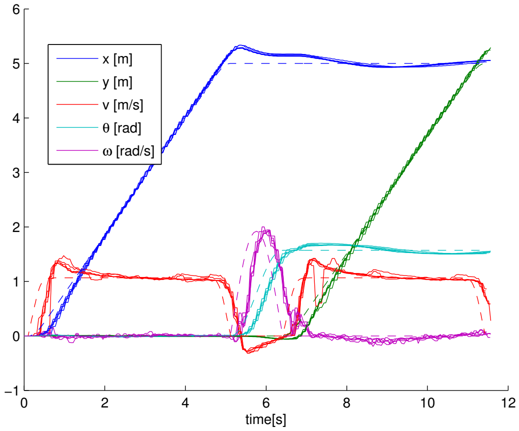


Figure 3.6: Minimal delays in Packbot while completing an L-shaped trajectory

Figure 3.5 shows the different subsystems active for our test setup. This figure expands the on-line portion of Figure 3.1 to show the details of the test setup. The controller and data recorder are implemented on an external laptop, accepting data over a wireless network from the MIDG and the UM MRL interface. With data being acquired by two different systems, the data must be synchronized. Driving commands are calculated based on a reference trajectory and position updates from the MIDG and are transmitted over the wireless network.

This controller assumes that the robot can immediately respond to new commands, though in reality there are communication and acceleration delays. Figure 3.6 shows the Packbot following an L-shaped trajectory of moving forward 5 meters, turning 90 degrees, then moving another 5 meters forward. The dashed lines represent the reference trajectory. The actual trajectories from

Table 3.1: Data fields recorded during tests

Measurement	Source	Variable	Units
velocity	Packbot	v	m/s
turn rate	Packbot	ω	rad/s
battery power draw	Packbot	P_{batt}	W
battery energy remaining	Packbot	E_{batt}	J
motor currents	Packbot	$I_{r,l}$	A
position (GPS and INS)	MIDG	$(x, y)_{GPS,INS}$	m
heading	MIDG	θ	degrees
velocity (GPS and INS)	MIDG	$(v_x, v_y)_{GPS,INS}$	m/s

five different runs are shown in the figure and consistently have a 0.3 second delay in responding to commands. Notice that the delay also causes overshoot on the velocity as the robot tries to return to the desired location. We decided that this delay was acceptable for our application.

We executed the rounded box paths on grass and asphalt surfaces to compare the energy use on different surface types. The asphalt presents a hard surface that does not exhibit sinkage and provides good traction for the Packbot. The grass allows for more slippage and sinkage. The initial grass tests, denoted as “soft grass,” occurred in the winter on a sports field.

The rounded box path was 35 m long and 15 meters wide with a corner radius of 6 m. The size of the box was based on limitations of the area under consideration and our test setup. Based on experiments presented by [39], as discussed in Section 2.3.2, we determined that the 6 m corner radius would not introduce significant energy loss due to turning. For our robot, this radius is 8.6 times the track length and 15 times the separation between the tracks.

For the Packbot, the total length of the path for our tests was chosen to be 1000 meters, approximately one quarter of the Packbot’s range. Figure 3.7 shows the executed path of 11 laps around the rounded box.

The tests were run at at the following speeds: 0.75, 1.0, 1.25, 1.5, and 1.75 m/s. The Packbot used in these experiments had a top speed of approximately 2 m/s. We decided to limit our tests to 1.75 m/s to allow the Packbot to drive faster than the nominal speed if needed to maintain the planned trajectory. We did not go lower than 0.75 m/s because we were interested in looking at speeds that would be practical while covering large fields; slower speeds would mean prohibitively large energy losses due to constant electrical loads.

Additionally, we had access to a grass runway at a local airport. We were able to run the Packbot straight along the length of the runway, as well as running the “rounded box” configuration at the same location to compare energy usage between straight and turn trajectories, verifying the assumption that the turns do not introduce significant energy loss. These tests are labeled as “hard grass” since they were run in the summer when the ground was hard. The length of the runway is

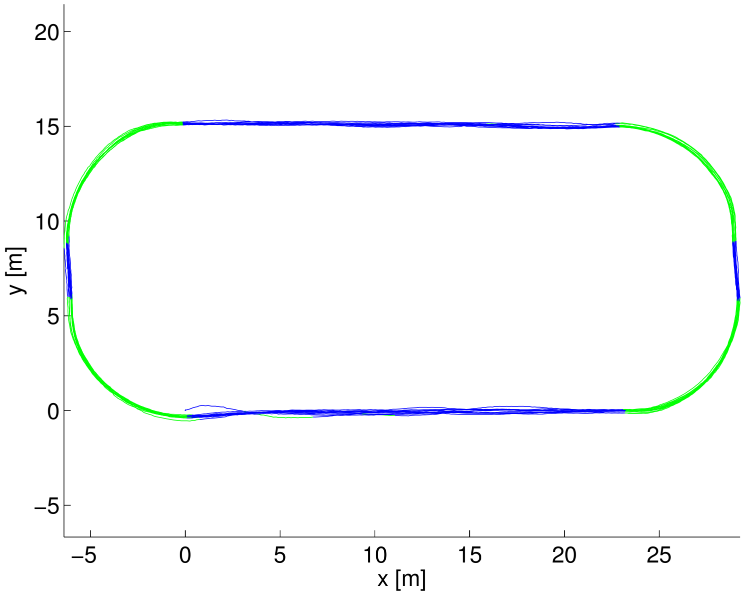


Figure 3.7: Rounded box trajectory, divided into straight and turn segments based on commanded turn rate, $v = 1.25\ m/s$, on asphalt

Table 3.2: Summary of energy-characterization tests conducted

Surface	Path	Speeds
Asphalt	Rounded Box	0.75 m/s, 1 m/s, 1.25 m/s,1.5 m/s,1.75 m/s
Hard grass	Rounded Box	0.75 m/s, 1.75 m/s
Hard grass	Straight Line	0.75 m/s, 1.75 m/s
Soft grass	Rounded Box	0.75 m/s, 1 m/s, 1.25 m/s,1.5 m/s,1.75 m/s

1.04 km, though not all runs were for the full length of the runway. For simplicity in setting up the straight line tests, we used a constant speed controller with human input for steering to keep the Packbot on the runway. Table 3.2 summarizes the tests conducted to explore Packbot energy usage.

3.3 Test Results and Analysis

Here, we present the data acquired following the methodology described in Section 3.1. Specifically, we present the energy usage per distance traveled for a set of speeds on two different terrains and the terrain independent electrical-to-mechanical efficiency curve. This data comprises Contribution 1 on this dissertation.

Figure 3.8 shows the electrical and scaled mechanical power usage for a portion of a test on asphalt, with turn and straight segments delineated in different colors. We used the turn rate com-

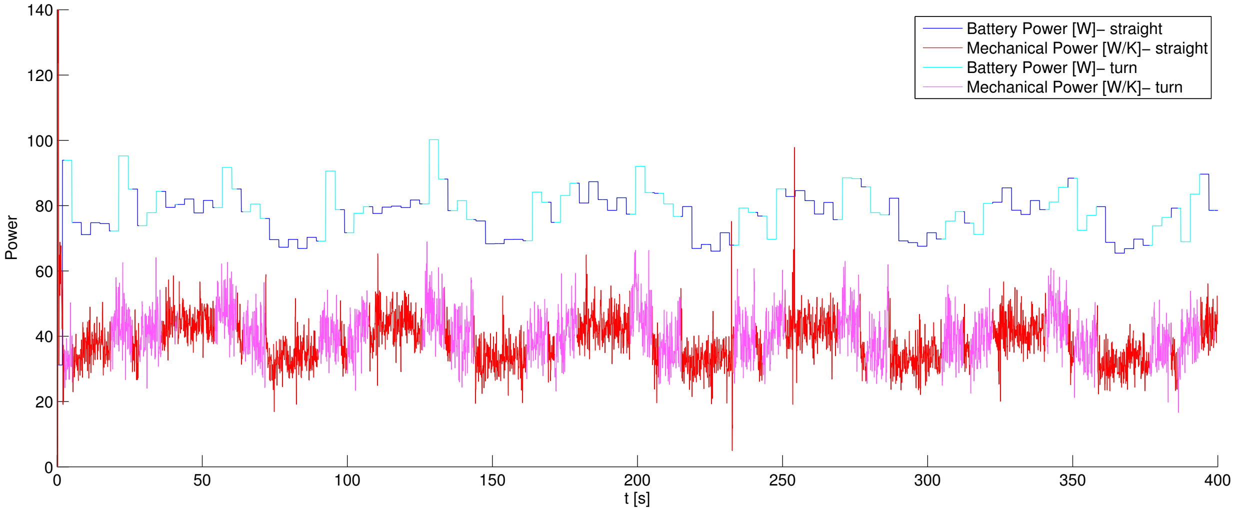


Figure 3.8: Electrical and mechanical power over time for one run, $v = 1.25\ m/s$ on asphalt

mand to differentiate between the different segments, using a threshold value of $\omega_{thresh} = 0.05$ rad/s. The resulting $x - y$ plot is shown in Figure 3.7. While it is difficult to see from the graphs, the turn segments do have a higher average power use. Table 3.3 summarizes the average power used at different speeds and surfaces for the straight and turn segments. Based on this data, we conclude that the turns do introduce extra power draw, although the increase is less than 7%. The periodic variation visible in Figure 3.8 is due to a slight slope of the surface where the test was conducted.

3.3.1 Energy Usage

For each run, we recorded the energy used, as described in Section 3.2, and divided by the actual distance traveled, as recorded by the GPS. Each test was divided into individual laps around the

Table 3.3: Average battery power draw for different speeds

Speed	0.75 m/s	1 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Straight	76.9 W	87.1 W	102 W	115 W	127 W
Turn	82.3 W	93.6 W	104 W	117 W	126 W

(a) Soft grass

Speed	0.75 m/s	1 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Straight	62.1 W	70.7 W	76.2 W	91.4 W	98.2 W
Turn	64.9 W	74.8 W	79.5 W	95.7 W	101 W

(b) Asphalt

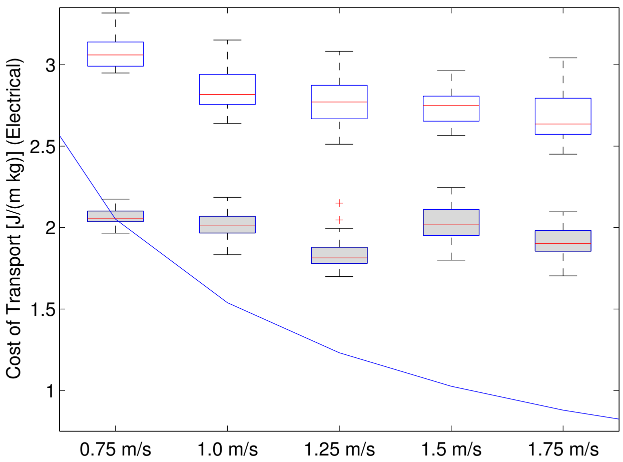


Figure 3.9: Statistical analysis of electrical cost of transport for Packbot (white- soft grass, gray- asphalt) compared with peripheral electronics (blue line)

Table 3.4: Variance and mean for individual laps (J/m)

	0.75 m/s	1.0 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Mean	61.45	57.09	55.45	54.88	53.47
Variance	3.85	6.90	5.90	3.50	9.05

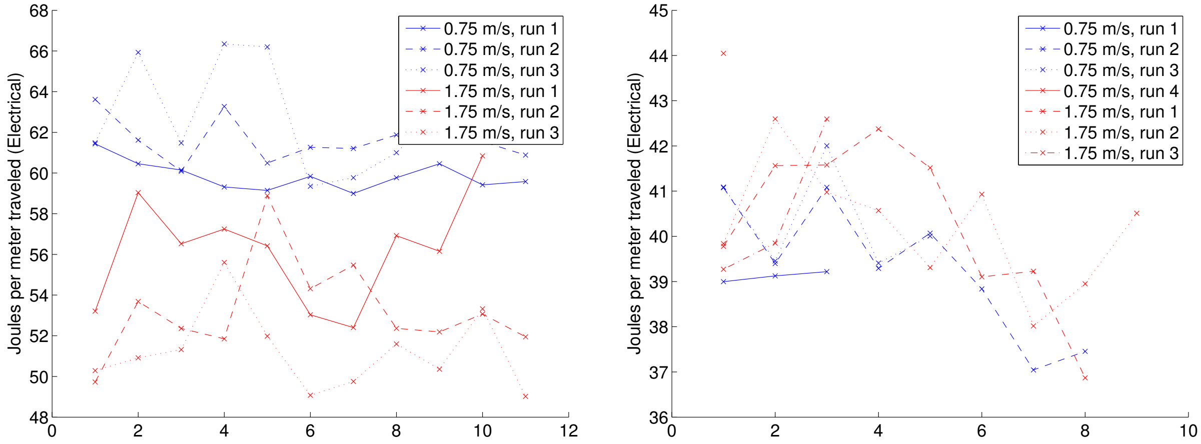
(a) Soft grass

	0.75 m/s	1.0 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Mean	41.36	40.24	36.74	40.61	38.14
Variance	1.20	2.23	2.93	4.19	3.68

(b) Asphalt

rounded box trajectory based on the GPS position. Figure 3.9 shows the cost of transport for the different tests. As described in Section 3.1, the electrical energy per distance traveled is divided by the mass of the Packbot, in this case 20 kg, to produce the cost of transport. The boxes show the spread of the electrical energy used for locomotion; the red center lines show the median value, the boxes show the 25%-75% percentile of the data and the whiskers show the limit of the data not considered outliers, with outliers individually plotted. The continuous blue line shows the energy used per meter traveled of the peripheral electronics. The power draw is assumed to be constant for different speeds; dividing the power draw by the speed gives the desired value of energy per meter.

There are two main trends in the data. The tests on grass show average cost of transport decreases as speed increases. For the asphalt cases, the cost of transport is significantly lower and is roughly constant over the speeds tested. Looking at Figure 3.9, we can compare the distribution



(a) Soft grass, laps

(b) Hard grass, 100 m segments

Figure 3.10: Energy usage for consecutive 100 meter segments

of energy usage from the individual segments. On the soft grass, there is considerable overlap in the distributions, with the median value (red line) decreasing with increasing speed. The asphalt tests show a decreased variation in energy used, presumably due to the more uniform, harder surface. Table 3.4 shows the sample variance of mean for the different trajectories. Using the statistical variance, we see that the variance is smaller for the runs on asphalt, with the exception of 1.5 m/s. The variance can be influenced by many factors, including differences in terrain between runs.

Looking at consecutive laps around the rounded box, we see considerable variation in the energy use required. Figure 3.10a shows the electrical energy usage of subsequent laps for 0.75 and 1.75 m/s. Figure 3.10b shows the straight line tests broken into 100 m segments. These tests show less variability than the soft grass tests.

Comparing the energy used for peripheral electronics with the energy used for motion shows that, on asphalt, for 0.75 m/s, the energy usage is split approximately equally between locomotion and peripheral electronics. For 1.75 m/s on asphalt, the energy used for peripherals is about half the energy used for locomotion.

To compare the energy consumption differences between straight line travel and the rounded box trajectory, the results of the tests on hard grass are shown in Table 3.5. For 0.75 m/s, the

Table 3.5: Energy per meter traveled of straight-line and rounded-box tests (J/m)

	0.75 m/s	1.75 m/s
Straight line (Hard Grass)	39.46	40.60
Rounded box (Hard Grass)	42.02	44.45
Rounded Box (Asphalt)	41.36	38.14

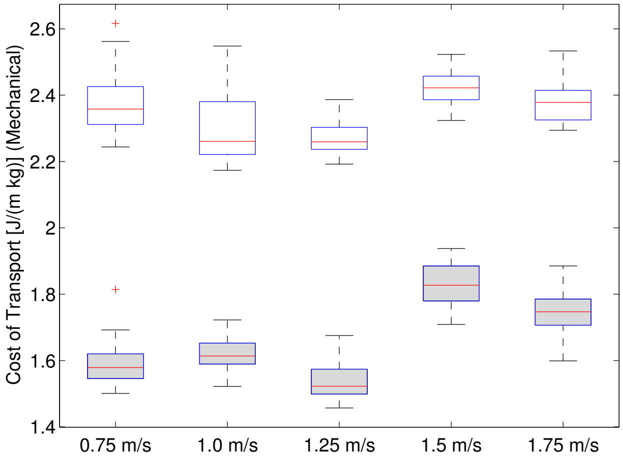


Figure 3.11: Scaled mechanical cost of transport for Packbot in different conditions(white- soft grass, gray- asphalt)

rounded box tests used 6.3% more energy than the average of the straight traversals down the airport runway and 9.6% more for 1.75 m/s. These tests show that the rounded box is a good estimate for straight line travel on the hard grass. These values are similar to the values obtained for the asphalt tests, rather than the softer grass, showing a strong dependence on soil conditions in the energy required for travel.

In addition to the electrical energy usage, we looked at the mechanical energy used. Figure 3.11 shows the scaled mechanical cost of transport for the same tests presented in Figure 3.9. The trends present in the electrical cost of transport are not present in these results; the tests on soft grass stayed relatively constant while the energy usage for the asphalt tests generally increased with speed.

Since the mechanical energy used is closely related to the motion resistance (assuming a constant normal load), we expected to see the same trends summarized in Section 2.3.2 for soft grass; however, the energy used does not decrease with speed. There are many possibilities for this discrepancy; however, due to the use of a fielded robot and not a dedicated test setup, we are unable to conclusively analyze the track-ground interactions of the Packbot during use. We leave as future work further experimental analysis of the track-ground interactions for the Packbot.

On both asphalt and the hard grass surfaces, the scaled mechanical energy use increases with speed. In particular, Table 3.6 shows the comparison of the straight line and rounded box trajectories for hard grass. In both cases, the energy usage increases, although the increase is greater for the rounded box trajectory. For the hard grass scenario, this implies that the rounded box does increase the resistance, though it is not the only factor involved.

Table 3.6: Scaled mechanical energy per meter of straight-line and rounded-box tests (J/(Km))

	0.75 m/s	1.75 m/s
Straight line (Hard Grass)	31.09	38.52
Rounded box (Hard Grass)	31.91	42.34
Rounded Box (Asphalt)	31.76	34.93

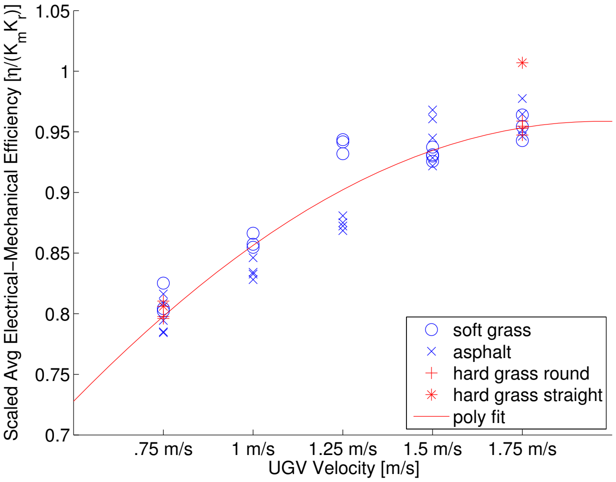


Figure 3.12: Efficiency values consistent over the different runs

3.3.2 Electrical-to-Mechanical Efficiency

Power loss in the motors and drive train is one source of loss in the system. Using the electrical and scaled mechanical energy, discussed in the previous section, we can calculate the efficiency of the conversion from electrical to mechanical energy. Since we are dealing with a "black box" model of the Packbot, we must make the assumption that the gear ratio from the track drive wheels to the motors is constant. Based on this assumption we can parameterize the efficiency based on the track speeds.

To compute efficiency we define

$$\text{efficiency} = \eta = \frac{\text{Power Out}}{\text{Power In}} = \frac{K_m(I_l\omega_l + I_r\omega_r)}{\text{Battery Power}} = \frac{K_m K_r\left(I_l\frac{v-\omega B}{R} + I_r\frac{v+\omega B}{R}\right)}{\text{Battery Power}}. \tag{3.4}$$

Since K_m and K_r are unknown, we calculate the scaled efficiency, $\eta/(K_m K_r)$. In addition to regularizing the data, we excluded some data points where, due to experimental conditions, the battery power input dropped down to near 0, resulting in large efficiency values that could skew the final results.

Figure 3.12 shows the scaled efficiencies from the different tests. Efficiency varies with vehicle speed and the efficiency values are consistent at the same speed on different runs, including on different surfaces. This figure also corresponds closely to measured efficiency curves for ground robots (see [93] for an example). Also, interestingly, scaled efficiency for the rounded box trajectory on hard grass at 1.75 m/s is significantly higher than for other surfaces, possibly due to the low state of battery charge for that test. Since DC motor efficiency varies with speed and torque [94, Chapter 10], it is interesting to note that there is little difference in efficiency between surfaces. Since higher speeds and lower torques are more efficient, we conclude that the speed increase is more important for efficiency than the increase in output torque.

Since our tests approach the maximum speed of our Packbot, we could not test higher speeds to see if the efficiency continues to increase or where the peak efficiency occurs. In fact, as described in [93], one method for motor selection and gear design is to put the maximum efficiency of the system at the largest power output. We expect the efficiency curve to have a peak, and we assume this peak to be around 2 m/s, the maximum speed of the robot, since we only observed efficiency increasing in the range of tested speeds. Above this speed, the efficiency decreases until the motor stall torque is reached and the motor outputs no work. To shape the portion of the curve that we could not test, we added three extra data points at higher speeds: 0.9 at 2.5 m/s, 0.8 at 3.5 m/s, and 0.7 at 4.5 m/s. Without including these points, the polynomial fit increased rapidly above 1.75 m/s, an infeasible result. We used a fourth order polynomial fit, which resulted in the following equation:

$$\begin{array}{l}\text{scaled}\\ \text{efficiency}\end{array} = \frac{\eta}{K_m K_r} = .0055v^4 - .033v^3 - .036v^2 + .3586v + .5613 \tag{3.5}$$

This curve partially explains the results presented earlier; the tests at a higher speed are operating in a more efficient regime of operation and have lower power loss in the motors and drivetrain. Path planners can use this curve to plan efficient operation.

3.3.3 Range Prediction

Using the methodology proposed in Section 3.1, we have characterized energy usage for the Packbot. As one example usage of the data, we present simple range estimates based on the traveling speed. In this case, we will assume a constant speed for the entire duration of the mission.

Using the data from Figure 3.9 and taking into account the average power for peripherals, we can estimate the range of the Packbot traveling at a constant forward velocity v using the following formula:

$$\text{range} = \frac{E_{batt}}{E_{dist} + P_{per}/v}\,\frac{[J]}{[J/m]} \tag{3.6}$$

where E_{batt} is the available battery energy at the start of the traversal (J), E_{dist} is the average energy expended per unit distance from Table 3.4 (J/m) and P_{per} is the average power for peripherals (W).

Table 3.7: Estimated Packbot range at constant speed from full battery [km]

	0.75 m/s	1 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Grass	3.39 ± 0.06	3.94 ± 0.12	4.33 ± 0.13	4.60 ± 0.11	4.90 ± 0.11
Asphalt	4.21 ± 0.06	4.88 ± 0.10	5.66 ± 0.16	5.68 ± 0.19	6.22 ± 0.19

Table 3.8: Total distance and percentage of battery depleted for the tests

	0.75 m/s	1.0 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Distance [km]	3.17	3.12	3.10	4.11	3.05
Battery Discharge [%]	91.55	77.96	70.94	88.53	62.09

(a) Grass

	0.75 m/s	1.0 m/s	1.25 m/s	1.5 m/s	1.75 m/s
Distance [km]	4.08	4.16	4.08	5.11	4.51
Battery Discharge [%]	94.60	83.59	71.50	89.20	71.95

(b) Asphalt

Table 3.7 lists the estimated range for a fully charged battery for the Packbot, with error values based on the variance of energy usage. The accuracy of these estimates depends on many factors: the variability of the terrain, the accuracy of battery charge estimate, and the variance of the energy usage data. Table 3.8 shows the range from the tests presented in the previous section along with the percentage of the battery used. All the tests for a particular velocity were run on the same battery with minimal breaks between tests, although the tests weren't run until the battery was depleted.

One interesting aspect of the error values in Table 3.7 is the increase of the error range as speed increases, even though the variance of the energy usage data varies with speed. This is due to the relative size of P_{per}/v and E_{dist}. Table 3.9 compares the values of E_{dist} and P_{per}/v from our tests. When v is larger, the load from electronics is less per unit distance, so the variance of E_{dist} has a larger effect. In addition, modifying the setup to change P_{per} would not result in a uniform effect for all speeds; slower speeds would have a larger range increase when P_{per} decreases than faster speeds.

Table 3.9: Comparing E_{dist} and P_{per}/v

v	E_{dist} (soft grass)	E_{dist} (asphalt)	P_{per}/v
0.75 m/s	61.45	41.36	41.04
1.25 m/s	55.45	36.74	24.62
1.75 m/s	53.47	38.14	17.59

This range estimate can be used to augment battery range estimation algorithms to improve accuracy. This is a simple range estimation; research into estimating range based on robot history and estimated future action is ongoing [28]. Battery models under load also can affect the range and thus the results of these projections (see [95] for an example of a battery model).

3.4 Contributions

We have presented a methodology for characterizing energy usage for a mobile ground robot (Innovation 1) and results obtained by applying this methodology to the iRobot Packbot. This methodology entails running the robot through a series of tests at different speeds and on different terrains. From these tests, we record the energy used per distance traveled and the electrical-to-mechanical efficiency. This data can be used to produce more efficient path plans and better estimate remaining range.

For the Packbot, our methodology showed that the robot required less energy when traveling on asphalt than on soft grass. On the soft grass, the energy requirements decreased with increasing speed, while the same trend was not present on asphalt. The electrical-to-mechanical efficiency increased with speed up to the maximum speed allowed by the Packbot, but did not depend on the terrain. This entire set of data constitutes Contribution 1.

Data from these experiments can be used to improve battery life prediction, and better inform autonomous planning and operation protocols for small robotic vehicles. If the energy per distance at different speeds is known for the terrains to be traveled upon, a range prediction can be generated. Alternatively, an energy efficiency curve can be used in conjunction with a more detailed terrain interaction model to estimate the range for a given battery state of charge. Models for energy usage with more complicated maneuvers and/or more variable terrain is left for future research.

CHAPTER 4

Area Coverage Trajectory Planning

Area coverage is a simple mission that can be accomplished by a UGV. For some missions such as search and rescue or reconnaissance, the robot needs to cover the area with a sensor. For a cleaning or mowing application, some physical device must be swept across the entire region. For either application, the basic problem is the same: find a path where the robot passes near every point in the region. Assuming a perfect detection rate, one pass of the sensor is sufficient to consider an area covered. Depending on the location of the search, the area can be filled with various obstacles and regions that are unsearchable.

Previous research has focused on algorithms to generate paths that cover an area, assuming that the vehicle is capable of traversing the path without time or energy limitations [40]. We want to be able to increase the capability of a UGV to cover a region by proposing an optimal formulation to conserve energy. Using these optimal formulations, the operator can choose between a trajectory that covers the area quickly while using more energy or takes longer due to conservative operation, increasing the area that can be covered in a single mission.

We formalize the problem of combining area coverage with limited energy as follows: suppose a robot executing a trajectory P through an area A covers $C \subset A$ and consumes energy E, which is dependent on many factors, including speed, acceleration rates and motor efficiencies. The trajectory consists of the continuous states of the robot, e.g. location, heading, and velocity, over time. A point $s \in A$ is covered if $\exists p \in P : s \in S_p$, where S_p is the region that the UGV can cover at the state p. We want to find the trajectory P that minimizes a weighted sum of UC and E, where UC is the uncovered area $A \backslash C$.

Our solution to this problem consists of the following steps:

1. Plan a coverage path using algorithms from the literature, treating the path as a series of waypoints.

2. Define a cost function combining energy, coverage, and motor efficiency.

3. Solve the optimal control problem for the cost function with the restriction that the UGV must visit all the waypoints. Two options for turns are possible:
   (a) UGV executes turns in place
   (b) UGV moves through turns
4. Use trajectory tracking control to drive the UGV along the optimal trajectory

In this chapter, we present Innovation 2: the novel cost function used in step 2 that incorporated energy, coverage and motor efficiency. The optimal velocity profile is found by applying optimal control theory to the cost function. Inherent in the cost function are tradeoffs between the different terms by varying weighting parameters. As Contribution 2 of this dissertation, we show that our mathematical model predicts that driving the UGV more slowly can save energy over the length of the coverage maneuver. Using these tradeoffs between time and energy, final values for a particular maneuver can be selected based on relative availability of time and energy. Experimental results for a subset of the tradeoffs are presented.

We review background material on energy usage in Section 2.3 and related work on area coverage in Section 2.4. In particular, we use the Boustrophedon coverage planner in this chapter. The mission studied in this chapter is shown in Figure 4.1. In Section 4.2, we present a novel cost function for the coverage problem and create trajectories for the coverage paths. Trends created by varying weighting parameters are presented in Section 4.3. Experimental results are presented in Section 4.4. These results were initially published in [8, 9].

4.1 UGV Model

In this chapter, we consider a tracked vehicle operating in a known environment. This model was first introduced in Section 2.2 and is repeated here. The system state $\mathbf{x}$ can be described by the robot's location x_R and y_R, velocity v, heading θ, and angular velocity ω. Mathematically, this is described as $\mathbf{x} = [x_R, y_R, v, \theta, \omega]^T$. Taking our inputs as u_l and u_r, the forces exerted by the left and right tracks respectively, we can describe the system dynamics as (adapted from [15])

$$\dot{\mathbf{x}} = f(\mathbf{x}) = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \frac{u_r+u_l}{M} - bv \\ \omega \\ \frac{B(u_r-u_l)}{I_z} - b_r\omega \end{bmatrix}. \tag{4.1}$$

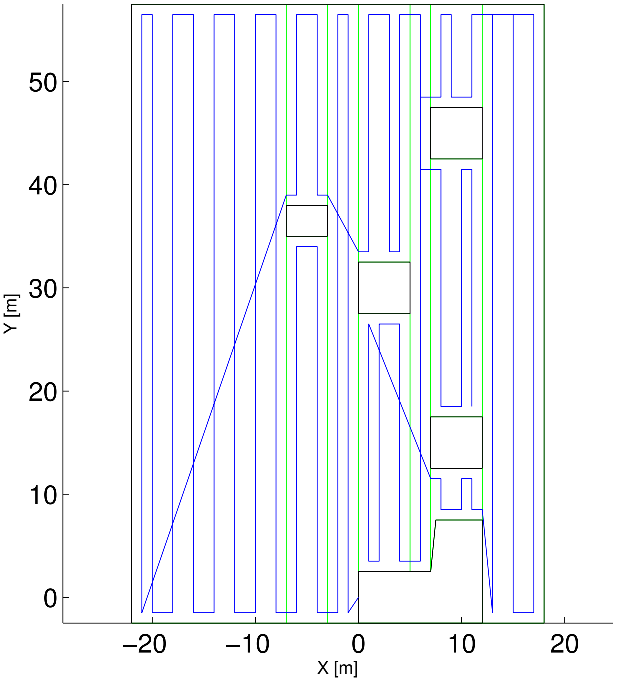


Figure 4.1: Path (blue) generated by BD algorithm with cells marked (green)

where M is the vehicle mass, bv is the linear loss due to drag and friction, B is the length from the track to the center of the vehicle, I_z is the moment of inertia, and $b_r\omega$ is the turning loss due to friction and drag. We assume the magnitude of each of input $u_{r,l}$ is bounded by u^*_{max} and the velocity is bounded by v_{max}. In our simulation model, we used the parameters based on the iRobot Packbot, which is also used for experiments.

4.2 Optimal Trajectories

To quantify a tradeoff between coverage, time and energy usage, we introduce the following cost function:

$$J = \int_{t_0}^{t_f} K_0(u_r^2 + u_l^2) + K_1 \text{meas}(\Psi(t)) + K_2 \text{E}(\omega_{motor})\, \text{d}t. \tag{4.2}$$

Each of the terms in the cost function is used to penalize either coverage, time, or energy usage. The first term $u_r^2 + u_l^2$ penalizes the forces used to accelerate the UGV. This penalizes energy usage through limiting forces applied by the UGV and indirectly influences the final time by limiting acceleration and deceleration, resulting in longer traversals.

The function $\text{meas}(\Psi(t))$ represents the area that remains to be covered. This function depends on all previous positions of the UGV, not just the current state $\mathbf{x}(t)$, so $\Psi(t)$ represents the set of all states on the interval $[t_0, t]$. A point p in the region is covered if, at some time $\tau \in [t_0, t]$,

$|p - x(\tau)| < R$, where R is the search radius of the UGV. In this dissertation, we consider the case $R = 1\ \mathrm{m}$. This directly penalizes coverage, as well as time required to cover the designated region since faster trajectories will have $\mathrm{meas}(\Psi(t))$ functions that decrease more quickly.

The function $\mathrm{E}(\omega_{motor})$ penalizes energy wasted by using the UGV motors in inefficient regions of operation. For this application we use

$$\mathrm{E}(\omega_{motor}) = \mathrm{eff}_{max} - \frac{\mathrm{eff}(\omega^l_{motor}) + \mathrm{eff}(\omega^r_{motor})}{2} \tag{4.3}$$

with the superscripts denoting the left and right motors respectively and eff_{max} denoting the maximum efficiency of the motor and $\mathrm{eff}(\omega_{motor})$ denoting a known motor efficiency curve for the UGV, parametrized by motor speed. As described in Section 3.3.2, we have obtained experimental efficiency data for an iRobot Packbot. We chose to parameterize by vehicle speed based on the data available from the Packbot in our experiments. Since there is no transmission, there is a constant relationship between vehicle speed and motor speed.

Using only the above cost function, without constraining the position of the UGV, the optimal action is to say the mission is completed when it begins ($t_f = t_0$). There are two possible solutions to change the optimal action: penalize the value of $\mathrm{meas}(\Psi(t_f))$ or constrain the UGV to follow a path that produces a coverage maneuver.

Penalizing the final value of $\mathrm{meas}(\Psi(t))$ would allow for an optimal derivation of the path in addition to a trajectory along that path. The path would be free to take sharp or easy turns and mold to the contours of the obstacles and the region. However, there would be a large dependence on the weight of $\mathrm{meas}(\Psi(t_f))$. To achieve a complete coverage, the weight would have to be infinite. With any finite weight, $\mathrm{meas}(\Psi(t_f))$ would depend on the size of the region and the placement of obstacles, making a complete analysis difficult.

By constraining the UGV to follow a path, the fraction uncovered at each position is known before creating the optimal trajectory. The path, however, is not generated considering the cost function. The paths described in Section 2.4 assumes that turns are completed in place, a reasonable assumption, but requiring the UGV to stop and then accelerate can be a waste of energy.

The second approach is taken in this dissertation: we use the paths described in Section 2.4 to constrain the motion of the UGV to cover the region. The path is divided into individual segments, either straight line motion or turning in place. An optimal trajectory is generated for each individual segment and the trajectories are concatenated together to form an complete trajectory. The optimal trajectory is found using the method described in Section 2.7. The optimal trajectories are numerically solved using the `bvp4c` boundary value problem solver in MATLAB.

4.2.1 Straight line segments

Since traveling along a segment represents one dimensional motion, we can simplify the model introduced in Section 4.1 for determining the optimal trajectory. We can describe the system dynamics as $\ddot{s} = u/M - b\dot{s}$, where s is the distance traveled and $u = u_l + u_r$, with $u_{max} = 2u^*_{max}$. Since the UGV is traveling straight, $u_l = u_r$. In addition, we need to add a constraint to the problem to enforce the maximum velocity. We decided to add a penalty in the cost function for exceeding the maximum velocity, using the term $\tilde{v} = \dot{s} - v_{max}$. Framing the problem this way does not force that the solution exactly follows the v_{max} constraint, but by increasing the penalty gain, the solution is arbitrarily close. The cost function for the straight segment can be written as

$$\begin{aligned} J_s = \int_{t_0}^{t_f} & \frac{1}{2} K_0 u^2 + K_1 \operatorname{meas}(\Psi(t)) + K_2 \mathrm{E}(\omega_{motor}) \\ & + P(\tilde{v})^2 H(\tilde{v}) \, \mathrm{d}t. \end{aligned} \tag{4.4}$$

Note that the coefficient on the u term is $1/2$. In this simplified model, u is the sum of the inputs in the original model presented in Section 4.1; changing the coefficient preserves the same penalty weight on the input. The fourth term penalizes v exceeding v_{max}, where H denotes the unit step function. Choosing $P = 5000$ resulted in a 1% overshoot of v_{max} using parameters for the Packbot, acceptable for our application. In this case, solving for $\mathrm{E}(\omega_{motor})$ is simple since our efficiency function is parametrized by the forward speed $\dot{s}$.

Following the method presented by [74], we augment the cost function with the costate equations $\mathbf{p} = [p_1, p_2]^T$ and derive the Hamiltonian and conditions for optimality:

$$\begin{aligned} \mathscr{H} = & \frac{1}{2} K_0 u^2 + K_1 \cdot \operatorname{meas}(\Psi(t)) + K_2 \mathrm{E}(\omega_{motor}) \\ & + p_1 \dot{s} + p_2 (u/M - b\dot{s}) + P\tilde{v}^2 H(\tilde{v}) \end{aligned} \tag{4.5}$$

$$\dot{\mathbf{p}} = \begin{bmatrix} -K_1 \frac{\partial \operatorname{meas}(\Psi(t))}{\partial s} \\ -p_1 + bp_2 - K_2 \frac{\partial \mathrm{E}}{\partial \dot{s}} - 2P\tilde{v}H(\tilde{v}) \end{bmatrix} \tag{4.6}$$

$$u = \begin{cases} \frac{-p_2}{MK_0} & \left| \frac{-p_2}{MK_0} \right| \leq u_{max} \\ -u_{max} & \frac{-p_2}{MK_0} < -u_{max} \\ u_{max} & \frac{-p_2}{MK_0} > u_{max} \end{cases} \tag{4.7}$$

with a free end time condition of

$$\begin{aligned} 0 = & \frac{1}{2} K_0 u^2 + K_1 \operatorname{meas}(\Psi(t_f)) + K_2 \mathrm{E}(\omega_{motor}) \\ & + p_1 \dot{s} + p_2 (u/M - b\dot{s}) + P\tilde{v}^2 H(\tilde{v}). \end{aligned} \tag{4.8}$$

Since our paths are parallel for the most part, we can write $\mathrm{meas}(\Psi(t)) = \mathrm{meas}_0 - 2sOR/A$, where meas_0 is the value of $\mathrm{meas}(\Psi(t))$ when the UGV begins to traverse the segment and O represents the fraction of the path that does not overlap with a previous segment. For most of the segments, $O = 1$; however on some segments next to the edges of the cells the value can be lower. For segments that cross over previously-covered area, we assume $O = 0$ over the length of the path.

This solution is similar to the minimum time-energy optimization problem described in [74], with $\mathrm{meas}(\Psi(t))$ playing a similar role to the time weighting. The one difference is the time weighting term, for this application $\mathrm{meas}(\Psi(t))$, decreases as the UGV moves. Since the area covered in each segment of the path is generally much smaller than the total search area, $\mathrm{meas}(\Psi(t))$ changes very little along a segment.

4.2.2 Turn in place segments

Since turning in place is also a one dimensional motion, we can simplify the dynamics to $\ddot{\theta} = \tau/I_z - b_r\dot{\theta}$, where $\tau = B(u_r - u_l)$, $\tau_{max} = 2Bu^*_{max}$ and $u_l = -u_r$. The optimal control derivation is simpler because $\mathrm{meas}(\Psi(t))$ is a constant while the UGV turns in place since no new area is covered. We can write the cost function for the turn in place segment as

$$J_t = \int_{t_0}^{t_f} \frac{1}{2} K_0 \left(\frac{\tau}{B}\right)^2 + K_1 \mathrm{meas}\left(\Psi_0\right) + K_2 \mathrm{E}(\omega_{motor})\, \mathrm{d}t \tag{4.9}$$

In this case, we assume that the UGV does not exceed the maximum turn rate for in-place heading changes and have not included a penalty term in the cost function. Also, to calculate $\mathrm{E}(\omega_{motor})$, we must convert from $\dot{\theta}$ to $v_{wheel} = B\dot{\theta}$, which we can use to determine the efficiency of the motors.

Using the same method as before, we find the optimal control as

$$\begin{aligned} \mathscr{H} = \frac{1}{2} K_0 \left(\frac{\tau}{B}\right)^2 &+ K_1 \mathrm{meas}\left(\Psi_0\right) + K_2 \mathrm{E}(\omega_{motor}) \\ &+ p_1\dot{\theta} + p_2(\tau/I_z - b_r\dot{\theta}) \end{aligned} \tag{4.10}$$

$$\dot{\mathbf{p}} = \begin{bmatrix} 0 \\ -p_1 + b_r p_2 - K_2 \frac{\partial}{\partial} \end{bmatrix} \tag{4.11}$$

$$\tau = \begin{cases} \frac{-B^2 p_2}{I_z K_0} & \left|\frac{-B^2 p_2}{I_z K_0}\right| \leq \tau_{max} \\ -\tau_{max} & \frac{-B^2 p_2}{I_z K_0} < -\tau_{max} \\ \tau_{max} & \frac{-B^2 p_2}{I_z K_0} > \tau_{max} \end{cases} \tag{4.12}$$

with a final time constraint of

$$\begin{aligned} 0 = \ & \frac{1}{2} K_0 \left(\frac{\tau}{B}\right)^2 + K_1 \text{meas}\left(\Psi_0\right) + K_2 \text{E}(\omega_{motor}) \\ & + p_1 \theta + p_2 (\tau / I_z - b_r \dot{\theta}). \end{aligned} \tag{4.13}$$

4.2.3 Moving Turns

Since coming to a complete stop is inefficient for a UGV in terms of energy usage, one heuristic to improve energy conservation is to have the UGV move through a turn instead of turning in place. The difficulty in making such a turn lies in the fact that we don't want to miss covering some area, although in future work the turning path itself could be optimized to miss some area while minimizing energy use.

To guarantee coverage, the moving turn begins at the waypoint. The UGV returns to the straight line path between the waypoints at one search radius away from the originating waypoint. The straight segment trajectory between turns is generated using the same method described in Section 4.2.1 with non-zero initial and final velocities specified.

Since the trajectory for this turn includes both turning and forward motion, we must use the full-state model given in Section 4.1. For simplicity, we assume that $\text{meas}(\Psi(t))$ is constant during the turn. Using the same optimization method, we derive the conditions for optimality:

$$\begin{aligned} \mathscr{H} = \ & K_0 \left(u_r^2 + u_l^2\right) + K_1 \text{meas}(\Psi_0) \\ & + K_2 \text{E}(\omega_{motor}) + \mathbf{p}^T f(x) \end{aligned} \tag{4.14}$$

$$\begin{aligned} \mathbf{p}^T f(\mathbf{x}) = \ & p_1 v \cos\theta + p_2 v \sin\theta + p_3 \left(\frac{u_l + u_r}{M} - bv\right) \\ & + p_4 \omega + p_5 \left(\frac{B(u_r - u_l)}{I_z} - b_r \omega\right) \end{aligned} \tag{4.15}$$

$$\dot{\mathbf{p}} = - \begin{bmatrix} 0 \\ 0 \\ p_1 \cos\theta + p_2 \sin\theta - b p_3 - \left(\frac{\partial \text{E}}{\partial v} + \frac{\partial \text{E}}{\partial v}\right)/2 \\ -p_1 v \sin\theta + p_2 v \cos\theta \\ p_4 - b_r p_5 - \left(\frac{\partial \text{E}}{\partial v} - \frac{\partial \text{E}}{\partial v}\right)/2 \end{bmatrix} \tag{4.16}$$

$$\mathbf{0} = [p_3/M + Bp_5/I_Z + 2K_0 u_r, p_3/M - Bp_5/I_z + 2K_0 u_l], \tag{4.17}$$

$$\begin{aligned} & \Big[K_0 \left(u_l^2 + u_r^2\right) + K_1 \text{meas}(\Psi(t_0)) \\ & + K_2 \text{E}(\omega_{motor}) + \mathbf{p} f(x) \Big] \Big|_{t_f} = 0 \end{aligned} \tag{4.18}$$

We assume that over this short time span v does not exceed v_{max}, and did not penalize exceeding v_{max} in the cost function. Additionally, the values for u_l and u_r in Eqn. (4.17) are saturated to $\pm u^*_{max}$.

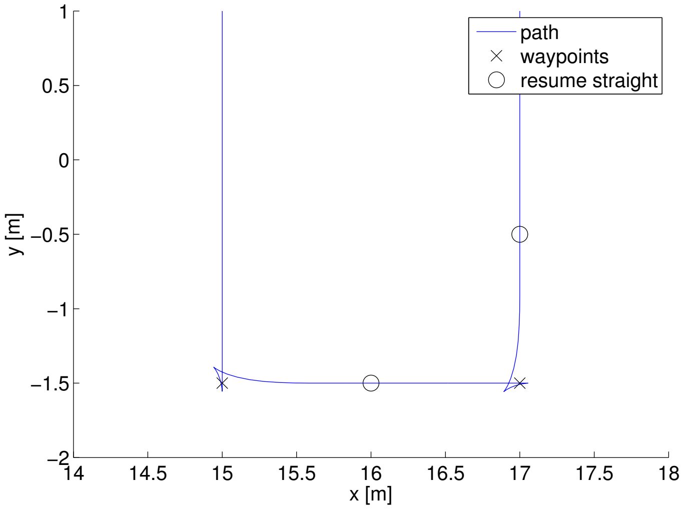


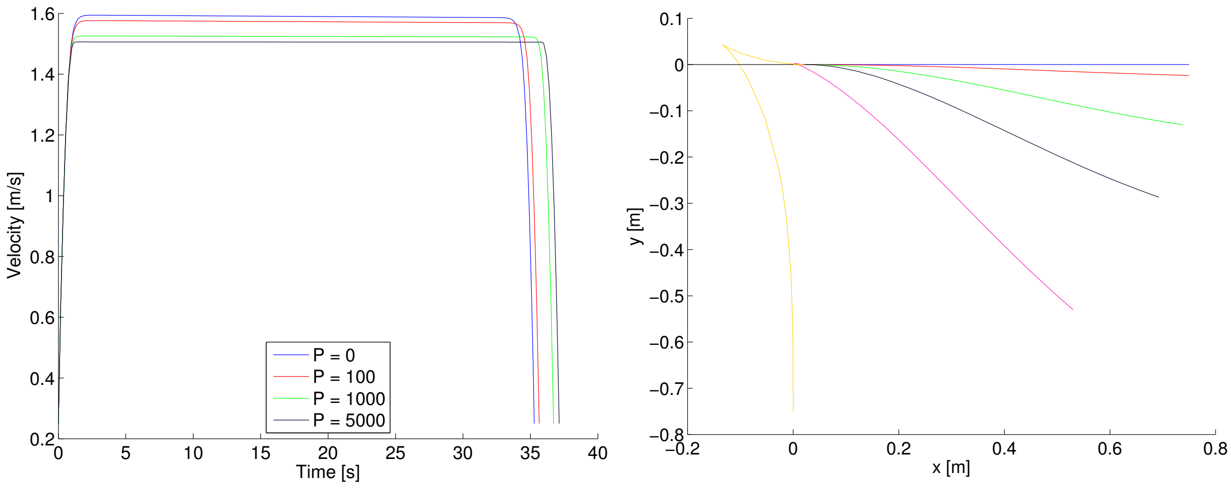
Figure 4.2: Turn modification in path

Figure 4.2 shows the turns in the context of the path. Note that the UGV stops and backs up before proceeding to the final point, resulting in peaks in the path. While a continuous turn is the most intuitive solution, in this case, it is infeasible due to the small turning radius. Some solutions showed a looping trajectory that greatly increases the time and distance traveled for the turn. Comparing the cost of this looping turn with the turns in Figure 4.2 showed that the paths shown here were optimal.

4.2.4 Implementation Details

The optimal solution was found using the `bvp4c` boundary value problem solver available in MATLAB. One of the limitations of this method is the solver requires an initial estimate of the solution and if the initial estimate is not sufficiently close to the optimal solution, the solver will fail to converge. Finding an initial guess for these equations was a complicated process. For the straight line case, the velocity penalty P, used to enforce the maximum velocity constraint on the robot, made the initial solution much harder to find. Instead, the optimization was first run with $P = 0$. The results of this optimization were used as an initial estimate for the case where $P = 100$. This incremental process was repeated, with higher values of P, until $P = 5000$, the desired value. Figure 4.3a shows the gradual stepping up of the constraints to enforce the maximum velocity.

Since this process takes a significant amount of time, a template solution, with an intermediate value of $\mathrm{meas}(\Psi(t))$, was saved and use as the initial guess for the different segments of the path. Since there is a large difference in path lengths, separate template solutions were used for short (less than 3 m), medium (3-35 m), and long (greater than 35 m) path lengths. A similar method is used for the turn equations as well. For the moving turn, the parameterization is in the angle of the turn. Initially, no turn is used (the robot travels straight forward) and the turn angle is stepped up until the desired turn is reached, as shown in Figure 4.3b.



(a) Velocity over straight segment

(b) Trace of moving turn

Figure 4.3: Stepping up constraints to solve the optimization problem

4.3 Tradeoffs in Time and Energy

Using the cost gains K_1 and K_2, we can tradeoff time to cover, input force required, and energy used. Without loss of generality, we will take $K_0 = 1$. We compare these trajectories to a minimum-time trajectory, formed by applying the maximum acceleration (or deceleration) possible at all times. We show the extra time taken, as well as the energy saved, for these tradeoffs.

In this section, we discuss the tradeoffs, generated in simulation, between time and energy in the following scenarios:

1. motor efficiency neglected in the cost function, setting $K_2 = 0$
2. motor efficiency peak velocity v_{eff} greater than maximum velocity, $v_{max} \leq v_{eff}$
3. motor efficiency peak velocity less than maximum velocity, $v_{eff} < v_{max}$.

The tradeoffs for the entire path are presented using total time to completion of the coverage path (T_f), the integral of the control effort $\int_0^{T_f} u_r^2 + u_l^2 \mathrm{d}t$, and the mechanical energy $\int_0^{T_f} \frac{\omega_{motor}\tau_{motor}}{\mathrm{E}(\omega_{motor})}\,\mathrm{d}t$. Additionally, comparisons of individual segments with different cost gains and in different portions of the path are made using the velocity profiles. The tradeoffs presented use the BD path presented in Figure 4.1.

The results presented here are for one specific region. Our experience shows that the trends are more pronounced in regions with many obstacles, resulting in many shorter segments.

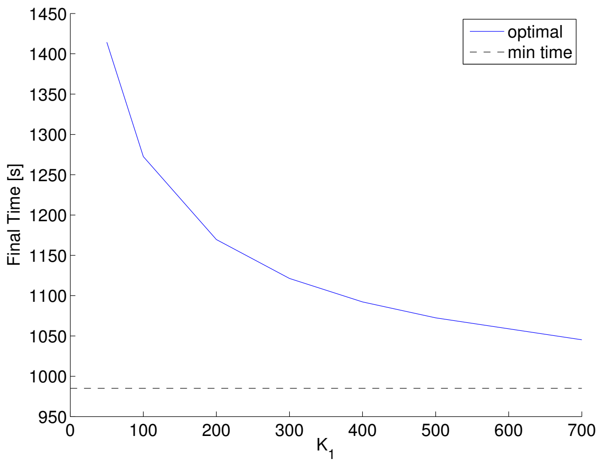


Figure 4.4: Final time decreasing with K_1 increasing ($K_2 = 0$)

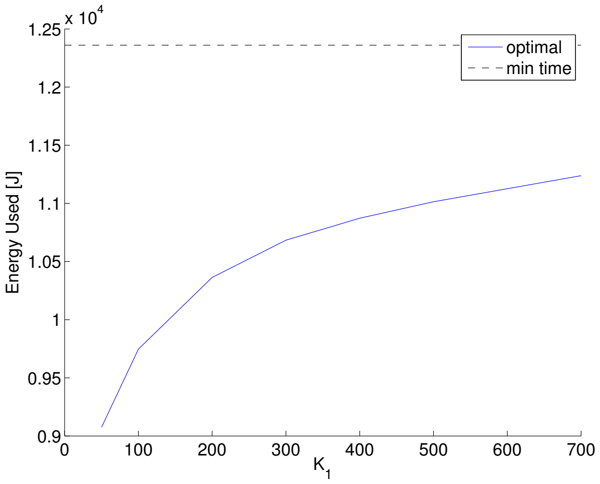


Figure 4.5: Estimated energy use increasing with K_1 ($K_2 = 0$)

4.3.1 Tradeoffs While Neglecting Motor Efficiency

For the first set of tradeoffs, we set $K_2 = 0$. These results were initially discussed in [9].

We selected values of K_1 ranging from 50 to 700 to determine the tradeoffs between the area covered, time, and energy required. As shown in Figure 4.4, the total time required to cover the area decreases as K_1 increases, approaching the minimum time value. In addition, the energy required increases with K_1 (Figure 4.5). The term $\int u_l^2 + u_r^2 \,\mathrm{d}t$ follows a similar trend to the energy. Table 4.1 calculates the predicted energy savings versus the time increase over the minimum time trajectory. For $K_1 = 50$, we predict a savings of 27% while increasing the time required by 44%. These values decrease as K_1 increases, though the time increase decreases at a faster rate. Using these values, a decision can be made for the desired value of K_1 to balance between time and energy used. Figure 4.6 compares the fraction of uncovered area versus the predicted energy used. Note that the lower values of K_1 use less energy to cover the same amount of area. The difference in energy consumed is larger towards the end of the trajectory, where decreasing values of $\mathrm{meas}(\Psi(t))$ result in slower trajectories.

Comparing the optimal segment traversals as the robot travels along the path yields an interesting observation. For the first segments, the larger fraction of uncovered area creates an optimal segment traversal with higher forces, resulting in faster accelerations. With each subsequent segment, the fraction of uncovered area decreases and the corresponding optimal trajectories have lower forces. The velocity profiles for individual segments for different $\mathrm{meas}(\Psi(t))$ values is shown in Figure 4.7. The largest differences occur when less than 5% of the area remains to be covered, in which case the velocity profile dramatically slows down to reduce energy consumption.

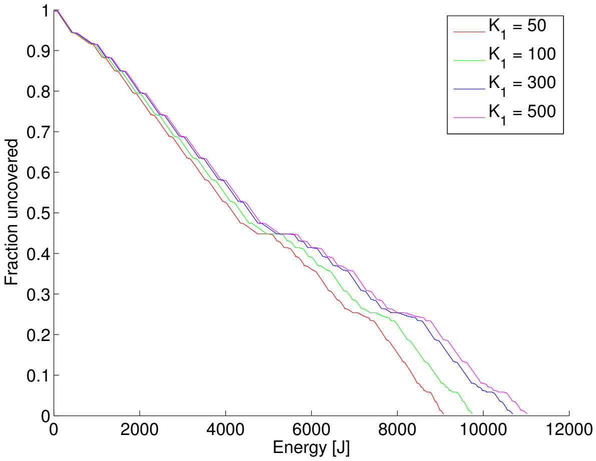


Figure 4.6: Energy versus $\mathrm{meas}(\Psi(t))$ for different runs

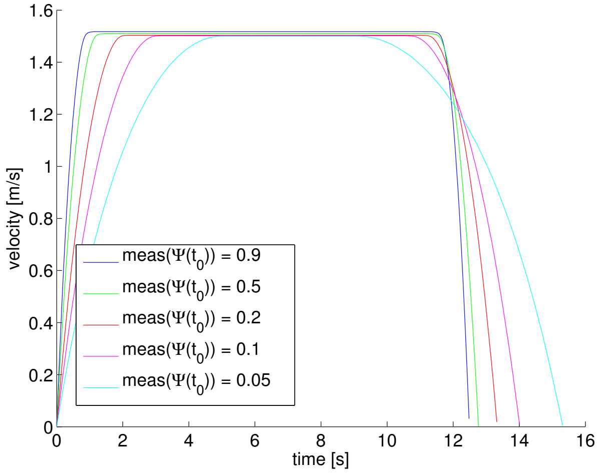


Figure 4.7: Later segments in the path (lower $\mathrm{meas}(\Psi(t_0))$) take longer to complete, ($K_2 = 0$, $K_1 = 300$)

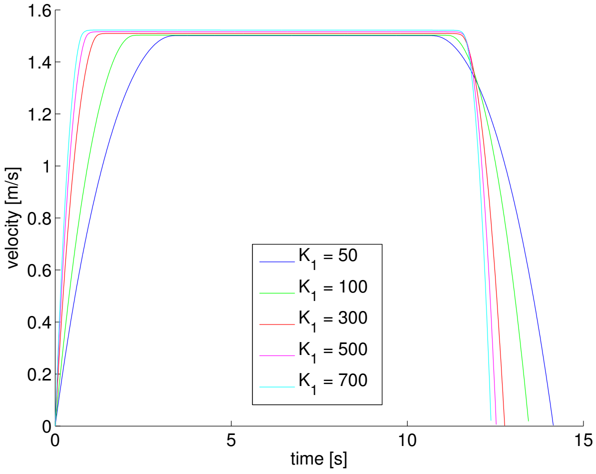


Figure 4.8: Smaller K_1 takes longer to reach max velocity and end of segment, ($K_2 = 0$, $\mathrm{meas}(\Psi(t_0)) = 0.5$)

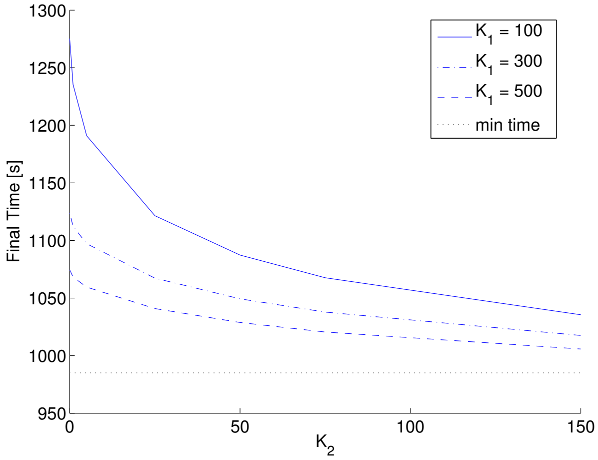


Figure 4.9: Final time decreases monotonically as K_1 and K_2 increase

In addition, varying K_1 also affects the velocity profile. Figure 4.8 shows how the velocity profile changes with K. In addition, for lower values of K_1, large differences mentioned above begin to occur at larger values of $\mathrm{meas}(\Psi(t))$.

4.3.2 Motor Efficiency Peak At Maximum Velocity

For this test, we remove the restriction that $K_2 = 0$. In this case, the time to completion of the path decreases with both K_1 and K_2, as shown in Figure 4.9. Similar to the previous case,

Table 4.1: Energy Saved and Time Increase ($K_2 = 0$)

K_1	50	100	200	300	400	500	700
Time Increase	44%	29 %	19%	14%	11%	9%	6.1%
Energy Saved	27%	21%	16%	14%	12%	11%	9.1%

Table 4.2: Energy Saved and Time Increase ($v_{max} < v_{eff}$)

K_1/K_2	100/0	100/5	100/75	300/5	300/75	500/75
Time Increase	29.5%	20.9%	8.4%	11.4%	5.4%	3.6%
Energy Saved	21.8%	19.7%	13.3%	13.4%	10.9%	9.4%

Figure 4.10 shows that the energy required increases with both K_1 and K_2. Note that the $K_2 = 0$ data points align with the results from the previous section. The decrease in time with K_1 is for the same reason as above. Table 4.2 shows that as K_2 increases, the time increase goes down as well as the energy savings. The time going down shows the major difference between these two trajectories: including K_2 changes the optimal trajectory so that the slow down over the path, shown in Figure 4.7, is much smaller, resulting in faster trajectories. The energy increases due to fact that the K_2 term attempts to minimize energy loss due to inefficiency, resulting in trajectories that, for short distances, accelerate quickly, then immediately decelerate and stop.

In this case, there is very little difference between trajectories in different sections of the path (Figure 4.11). This is due to the penalty on inefficient motor velocities, which effectively stabilizes the velocity profile near the most efficient velocity, regardless of the position along the path.

The trends in K_1 and K_2 are the same in this case because both terms have a common goal: drive at the maximum allowable velocity to cover new area or increase efficiency. The small differences in time to accelerate to v_{max} are due to a small tradeoff between $u_l^2 + u_r^2$ and the other cost terms; this tradeoff would be more visible at much lower values of K_1 and K_2.

When using a UGV with $v_{eff} \geq v_{max}$, the values of K_1 and K_2 can be set based on the relative availability of time and energy. These tradeoffs show that increasing both K_1 and K_2 will be better for time limited maneuvers and decreasing both weights for energy limited maneuvers.

Table 4.3: Energy Saved and Time Increase ($v_{max} > v_{eff}$)

K_1/K_2	100/5	100/50	100/150	300/150	500/150
Time Increase	25.9%	30.7%	42.8%	22.6%	14.7%
Energy Saved	21.4%	25.6%	31.6%	23.0%	18.3%

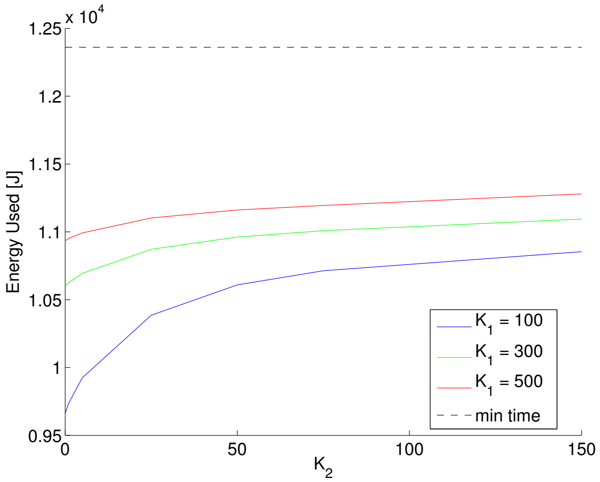


Figure 4.10: Energy usage increases with K_1 and K_2

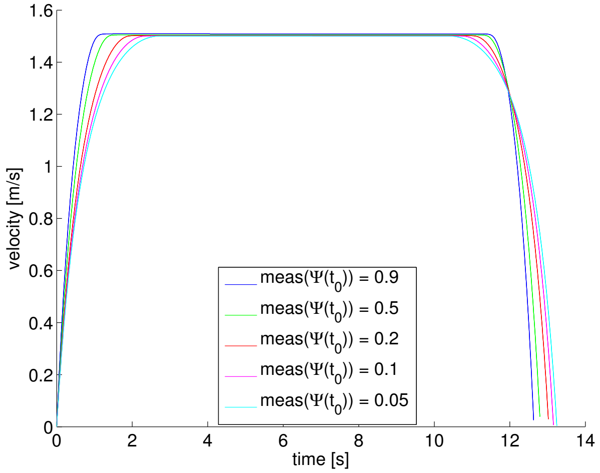


Figure 4.11: Small change in velocity profile for different segments in path ($K_1 = 150$, $K_2 = 75$)

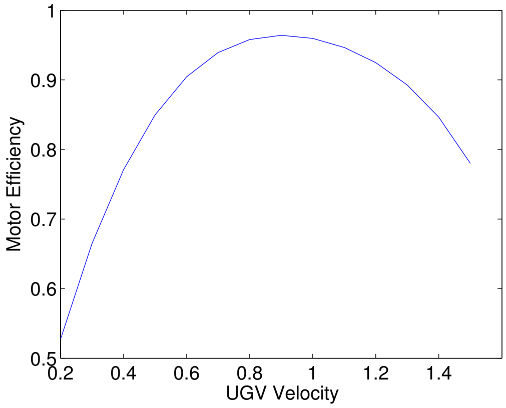


Figure 4.12: Motor efficiency curve used for $v_{eff} < v_{max}$

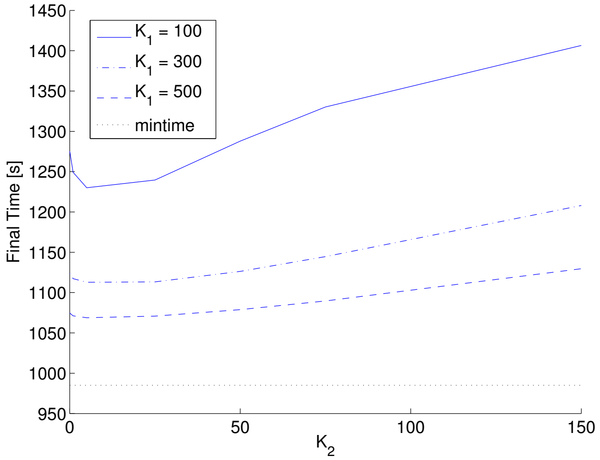


Figure 4.13: Multiple trends for final time with different values of K_1, K_2

4.3.3 Motor Efficiency Peak Below Maximum Velocity

Since using the parameters for the Packbot did not show any different tradeoffs than the case where $K_2 = 0$, we consider a system where the peak of the motor efficiency curve is below the maximum velocity for the system, shown in Figure 4.12 with all other system parameters unchanged.

Figure 4.13 shows a different trend from the previous cases in the final time; when K_2 is small, increasing K_2 decreases the final time. This is mostly due to a lack of slow down between segments in different portions of the path. As K_2 continues to increase, the final time starts to increase, due to the UGV driving slower to stay in the more efficient regions of the motors. The trend for K_1 is

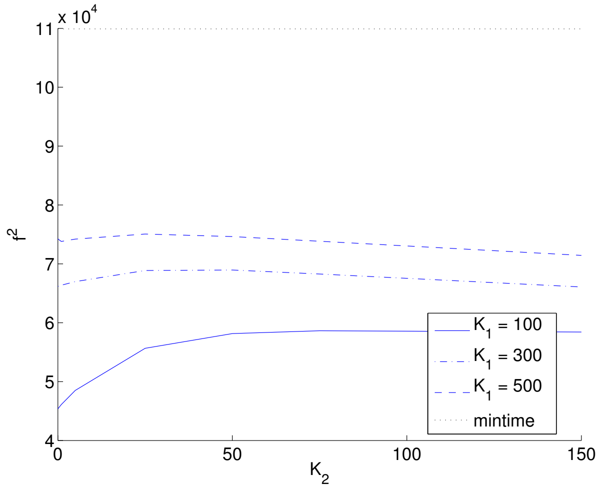


Figure 4.14: Multiple trends for $\int u_l^2 + u_r^2 \, dt$ with different values of K_1, K_2

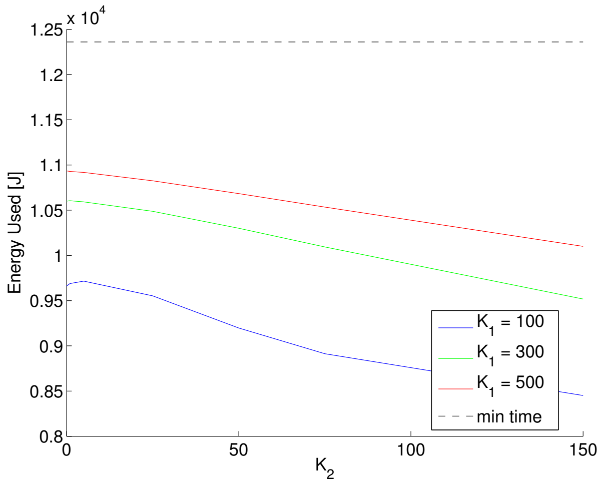


Figure 4.15: Energy usage decreasing as K_2 increases

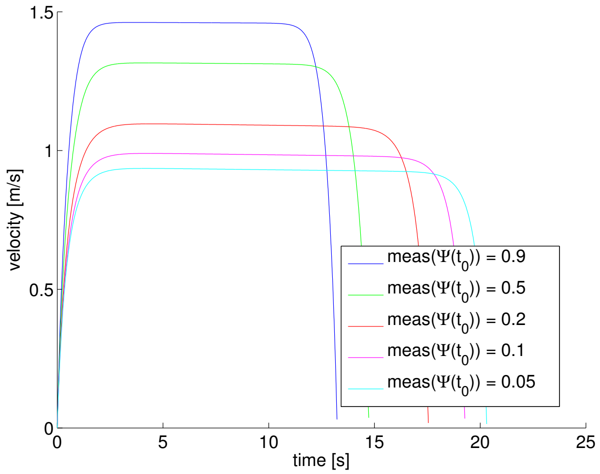


Figure 4.16: Velocity profile slower with increasing $\text{meas}(\Psi(t_0))$ ($K_1 = 150$, $K_2 = 75$)

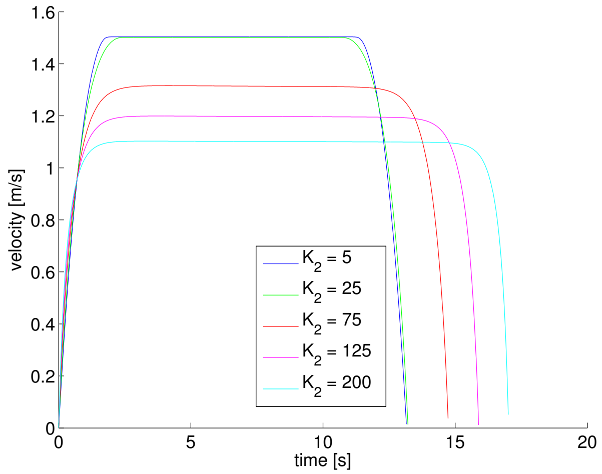


Figure 4.17: Velocity profile slows down when K_2 increases ($\text{meas}(\Psi(t_0)) = 0.5$, $K_1 = 150$)

the same as in previous cases; increasing this weight will only speed up the UGV to try and cover the region faster. The different trends based on the value of K_2 show that this is a 3-way tradeoff between force inputs, coverage, and efficient motor operation.

When comparing the control effort in Figure 4.14 used in these trajectories, there are different trends. It is interesting to note that in some places, by increasing K_2, the final time and control effort both increase, because more effort is exerted to quickly reach v_e, but then stays closer to v_e, slowing down the final time. However, the energy required, plotted in Figure 4.15, shows a different trend; with K_2 increasing, the motor stays in the efficient region and reduces the energy required for the maneuver.

Figure 4.16 shows the velocity profile of different segments along the path. Note that there is more variation early on in the path than in previous cases. The initial trend is similar to the first case, where as the area covered increases, the UGV slows down. The efficiency term in the cost function prevents the velocity from decreasing to low values.

Figure 4.17 shows the velocity profile for different values of K_2. For higher values of K_2, the trajectory more closely follows the most efficient velocity. If K_1 and K_2 are both scaled equally, the resulting plots would have steeper slopes at the beginning and the end but similar velocities in the middle of the segment.

When using a robot with $v_{\mathit{eff}} < v_{max}$, the relative weights of time and energy govern the optimal velocity profile. If there is an emergency scenario where time is the most important factor, K_2 should be small relative to K_1, allowing for a quick traversal of the area, potentially at the expense of missing part of the coverage area due to battery depletion. When a large area needs to be completely covered and time is not an issue, K_2 should be larger to allow for an energy-efficient coverage solution.

4.3.4 Effect of Moving Turn

To implement the moving turn, we must specify the incoming and outgoing velocities, reserving details on optimizing boundary value velocities for future work. Here, a constant value of $v_{enter} = v_{exit} = .25\mathrm{m/s}$ was used, based on our Packbot model. Substituting the moving turn for the turn in place yields a decrease in the time and energy required to complete the trajectory. The trajectory for the straight section is the same as in the previous trajectories.

Figure 4.18 shows the change in final time by adding the moving turn while neglecting motor efficiency. Note that the time difference is approximately constant over the values of K_1 shown. The energy savings, shown in Figure 4.19, is also fairly constant over the interval shown.

This is one possible heuristic to improve the energy usage by modifying the path. There are many other ways to change the path to improve energy usage, ranging from small path adjustments to different path planners optimized for various parameters.

4.4 Experiments

For experimental comparison, we drove an iRobot Packbot along the trajectories from Scenario 1, described in Section 4.3.1. The trajectories tested used the moving turn instead of the turn in place with $K_1 = 100, 300, 500$ and $K_2 = 0$. We used the moving turn because, with our controller, the Packbot was able to execute the moving turn more effectively.

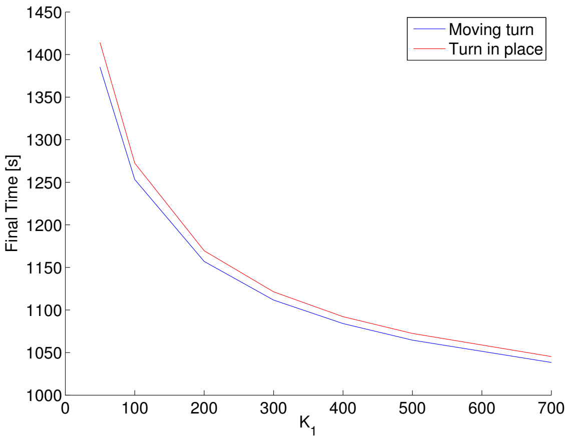


Figure 4.18: Moving turn decreases time required ($K_2 = 0$)

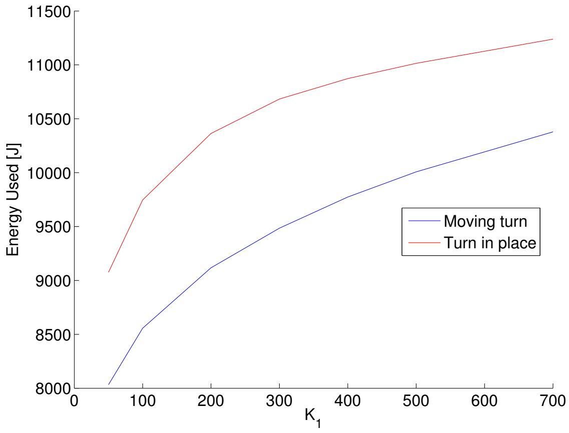
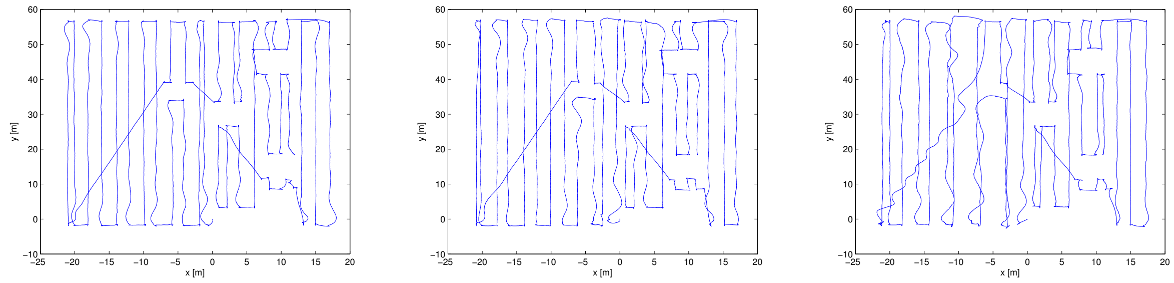


Figure 4.19: Moving turn decreases energy required ($K_2 = 0$)

These tests were run on a large, flat, grass field. We used a trajectory tracking controller presented in [92] to follow the reference trajectory. Each test was run with the battery beginning in the same state of charge. Figure 4.20 shows several different paths traversed by the Packbot. The Packbot is able to follow the path in general, with deviations from the reference path. For the run in Figure 4.20c, about 10% of the area was uncovered at the end, instead of the nominal .9%. Since this run is using the same trajectory as the run in Figure 4.20a, which ends with 3% uncovered at the end, these deviations are a function of external conditions, such as GPS noise.

During the run, we recorded at 20 Hz the robot location and battery power. From the location data, we were able to determine what fraction of the region had been covered. We discretized the field at .05 meter increments. At each time step, each previously unmarked cell within 1 meter of the Packbot was marked as visited. The fraction of unvisited area was computed as the number of unmarked cells divided by the number of total cells in the region, not including obstacles. We



(a) $K_1 = 100$, run 1

(b) $K_1 = 500$

(c) $K_1 = 100$, run 2

Figure 4.20: Executed paths by the Packbot, ($K_2 = 0$)

Table 4.4: Comparing Simulation and Experimental values ($K_2 = 0$)

		Simulation	
K_1	Final Time [s]	Area Uncovered	$\int u^2 dt$
100	1278.4	.9 %	10486
300	1148.4	.9 %	18906
500	1081.1	.9 %	24872
		Experimental	
	Final Time[s]	Area Uncovered	Energy [J]
100	1278.5	3.1%	100659
100	1278.6	10.4%	85090
300	1148.6	7.6%	83793
300	1148.6	5.6%	82612
500	1081.3	5.7%	80913

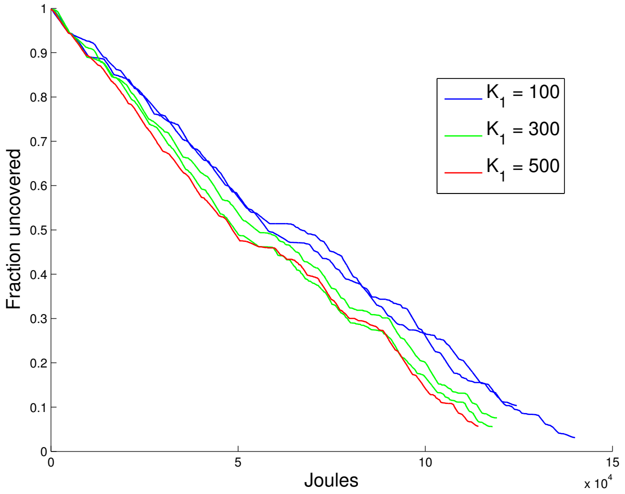


Figure 4.21: Experimental results: energy versus uncovered area for the different values of K_1, ($K_2 = 0$)

determined the energy used by measuring the batter power draw and integrating over time. To measure the energy used for motion, we subtracted the battery power draw while the Packbot was stationary with all computer systems running.

Since we are using a trajectory tracking controller, the final time of the experiment matches the final time of the optimal trajectory; the controller will drive the Packbot to the desired position and make up for lost distance due to inexact following of commands. Figure 4.21 shows the comparison of energy expended and the fraction of the area uncovered. Tests with a lower fraction uncovered for a given energy are more efficient because they use less energy to cover more area, the opposite of what is predicted in Figure 4.6. Looking at the BD path (Figure 4.1), the first redundant movement occurs after approximately half the area has been covered.

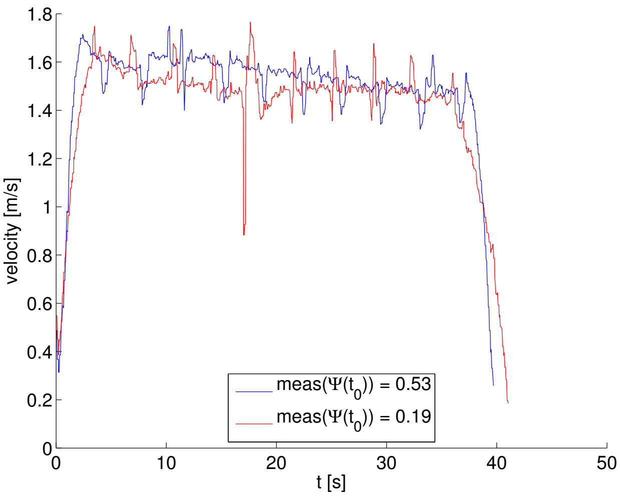


Figure 4.22: Comparing segments from middle and end of path

Table 4.4 compares the simulation results with the experimental results. In the simulation results, as the area weighting factor K_1 increases, our energy usage, using the energy model u^2, increases, and the time required to complete the coverage decreases. Our experimental results show the opposite: as K_1 increases, the energy required for the Packbot to complete the coverage maneuver decreases. We conclude that our model u^2 does not accurately represent the energy used by the Packbot and further study and model identification is required to present a better cost function that gives the desired results. Motor efficiency is one possible explanation for the discrepancy and was added to the cost function (4.2) due to these results.

Figure 4.22 shows the velocity profile for two different segments of the path. Note that the velocity exceeds the "maximum" velocity of 1.5 m/s. We limited the velocity to 1.5 m/s because we are using a trajectory tracking controller and need to be able to speed up to eliminate position errors. The segment later in the path (lower $\text{meas}(\Psi(t_0))$) takes longer to complete and converges to the 1.5 m/s "maximum" more quickly, due to less position error from traveling less aggressively.

4.5 Contributions

In this chapter, we have developed a cost function and resulting optimal trajectory for covering an area with known obstacles (Innovation 2). We use the Boustrophedon Decomposition path planning method to find a set of waypoints and an obstacle free path that would cover the region. We then used an optimal control framework to plan a trajectory that would follow the path, trading off coverage time for energy usage. By varying the values of the gains in the cost function, we are able to quantify tradeoffs between the area covered, force inputs, and efficient motor operation (Contribution 2). This is demonstrated in three different cases. Simulation results indicated that up to 25% of the energy required could be saved by lengthening the time required to travel the coverage path distance. We also introduced the moving turn modification to the trajectory that reduced the time and energy required to traverse the coverage path.

When completing a coverage operation, cost function weights K_1 and K_2 can be set based on the relative scarcity of time and energy. In situations where energy concerns are greater, K_1 and K_2 can be set based on the tradeoffs presented in this chapter.

Future work includes modifications to the path generation algorithms to better support our coverage cost function. An initial idea is to develop algorithms that place a priority on long passes before searching areas that require short segments. Additional experiments can be conducted to include more data points to match the results in Section 4.3.

CHAPTER 5

Energy and Thermal Management of Hybrid UGV Power System

Future robotic vehicles including both small unmanned ground vehicles (UGVs) and full-scale vehicles will carry multiple sources of power, including batteries, fuel cells, combustion engines, ultracapacitors, and/or solar cells, to enable long-endurance operation. Fuel-based power sources have a higher specific energy than batteries, thus most current automobiles are gasoline-powered. Batteries have other advantages, such as low noise profile, easy replacement, and direct energy conversion. Energy harvesting from solar or wind sources exploits natural resources to increase total energy reserves. Long-term missions, especially for autonomous robots that can operate indefinitely without human contact, will require power systems operating together at maximum efficiency.

This paper proposes a framework to model and manage multiple power sources on a single vehicle using a hybrid system framework [82]. Using this modeling framework, an optimization method is presented to maximize energy efficiency and minimize energy losses. The hybrid systems model adapts easily to systems that can be turned on or off and optimizes the on/off switching times. Optimization of the full nonlinear hybrid system is a complex problem [78–81]. To make the problem tractable, we take the approach of simplifying the nonlinear model to a hybrid automaton model with constant dynamics in each mode. The framework and simplification process comprise Innovation 3 of this dissertation.

In contrast to most control algorithms developed for hybrid-powered automobiles, this model incorporates both variable and constant power production levels. The motivation for this model is a hybrid power system containing an Ultra Electronics, Advanced Materials Inc. (AMI) 200 W fuel cell that can only be turned on or off. This fuel cell draws propane at a constant rate when producing power and requires several minutes and nontrivial power input to transition between on and off. The power system configuration studied in this chapter consists of the AMI fuel cell and a battery. Two different batteries are studied for this power system, a BB2590 lithium-ion battery and a lead-acid battery. We consider an area coverage mission, with trajectories planned using the method described in Chapter 4. To determine the full power usage over a coverage trajectory, we use simulation models developed in [39, 95, 96].

The operation of the fuel cell and battery is optimized using the hybrid automaton model and compared to a state-of-charge-based controller. We consider two optimization goals: energy efficiency and thermal efficiency. For energy efficiency, we penalize energy losses in the power components. Thermal efficiency penalizes high operating temperatures for some power components, such as a battery. Using this framework, we present Contribution 3: Intelligent management of the hybrid power system. We show increased energy efficiency of the power system and ability to maintain the battery temperature within the desired range. One of the key assumptions for the model simplification is that both dynamics and power demand can be approximated by averaging. This assumption is tested in a hardware system along with a comparison between a conservative power system controller and the optimization routine. These experiments constitute Contribution 4 of this dissertation.

This chapter is organized as follows: Sections 2.5 and 2.8 summarize related research and background on hybrid models. Section 5.1 formalizes the objectives of the power management controller and presents the modeling formalisms and optimization techniques used in this chapter. An application to the fuel cell/battery hybrid power system is presented in Section 5.2 with optimization results and comparison to a baseline controller presented in Section 5.3. Experimental validation of the optimization method is presented in Section 5.4. The optimization model and cost function are extended to include thermal dynamics and costs in Section 5.5, with conclusions and future work presented in Section 5.6.

5.1 Problem Statement and Approach

This chapter studies the optimization of UGV power systems comprised of multiple power sources. Specifically, a fuel cell with fixed power output and startup and shutdown transients augments a traditional battery system. Due to the extended transients and extended battery charge/discharge times, optimization is performed over the entire mission. The full optimization problem of determining the mode sequences and optimizing the timing of the controlled transitions is a complex Mixed Integer Nonlinear Programming problem [79]. Solving these problems is a very complex and time intensive process and intractable for the planning a mission duration considered in this paper. We present simplifications to the full hybrid model to create a system model that is quickly optimized. Using this simplified model, inputs to the individual components are computed.

In this chapter, we consider two different system costs: energy loss and high operating temperatures. Energy efficiency is explored first, with the model extended to include temperatures in Section 5.5. Another possible optimization criteria consists of costs for ending in particular energy

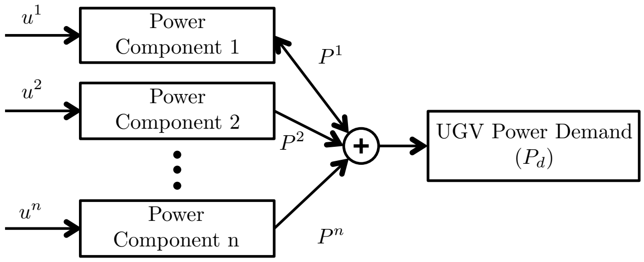


Figure 5.1: Interconnection of power components

states such as low battery state of charge. In addition, we require that the power components must not be operated such that they enter an inoperable energy-depleted state; this constitutes mission failure since the UGV would become stranded once it fully shuts down.

5.1.1 Modeling Framework

Figure 5.1 shows a generic interconnection of n different power components. The set of power component indices is denoted $S = \{1 \dots n\}$. In this section, the superscripts refer to $i \in S$. These components can either generate and consume power (bidirectional arrow) or exclusively generate power (unidirectional arrow). Together, the power components must provide sufficient power to meet the UGV power demand P_d.

5.1.1.1 Component Models

Each component is modeled individually as a hybrid system. In this chapter, we base our definition of a hybrid automaton on the definitions provided by [82, 97]. Each model is the tuple $M^i = (Q^i, X^i, E^i, G^i, U^i, f^i, P^i, J^i_c, J^i_d)$ where

- Q^i: set of discrete states. Each different $q^i_j \in Q^i$ represents a different mode of operation for the system. The discrete state of component i is denoted q^i.
- X^i: continuous state space $X^i \subseteq \mathbb{R}^{m^i}$. The continuous state is denoted x^i and m^i is the dimension of the continuous state.
- E^i: set of discrete state transitions. E^i_{jk} represents the transition between q^i_j and q^i_k. Each transition has a map Λ^i_{jk} that resets a subset of the continuous variables to a known value. The controllable transitions E^i_{cont} occur based on event inputs and form a subset of E^i.
- $G^i : (Q^i \times Q^i) \rightarrow 2^{X^i}$: transition guards. The transition from state q^i_j to state q^i_k is enabled when the state $x \in G_{q^i_j q^i_k}$. In this chapter, we represent the transition guards as the linear inequalities $G_{qq'} = \{x | C_{qq'} x \leq h_{qq'}\}$.

- U^i: Set of continuous input variables $U^i \in \mathbb{R}^{\ell^i}$. u^i is the continuous input and ℓ^i is the number of inputs for the ith component.
- $f^i : (Q^i, X^i, U^i) \to \mathbb{R}^{m^i}$: continuous vector field on the discrete and continuous states and continuous input. When $q^i = q^i_j$, the dynamics are written as $\dot{x}^i = f^i_j(x^i, u^i)$.
- P^i: power output. The power output is a function of state and input. When $q^i = q^i_j$, $P^i = p^i_j(x^i, u^i)$.
- J^i_c: continuous state cost function. In this chapter, we are first interested in optimizing energy usage to limit energy loss and define the cost function as the energy lost in each component (e.g., resistance and heating losses in a battery). In Section 5.5.2, a thermal penalty is added to some system components. When $q^i = q^i_j$, $J^i_c = h^i_j(x^i, u^i)$.
- J^i_d: discrete transition cost function. Each transition E^i_{jk} has an associated cost $J^i_{d,jk}$, which might be zero or positive.

Figure 5.2 shows an example of a battery model with four discrete states and one continuous state x, the state of charge. The transition guard G_{23} is $\{x = 1\}$, i.e. the battery is full, and E_{23} occurs when $x \in G_{23}$. Similarly, the guard G_{14} is $\{x = 0\}$. The other transitions are controllable and the guards for the these transitions are $\{x \in [0, 1]\}$. For each transition, the reset map Λ_{jk} is the identity.

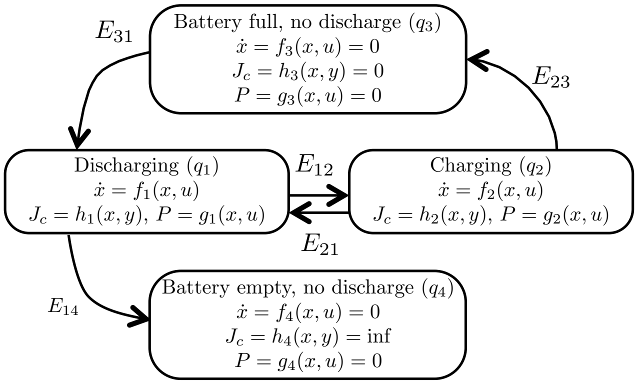

E_{31}
Battery full, no discharge (q_3)
$\dot{x} = f_3(x, u) = 0$
$J_c = h_3(x, y) = 0$
$P = g_3(x, u) = 0$
E_{23}
Discharging (q_1)
$\dot{x} = f_1(x, u)$
$J_c = h_1(x, y)$, $P = g_1(x, u)$
E_{12}
Charging (q_2)
$\dot{x} = f_2(x, u)$
$J_c = h_2(x, y)$, $P = g_2(x, u)$
E_{21}
E_{14}
Battery empty, no discharge (q_4)
$\dot{x} = f_4(x, u) = 0$
$J_c = h_4(x, y) = \text{inf}$
$P = g_4(x, u) = 0$

Figure 5.2: Example of a battery model with three discrete states

5.1.1.2 Combined Model

To create the combined model, we first create a parallel composition of the discrete state models $Q^1 \,||\, Q^2 \,||\, \ldots \,||\, Q^n$ [98, Section 2.3.2]. The size of the resulting discrete state space is the product of the size of the individual discrete state spaces. The continuous states and inputs are concatenated as $x = [x^1 \ldots x^n]$ and $u = [u^1 \ldots u^n]$, respectively. Some continuous states represent times that can trigger a transition and can be reset. The power output of the entire system is the sum of the individual power outputs $P = \sum_{i \in S} P^i$. The total continuous cost is the sum of the individual costs $J_c = \sum_{i \in S} J^i_c$.

To satisfy power demand over the mission, the power output P must always be greater than or equal to the demand P_d. If the maximum power that can be produced in q_k is less than P_d, that state can be removed from the model. For example, if the battery modeled in Figure 5.2 is paired with a device that can be turned off, when the other device is off, the battery must be discharging, otherwise the required power is not produced. This constraint allows pruning an infeasible subset of the discrete state space, simplifying the problem.

An additional discrete state q_{mc} is created to represent mission completion. In this chapter, we consider a mission to be complete after a certain amount of time t_{mc} has elapsed. To model the mission time, an additional continuous state representing time elapsed is added and transitions are added between every discrete state and the mission completion state.

5.1.2 Optimization

Given a UGV coverage trajectory, we seek to optimize power use over the mission. First, we define a sequence σ as a sequence of discrete states $q_1 q_2 \dots q_l$, $q_j \in Q$, with accompanying transition sequence $e_1 e_2 \dots e_{l-1}$, with e_j such that $E_{q_j q_{j+1}} \in E$. To maximize energy efficiency, our goal is to choose the continuous input u and the controllable transitions E_{cont} to minimize the following cost function:

$$\begin{aligned} J = \int_{t_0}^{t_f} J_c\left(x(t), q(t), u(t)\right) \; \mathrm{d}t + \sum_{q_i, q_{i+1} \in \sigma} J_{d, q_i q_{i+1}} \\ + D\left(x\left(t_f\right), q\left(t_f\right)\right). \end{aligned} \tag{5.1}$$

J_c is the continuous cost of the system and is dependent on how long the system is in a given state. Energy inefficiency is one possible continuous cost and is used in this paper. J_d is the discrete transition cost and can be used to apply a one-time cost or penalty for entering a certain mode of operation. D is the terminal cost for the model $D\left(x\left(t_f\right), q\left(t_f\right)\right) = \sum_{i \in S} D^i\left(x^u\left(t_f\right), q^i\left(t_f\right)\right)$ and t_f is the final mission time. The initial states $q(t_0)$ and $x(t_0)$ are assumed given and the final discrete state $q(t_f) = q_f \in Q_f$, the set of goal states. By defining $J_c(x(t), q(t), u(t))$ as energy loss, the optimization will find the most efficient usage profile for the different power sources over time. Using energy lost as the cost function, the optimization will find the most efficient combination of power sources over time for the UGV. Adding a thermal penalty for high temperatures will create a tradeoff between energy efficiency and temperature.

5.1.2.1 Simplification of the Optimization Model

Since our optimization considers the entire mission, optimizing the full hybrid model is complex [78, 79, 99]. To simplify the optimization problem, we transform the general hybrid model into a hybrid automaton model. For components with no continuous input, the continuous dynamics must be constant $\dot{x} = k_i,\ \forall i\ s.t.\ q_i \in Q$, defining an integrator hybrid automaton [97]. If the power system can be controlled through a continuous input, the dynamics must be in the form $\dot{x} = u,\ u \in [u_{lower,i}, u_{upper,i}],\ q_i \in Q$, defining a rectangular hybrid automaton [97]. The system models in Section 5.2 all simplify to an integrator hybrid automaton and only this type of automaton is considered. These methods can be extended to a rectangular hybrid automaton by using the methods in [97].

To represent system dynamics in the form $\dot{x} = k_i$, the dynamics are averaged. This requires averaging the power demand over the entire mission. The accuracy of this average depends on mission conditions (e.g. consistent terrain, slope, mission operation); we assume conditions are sufficiently uniform that the mission-averaged dynamics are sufficient for optimization. To compute the average dynamics, the model is simulated in each discrete state and the dynamics over the entire simulation are averaged. In the case of significant variation in continuous dynamics within one discrete state, that discrete state can be split into multiple discrete states. These new states partition the original discrete state into different regions, each of which can be adequately represented by averaged dynamics. In Section 5.2.4, we demonstrate creating additional states based on significant variation in dynamics. Section 5.5.2 considers a case where the dynamics are both positive and negative within a single discrete state. In addition, the cost J_c is averaged within each discrete state.

5.1.2.2 Optimization Method

Having simplified the model to a hybrid automaton, optimization methods described by Pang et al. can be used [97]. This algorithm for optimizing a hybrid automaton with no continuous inputs is decomposed into two steps:

1. Sort the (possibly infinite) set of state transition sequences from initial state q_0 to the goal states q_f by the sum of their discrete transition costs $\sum_{q_i,q_{i+1}\in\sigma} J_{d,q_i q_{i+1}}$
2. For each sequence, optimize the timing of the controllable transitions $e_j \in E_{cont}$ to minimize the cost function (5.1).

The sequence with the lowest cost is the solution to this optimization and represents the sequence of discrete states that the power system will traverse to achieve the mission with the least energy lost.

To explore the set of state transition sequences, we use a graph search algorithm defined in Algorithm 5.1. This algorithm uses the sum of discrete transition costs to define the order of sequences to be evaluated. The algorithm starts at the initial discrete state q_0 and creates state sequences by appending all possible next states of the last state in the current sequence. If a

Algorithm 5.1 Graph search algorithm for optimization

```
Initialize set of possible sequences $S = \{\sigma_i = q_0\}$
Initialize optimal sequence $\sigma_{opt} = \emptyset$ and optimal sequence cost $J_{opt} = \infty$
repeat
    Sort $S$ by sum of discrete transition costs $J_{d,\sigma} = \sum_{q_i,q_{i+1} \in \sigma} J_{d,q_i q_{i+1}}$
    Remove $\sigma_{min} = q_1 \dots q_n$ with minimum $J_{d,\sigma}$ from $S$
    if $q_f \in \sigma_{min}$ then
        Optimize transition timing using (5.2)
        Compute optimal cost $J_{min}$ for $\sigma_{min}$
        if $J_{min} < J_{opt}$ then
            $\sigma_{opt} = \sigma_{min}$, $J_{opt} = J_{min}$
        end if
    else
        $\forall q_j$ s.t. $E_{q_n,q_j} \in E$, add $\sigma = q_1 \dots q_n q_j$ to $S$
    end if
until $J_{d,sigma} > J_{opt}$ or $S = \emptyset$
```

sequence contains the final state q_f, then the transition timings are optimized by using the linear program described below in (5.2). The search terminates when the minimum sum of discrete costs exceeds the optimal cost thus far or the set of sequences not yet explored is empty.

Each sequence σ is optimized by setting up the following linear programming problem:

$$\begin{aligned} &\min \lambda^T t \\ &s.t.\ At \leq B \end{aligned} \tag{5.2}$$

In this equation, λ is the cost for being in each state of the sequence $\lambda = [\bar{J}_{c,q_1} \bar{J}_{c,q_2} \dots \bar{J}_{c,q_j}]^T$ and t is the time spent in each state of the sequence. The transition constraints $At \leq B$, which ensure that the transition guards are met given the time spent in each state, are written as [97]

$$A = \left[\begin{array}{ccccc} C_{q_1q_2}k_1 & 0 & 0 & \dots & 0 \\ C_{q_2q_3}k_1 & C_{q_2q_3}k_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ C_{q_{l-1}q_l}k_1 & C_{q_{l-1}q_l}k_2 & C_{q_{l-1}q_l}k_3 & \dots & C_{q_{l-1}q_l}k_{l-1} \\ \hline -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -1 \end{array} \right] \tag{5.3}$$

$$B = \left[h_{q_1q_2} - C_{q_1q_2}x_0,\ \dots,\ h_{q_{l-1}q_l} - C_{q_{l-1}q_l}x_0 \mid 0,\ \dots,\ 0 \right]^T \tag{5.4}$$

The variables $C_{q_iq_j}$ and $h_{q_iq_j}$ are defined in Section 5.1.1 as transition constraints, k_i is defined in Sections 5.1.2.1 and x_0 is the initial state value at the beginning of the mission. The Multi-Parametric Toolbox for MATLAB is used to solve the resulting linear programming problem [100].

5.2 Application to Fuel Cell/Battery System

In this section, we consider a UGV with a fuel cell and a battery as dual power sources. While the framework we propose is general for all system setups, the characteristics of this setup under consideration for long-duration UGV missions influenced the development of our framework. The fuel cell has model-specific limitations; in particular, it is either off, on, or transitioning between those two states. These transitions take a significant amount of time and require a power input.

Two different battery models are considered in this section, the BB2590 lithium ion battery and a lead acid battery pack. While the initial simulations were completed with the BB2590 model, the lead acid battery model was the only option available for laboratory experiments. Because of the general modeling framework that is developed, we were easily able to change the battery models in the optimization routine. The fuel cell and BB2590 models are discussed in greater detail in [101].

For each component, we define a continuous cost function, denoted J_{fc} and J_{batt} for the fuel cell and battery respectively. In this section, we are looking exclusively at energy efficiency in our optimization. Penalties for high temperature operation are discussed in Section 5.5.2. The overall cost function for this system is derived from equation (5.1) as

$$J = \int_{t_0}^{t_f} J_c \, \mathrm{d}t = \int_{t_0}^{t_f} J_{fc} + J_{batt} \, \mathrm{d}t \tag{5.5}$$

Note that we assume that there are no penalties for state transitions ($J_d = 0$) or for the final state ($D = 0$). These penalties could be added to change the characteristics of the optimal result. In particular, a penalty for low ending battery SOC could be added, potentially allowing for another mission to be started immediately.

5.2.1 Battery Model

We consider two different battery packs in this chapter. The BB2590 lithium-ion battery pack is commonly used in military applications. For this research, we use the thermal-electric battery model developed by Kim et al. [95]. The model has two continuous states, battery state of charge (SOC) and temperature T, and one input, battery current i_c. In this section, we only consider SOC dynamics and incorporate thermal dynamics in the optimization in Section 5.5. As such, we write

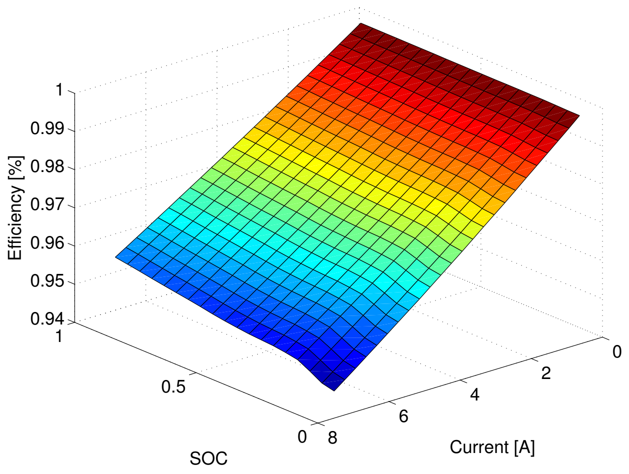


Figure 5.3: Battery efficiency as a function of SOC and current for the BB2590 battery pack

the continuous state $x = soc$ and input $u = i_c$. This model has been parameterized and validated against the BB2590 battery pack [95] and is encoded in a SIMULINK module.

We also use a lead-acid battery pack. While the lead-acid battery pack is less likely to be used in a military application, the hardware setup described in Section 5.4 requires the use of this battery pack. The lead-acid battery model is presented in [102] with parameters fitted to our particular battery pack. This model only has one continuous state, SOC, and the battery current input (positive for current draw). Details of this model fit are presented in Appendix A.

For both battery models, we formulate a discrete state structure as shown previously in Figure 5.2. The battery switches to the fully charged state q_3 when $x = 100\%$ and switches to the empty state q_4 when $x = 0\%$. When the battery is charged, the input current must be zero and if the current becomes negative (power drawn from the battery), the state switches to discharging. Transitions between charging and discharging are based on the battery current. In each discrete state, the dynamic function for SOC is written $\dot{x} = f_i(x, u)$. This function depends on which battery model is used.

For this battery system, the main loss of energy is due to heating and resistive losses. The cost of the battery is written as $J_{batt}(x, q, u) = (1 - \mathit{eff}(x, q, u))\, P_d$, which represents the power lost to resistance and heating. The efficiency of the battery $\mathit{eff}(x, q, u)$ is the percentage of total power generated by the battery used to power the UGV. Using the battery model, we parameterized the efficiency of the battery based on SOC and the input current, as shown in Figure 5.3. A similar efficiency map exists for the lead-acid battery pack. These values were obtained by simulating the system with alternating charging and discharging current of the same magnitude. Due to charging limits for the battery, this method could only calculate efficiency values for currents up to the charging limits. Efficiency values for higher discharge currents are obtained by extrapolating the

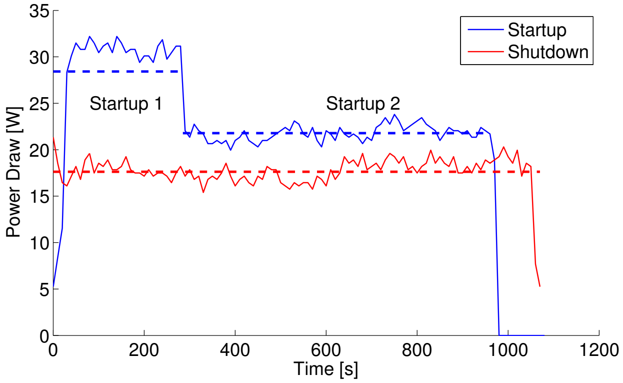


Figure 5.4: Power draws during startup and shutdown (dashed line- average value)

efficiency at lower currents. Since the efficiency is roughly constant for a given current draw at different SOCs, we can also average the efficiency in each discrete state for the simplified model.

5.2.2 Fuel Cell Model

This fuel cell power source used in this work was designed and manufactured by Ultra Electronics, Adaptive Materials, Inc (AMI). It is fueled by commercial propane canisters and consists of a 200 W solid oxide fuel cell, a fuel reformer, and a DC/DC converter. The propane gas is first desulfurized and then reformed via partial oxidation into a hydrogen-rich fuel stream to feed the fuel cell. This fuel cell operates at 14% thermal efficiency, converting 14% of the available chemical energy in the propane to electrical energy. The fuel cell was designed to be integrated with existing batteries on small UGVs. This combined power system significantly increases the possible mission duration, especially under low power loads such as persistent stare missions.

One of the challenges of integrating this power source is to develop an optimal duty cycle for using the fuel cell to recharge the batteries. Solid oxide fuel cells have significant warm-up and cool-down periods during which power is consumed to run blowers or heating elements. The power consumption of the fuel cell during these periods becomes a crucial factor for optimization, since there is an energy cost associated with each on/off cycle of the fuel cell. Figure 5.4 shows the power draws for startup and shutdown.

Figure 5.5 shows the states of the fuel cell represented by a hybrid automaton. As shown in the diagram, there are five discrete states, labeled q_1 to q_5, and two continuous states x, fuel level F and state timer t. When $q = q_3$, the fuel cell consumes fuel ($\dot{F}$) and produces up to power (P_{fc}). While the power output of the fuel cell can vary, the fuel input is constant, so any power output less than 200 W is wasting energy. There is no continuous input u and the transitions from *Fuel Cell Off* to *Startup 1* and from *Fuel Cell On* to *Shutdown* are controlled. To switch from off to on (or

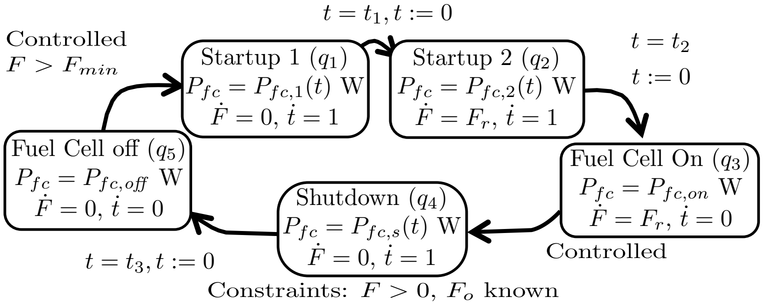


Figure 5.5: Hybrid model of fuel cell with discrete states

on to off), the fuel cell passes through transition states. The timer t is used to govern the transitions between these discrete states. While in these transition states, the fuel cell draws power from the rest of the system. Power usage during these phases was recorded from experimental tests. The transition guards and the reset map are shown in Figure 5.5 using notation from [82].

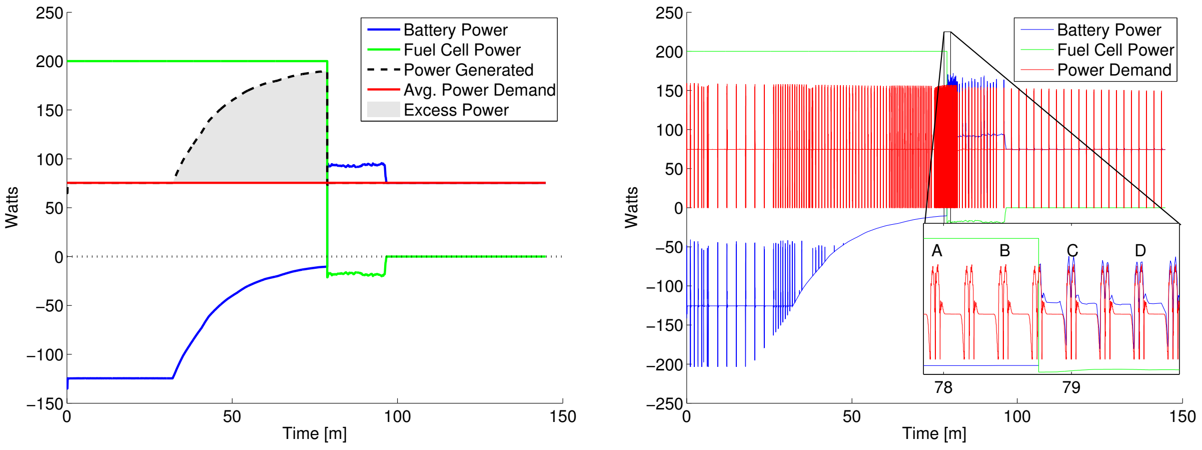
This fuel cell has two different sources of loss: power draws during startup/shutdown and excess power generation. The power draws are known based on measured startup and shutdown cycles. The fuel cell also draws a small amount of power when off due to internal monitoring electronics. Excess power generation is determined by subtracting the UGV power demand and the battery power from the power generated. We can write the cost function as

$$\begin{aligned} J_{c,fc} &= P_{draw} + P_{excess}, \\ P_{draw} &= \begin{cases} 0 & P_{fc} > 0 \\ -P_{fc} & P_{fc} \leq 0 \end{cases} \\ P_{excess} &= \begin{cases} 0 & P_{fc} \leq 0 \\ P_{fc} - P_d - P_b & P_{fc} > 0 \end{cases} \end{aligned} \tag{5.6}$$

In this equation, P_d is the power demand and P_b is the power from the battery.

5.2.3 Power System Simulation

The nonlinear power system model for the battery from Section 5.2.1 has been integrated in SIMULINK with the fuel cell model described in Section 5.2.2. Figure 5.6 shows the example power demand (P_d) trends for constant and variable power profiles. The first case, shown in Figure 5.6a, is based on a constant power mission, such as a persistent stare, where the power demand is constant. In general, persistent stare missions would have low power requirements, needing to only operate minimal electronic and communication loads. Previous experiments showed a 30 W power load while remaining stationary, as described in Chapter 3. Additionally, we can ap-



(a) Constant power demand, $P_d = 75.5$ W

(b) Variable power demand

Figure 5.6: Simulating the power components for a variable-power and a constant-power mission

proximate some variable power missions with an average power draw. For ease of comparison between constant and variable power demands, constant power demand P_d is set at the average power demand from the variable-power mission.

The variable power profile, shown in Figure 5.6b, is based on an area coverage mission. Using the method described in Chapter 4, we planned a reference path and trajectory for the coverage mission. Figure 5.7 shows a portion of the coverage path, with the robot moving from right to left. The marked locations correspond to Figure 5.6b, showing how the variations in power demand correspond to turning segments of the UGV's path, while straight travel segments have constant power demand.

In both cases in Figure 5.6, the battery is initialized to 25% SOC. As the battery charges, it uses all the available power from the fuel cell until about 60 minutes into the mission. At this point, the SOC is sufficiently high that the battery must be charged at a lower rate, based on the known charging curve for the battery. In both missions, the maximum charging rate follows a similar curve, with some deviation in the variable power demand case. In the constant power mission, the total power demand is plotted, showing that the power generated (fuel cell power plus battery power) is greater than the power demand. This excess power generation represents a loss for the system and should be avoided. When the fuel cell switches off, at about 79

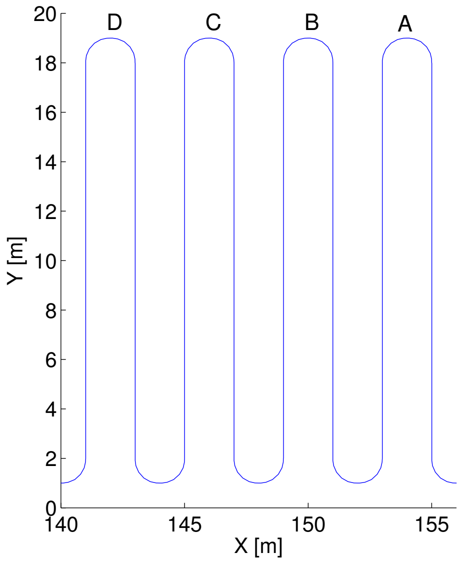


Figure 5.7: Portion of area coverage path, letters correspond with Figure 5.6b

minutes, the fuel cell power becomes negative and the battery power exceeds the power demand. Once the shutdown process is completed, the fuel cell power draw decreases to almost 0 W and the battery continues to primarily power the simulated UGV. For more discussion of this simulation, see [101].

5.2.4 Combined Model

Having defined the individual component models, we now present the combined model for the mission in question. We assume that the average mission power demand $P_{d,avg}$ and total mission time t_f are known. The mission parameters used in this section are presented in Section 5.3.1. The combined discrete state space has 21 states initially (4 battery states $\times$ 5 fuel cell states $+$ 1 mission completion state). Since the battery must be discharging while the fuel cell is in any state except on, most of the discrete states representing battery charging and fully charged are eliminated. Since we assume that $P_{d,avg} < P_{fc,on}$, the battery is always charging or charged when the fuel cell is on. Since the *Battery Empty* state has no transitions out and represents an incapacitation of the robot, regardless of the fuel cell state, we can create a single battery empty "failure" state.

There are three continuous states: the battery state of charge (x_1), fuel cell transition timer (x_2) and the mission timer (x_3). For simplicity, we assume that fuel levels for the fuel cell are sufficient and do not model the fuel level. For each discrete state, with a known, constant power demand and fuel cell power output, battery power required can be computed. Using the battery model described in Section 5.2.1, the required battery current u_{batt} and the resulting average $\dot{x}_1 = I_i$ for each state can be computed.

In this section, we use the BB2590 battery model for simulation. To calculate average battery dynamics, we initialize the battery to fully charged (for discharge states) and fully discharged (for charge state). While the battery won't necessarily start each state at full charge/discharge, we are averaging the battery dynamics over the entire range of possible continuous states. The battery is then discharged (or charged) at the constant power $P_d - P_{fc}$ until the battery is fully discharged (or charged).

For most of the fuel cell states, $\dot{x}_1$ remains fairly constant. The black line in Figure 5.8 shows the discharge current for the *FC Shutdown* state. There is little variation between the actual (solid) and average (dashed) dynamics. The other discharge states show similar current profiles for constant power.

However, when the battery is charging, the battery constraints limit current that can be applied, eventually dropping to zero when the battery is charged. In order to accurately capture this change in dynamics over time, we divide the *Fuel Cell on* state into several states based on the state

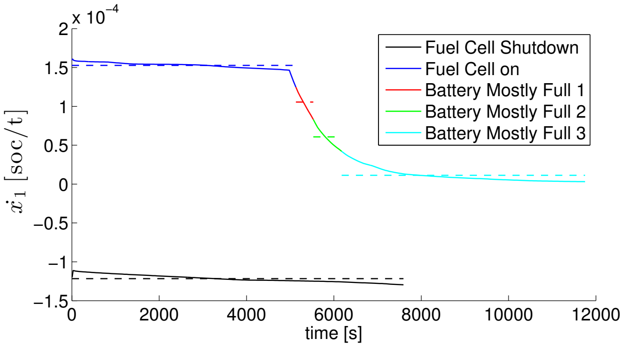


Figure 5.8: Comparison of actual and averaged $\dot{x}_1$ values for *Fuel Cell Shutdown* (battery discharging) and *Fuel Cell On* (battery charging) states

Table 5.1: Values for hybrid automaton model parameters, shaded values dependent on P_d

State Name	Number	Energy Cost [J]	I_i $[soc/t]$	P_B	P_{fc}	Time [m]	SOC limit [%]
FC Startup 1	1	31.4	-1.367e-04	-104	-28.8	5	-
FC Startup 2	2	24	-1.272e-04	-97.3	-21.8	11.5	-
FC On	3	3.49	1.526e-04	-124	200	-	85.8
Batt Mostly Charged 1	4	36.6	1.052e-04	-89.5	200	-	89.9
Batt Mostly Charged 2	5	73.5	6.054e-05	-51.4	200	-	93.7
Batt Mostly Charged 3	6	114	1.128e-05	-9.61	200	-	99.9
Battery Charged	7	125	0	0	200	-	-
FC Shutdown	8	19.7	-1.216e-04	-93.1	-17.6	17.9	-
FC Off	9	1.34	-9.801e-05	-75.5	0	-	-

of charge of the battery. Figure 5.8 shows the division into different discrete states for the area coverage mission, with the charging curve divided into different colors for the different states. The number of new states and the limits soc_i are manually determined based on the changes in dynamics in Figure 5.8 and the desired fidelity to the nonlinear model. Figure 5.9 shows the discrete states of the combined model and Table 5.1 lists the parameter values for the different discrete states.

As will be described in Section 5.3.3, there is some discrepancy between the averaged and full nonlinear dynamics. Part of the discrepancy is due to the fact that the average dynamics are calculated based on a full charge/discharge cycle. To prevent the battery from becoming discharged, a reserve battery level must be specified. We define a value soc_{low} where the optimization considers the battery empty. For the BB2590 battery, we chose $soc_{low} = 0.12$, which provides sufficient reserve to prevent battery depletion in simulation. This safety margin could be specified for different power demands in future work.

In each state q_i, we define a cost value J_{q_i}. This cost value is computed by adding the averaged individual cost functions J_{fc} and J_{batt}. This is done in the same step as determining the averaged dynamics for the battery at each power level.

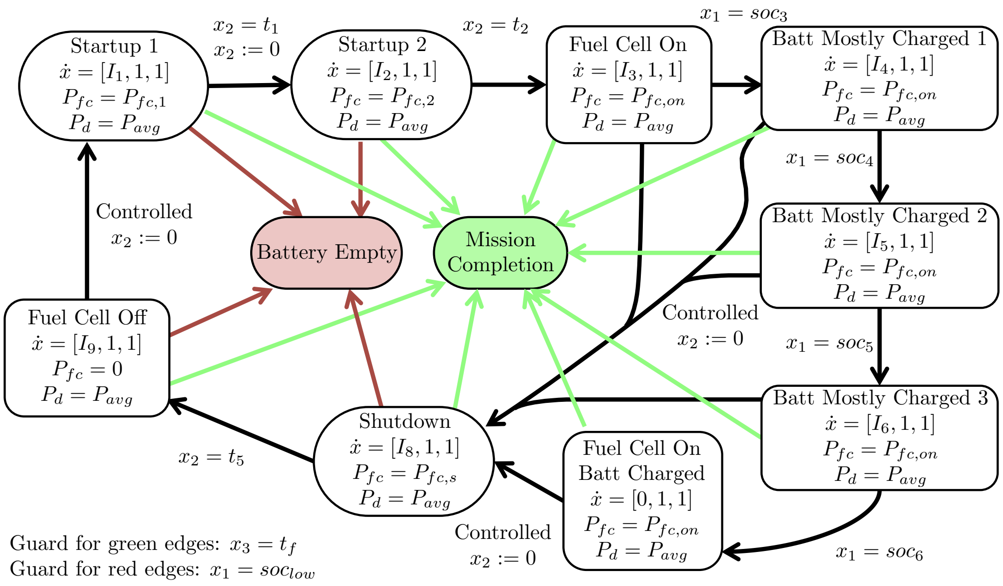


Figure 5.9: Power system modeled as a hybrid automata

The graph search algorithm, defined in Section 5.1.2.2, relies on discrete transition costs J_d to sort the possible sequences and as part of the termination condition. In our model, $J_d = 0$ for all transitions. To account for this issue, we can assign a cost J_d^* to the discrete transitions to create the sorting required for the optimization search algorithm. For each transition, we define the minimum time t_{min} that must elapse before the transition must be taken. For example, the transition between *Startup 1* and *Startup 2* can only occur after the minimum time, specified in Table 5.1, has elapsed. For the transitions between the *Battery Mostly Charged* states, the minimum time is determined based on the relative SOC limits soc_i and the battery charge rate I_i. For some transitions (including all contr

We define the new discrete transition cost for the transition $E_{q_i q_{i+1}}$ as $J^*_{d,q_i q_{i+1}} = t_{min} \bar{J}_{c,i}$ and modify the cost sum $J_{d,\sigma}$ in Algorithm 5.1 as $J_{d,\sigma} = \sum_{q_i, q_{i+1} \in \sigma} J^*_{d,q_1,q_{i+1}}$. Since $t_{min} \leq t_i$ for valid trajectories, $J_{d,\sigma} \leq t_i J_{c,i}$ and these discrete costs are an underestimation of the total cost of the trajectory for a state sequence. The linear programming problem for determining state transition times is unchanged and does not use the assigned cost value J_d^*.

5.3 Simulation Results

In this section, we first present the simulation to determine the power demand for an area coverage mission. This power demand is used in the rest of the simulation and experimental tests. Using this power demand, we explore different SOC-based control strategies. These strategies consist of turning the fuel cell on (off) when a low (high) SOC threshold is met. Different thresholds are used and energy efficiency is compared between the different thresholds. Next, we show the results of the optimization described in Section 5.1.2. In particular, we compare three different types of models: the optimization model, the “full battery” model, and the “variable power” model. The optimization model was presented in Section 5.2.4. The “full battery” model removes the assumption of constant battery dynamics by using the nonlinear battery model. The “variable power” model replaces the average power demand with the time-varying power demand obtained from simulation.

5.3.1 Power Demand

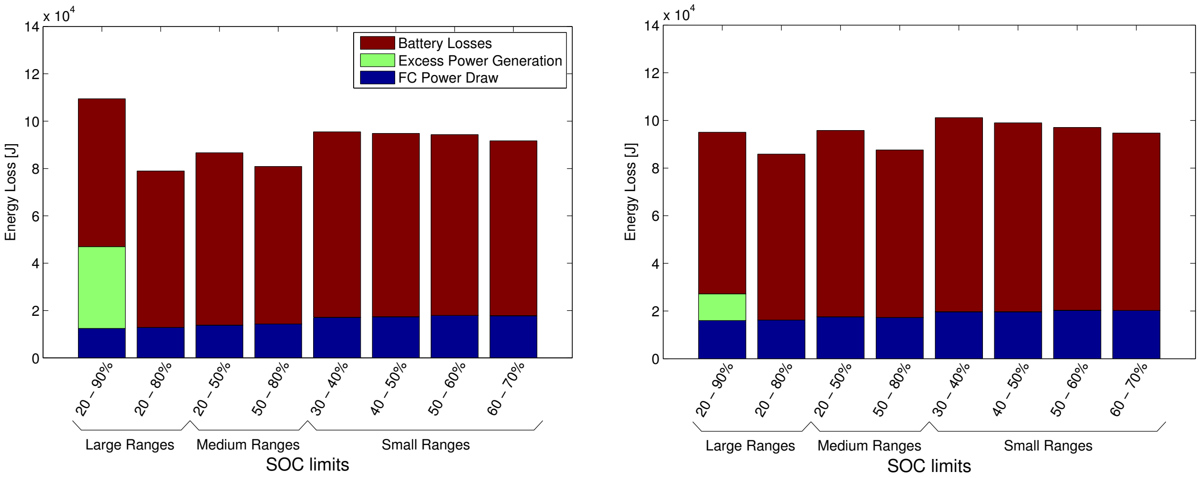
The power demand for a particular UGV will depend on many factors, including the mission and the terrain. In this chapter, we consider an area coverage mission, using an optimal trajectory planning scheme developed in Chapter 4.

To determine the power requirements for the mission, we use a simulation model for the UGV operating in a known environment. This simulation was initially developed as part of a case study for the Automotive Research Center at the University of Michigan [96] focused on modeling an iRobot Packbot. This model includes motor and track-terrain interaction models. We can simulate the desired mission and obtain the power demand over time P_d. To allow for continuous operation, we assume that the averaged power demand $P_{d,avg} < P_{fc,on}$. Otherwise, the battery would eventually deplete, even with the fuel cell on, and leave the UGV incapacitated.

The driving loads can be broken down into resistance due to terrain and changes in kinetic energy. We use the terrain model developed in [39] to compute energy loss due to the terrain. Based on the current vehicle speed and turn rate, for a given terrain, the resulting resistance torques can be calculated. The model must be tuned for every different terrain of interest and is currently calculated for clay, sand, and sandy loam soils.

To provide the torque inputs to the terrain model, we introduce a motor model [96]. This model was obtained experimentally by physically testing motors. The model takes in the current shaft speed and the power being delivered to the motors and calculates the torque output. Together with a simple rigid body model of the UGV, we can simulate the UGV completing a designated mission and record the power used. In addition to variable power demands due to locomotion, electronic components on board require power for operation. We assume that these loads are known and constant over the entire mission.

For the mission for simulation and experimental tests, we defined a coverage scenario for a 500 by 250 m area, with various sections denoted as obstacles. Assuming a search radius of 1 m for the UGV, we planned a coverage trajectory using the method described in Chapter 4. This trajectory requires about 450 minutes to cover. Using this Packbot simulation model, we calculated the total



(a) $P = 75$ W, mission energy usage: $2.02 \cdot 10^6$ J (b) $P = 100$ W, mission energy usage: $2.69 \cdot 10^6$ J

Figure 5.10: Energy losses for different SOC-based limits

(variable) and average power requirements over the entire mission. The variable power consists of long periods (10 seconds to 3 minutes) of fairly constant power draw broken up by brief (1-2 second) turns, with power demands ranging form 0 W to 150 W. The average power demand was 75.5 W.

5.3.2 Baseline Controller Design

As discussed in the previous sections, we consider three sources of energy loss: battery losses, fuel cell power draws (during startup and shutdown) and excess power generated by the fuel cell (see shaded area in Figure 5.6a, for example). Here we consider the two power demand levels shown in Figure 5.10. This figure shows the total energy loss over the 450 minute coverage mission. In all tests, the battery begins at 50% SOC and the fuel cell is initially on. Note that the energy loss is about 4% of the simulated mission energy usage (power demand multiplied by mission duration) for both power levels. For these simulations, the fuel cell is controlled based on battery SOC. When the battery is discharging and reaches the "turn on" SOC threshold, the fuel cell is switched on and begins the startup sequence. After the battery is charged to the "turn off" SOC threshold, the fuel cell is shut off.

Several different SOC thresholds are considered. The first ranges represent large SOC operational ranges for battery operation (20-90% and 20-80% respectively). These ranges have the smallest losses due to battery losses and fuel cell power draw, but with the high threshold at 90% SOC, there is a region where excess power is generated by the fuel cell that cannot be used for

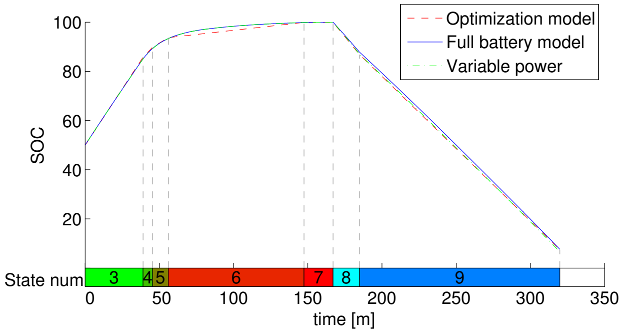


Figure 5.11: Comparison of the average-power model with full battery model for charge/discharge cycle with average power draw of 75.5W, with current state number (see Table 5.1) shown near the horizontal axis

battery charging or for the mission. The excess energy generated is significantly smaller at the higher power demand.

Medium (20-50% and 50-80%) and small (30-40%, 40-50%, 50-60%, and 60-70%) SOC ranges show higher losses due to the fuel cell power draw because the fuel cell turns on and off more frequently, but overall losses depend on the mission power demand. For the higher power demand, there are noticeable differences within the medium- and small-range groups. The range with the higher SOC limits has lower energy usage, due to the higher efficiency of the battery at higher SOC. In fact, with a 50-80% SOC range, the higher efficiency almost balances the additional fuel cell power draw compared to the 20-80% SOC range. For the lower power demand, these trends are not as noticeable.

These observations suggest that operating the fuel cell for as long as possible in each cycle yields the most efficient operation from an energy perspective. However, if the SOC becomes too high, significant amounts of energy are lost due to the inability of the battery to accept power at higher SOC. Also, the fuel cell must be restarted in time for the startup phase to be completed before the battery is fully discharged. Increasing power demand requires a higher turn-on threshold for the fuel cell. For the 40 W and 100 W power demands, the minimum lower SOC thresholds are approximately 9% and 17% respectively.

5.3.3 Fidelity of Optimization Model

Next, we present results comparing three models in a simple charge/discharge cycle: the simplified optimization model, full battery model with averaged power demand and the full power system model with variable power demand. The battery initially has 50% state of charge and the fuel cell is on. The fuel cell is turned off after the battery reaches 100% SOC. Note that this example only uses a portion of the simulation mission, enough to show one charge/discharge cycle.

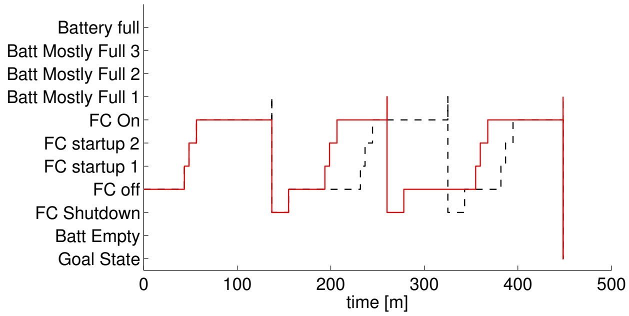


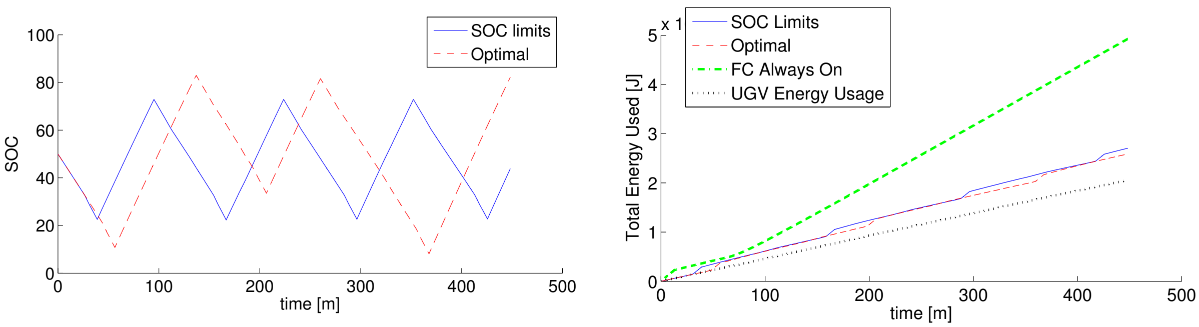
Figure 5.12: State sequence over time for two different optimal trajectories

Figure 5.11 shows the state of charge over time for this simulation. There is close agreement between the three models over the entire trajectory. There is a small gap between the optimization trajectory and the other trajectories when the state of charge is above 90%, when the model is in *Battery Mostly Charged 3* (state 6). This gap is due to the averaged dynamics required by the optimization model. This gap could be reduced by introducing a larger number of *Battery Mostly Charged* states in Figure 5.9 or changing the state of charge limits on the transitions between states, but there is a tradeoff with model complexity. Based on results, we concluded that three intermediate *Battery Mostly Charged* states with the given thresholds yield sufficient model fidelity.

5.3.4 Optimization Results

Running the optimization algorithm described in Section 5.1.2.2, we obtained the optimal power system control inputs for this coverage scenario. For this particular mission, we obtained 16 different state sequences that resulted in the same optimal cost. However, the cumulative time spent in each of the states is the same between the different optimal sequences. There are minor differences in timings, as shown in Figure 5.12. Many of the differences in the state sequences are also visible in this figure. Sometimes the system enters a state, but spends no time in that state. For example, the red trajectory in Figure 5.12 enters the state *Batt Mostly Charged 1* at about 200 m, but immediately transitions to the shutdown state. As a result, there are only two distinct timing sequences of fuel cell on/off transitions in the set of optimal solutions.

Figure 5.13a (line marked “Optimal”) shows the state of charge of the battery corresponding to the red trajectory in Figure 5.12. The UGV ends the mission at a high state of charge, due to the fact that the cost of being in the *FC On* state is less than the cost of being in the *FC Off* state (See Table 5.1).



(a) State of charge over time

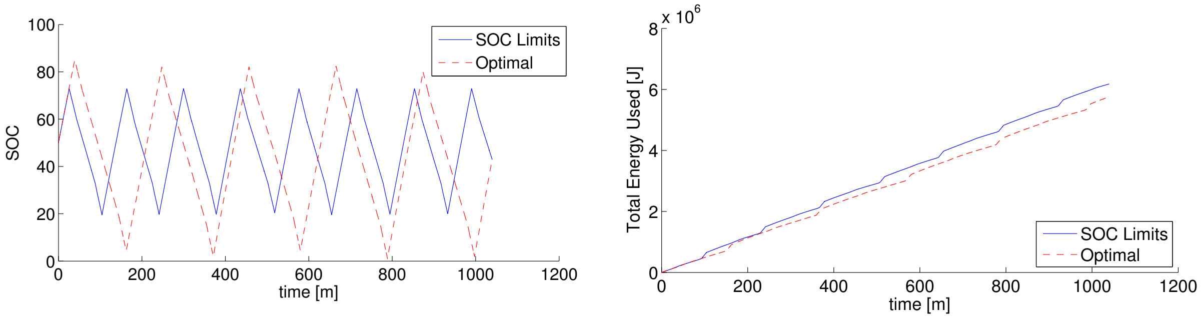
(b) Total energy used

Figure 5.13: Comparing optimal and SOC-threshold controllers, $P_d = 75$ W

5.3.5 Comparison to SOC-based controller

To evaluate the performance of this algorithm, we compare this optimal controller with a SOC-based control law, described in Section 5.3.2. Based on the results from that section, we want to choose as large of a SOC operating range as possible to maximize energy efficiency by minimizing the number of fuel cell on/off transients. However, we also want the controller to work well under a wide range of average power demands. The high threshold is chosen based on when the battery cannot use the entire output of the fuel cell for charging (73% SOC for the system studied). This corresponds to a low average power demand for the mission and this threshold prevents energy loss due to limited charging. The low threshold is chosen such that the battery can provide 200 W output and power the fuel cell for the entire startup procedure without running out of battery energy (33% SOC for this system). The assumed 200 W power demand corresponds to the maximum average power demand where mission duration is not decreased by switching off the fuel cell; at higher power demands, the battery discharges even when the fuel cell is on and will eventually become depleted. The difference in thresholds is as large as possible, demonstrated to be more efficient in Section 5.3.2, while allowing for operation in all conditions and always avoiding excess power generation of the fuel cell.

Figure 5.13 shows a comparison of state of charge over time and the total energy used. Figure 5.13a shows that the state of charge is kept within a narrower range using the SOC-limit controller, resulting in more cycles of the fuel cell. Figure 5.13b shows the total energy used over the course of the mission, which includes energy required to power the UGV during the mission as well as energy losses in the power system. The dotted black line shows energy use for the mission, forming a lower bound for total energy usage with power system losses. Comparing the energy usage by the optimization and the SOC-limit controllers shows a decrease of about 5% at the end of the mission by using the optimization method presented in this chapter. This is consistent with



(a) SOC

(b) Energy usage

Figure 5.14: Larger difference in energy usage as mission length increases

results from Section 5.3.2; the larger ranges of SOC in the optimization controller are more efficient than smaller ranges in the SOC-limit controller. The green line in Figure 5.13b shows that when the fuel cell is left on continuously, the energy usage increases dramatically due to the excess energy produced by the fuel cell when the battery is at a high state of charge.

During the majority of the mission, the energy usage for both the optimal and SOC-limit controllers increases at the same slope as the UGV energy usage. When the fuel cell is turned on, there is a sharp increase in the energy usage. The main improvement by using the optimization controller is the decrease in the number of times the fuel cell is switched on, reducing the excess energy usage during these transitions.

As the mission time increases, the difference in power usage between the optimization and SOC-limit controller will increase, as shown in Figure 5.14. In this case, we consider a mission that is over 1000 minutes long, over twice the length of the previous mission. Since the SOC-limit controller has a smaller range of SOC over the test, as shown in Figure 5.14a, the fuel cell is turned on and off more frequently, resulting in more frequent increases in the energy usage. This will cause the energy usage curves to diverge more over increased mission times, as shown in Figure 5.14b.

Note that in Figure 5.13, the optimal controller completes the mission with a higher ending state of charge. The energy required to charge the battery to a higher SOC is not included in the calculation of total energy usage. Instead, the battery acts as a buffer to store the excess power produced by the fuel cell, but still represents an onboard energy storage element that can later be used to power the robot and, as such, is not energy used. As a result, more fuel is used by the fuel cell during a mission that specifically aims to end with a higher SOC.

The majority of the fuel cell on/off switches occur at consistent SOC levels, especially for longer duration missions (See Figure 5.14a). For a given mission power demand, the optimization can be executed and a SOC limits chosen based on the optimal solution. However, the optimization also takes into account mission duration. In certain cases, the optimization can determine that charging the battery to a higher SOC can prevent an additional on/off cycle for the fuel cell. A simple limit-based approximation of the optimization would not be able to identify these cases, but would achieve better energy efficiency than the conservative limits chosen in this Section. Additionally, the limit based optimization would not be subject to accumulated errors due to discrepancies between the averaged dynamics and actual performance of the system over time. A full study of this approximation is left for future work.

5.4 Experimental Results

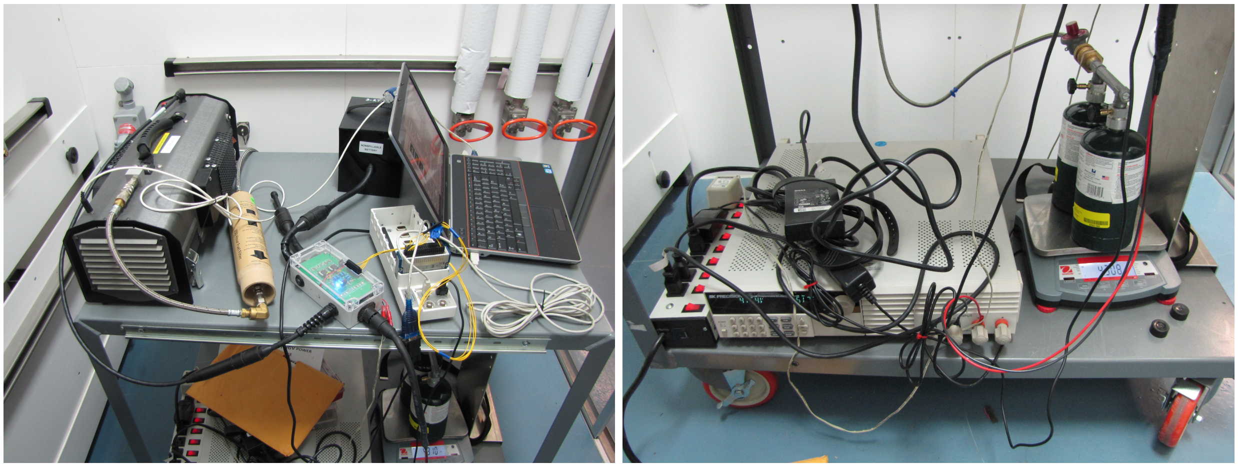
We now verify these models and simulation results using hardware-in-the-loop tests. There are two main experimental goals: confirm that battery performance using an averaged power demand is sufficiently similar to performance under variable power demands, and validate the optimization model against the real system. These tests were performed at US Army TARDEC Ground Systems Power and Energy Laboratory (GSPEL) in Warren, MI.

While the simulation results are based on the BB2590 battery pack, due to incompatibilities between the fuel cell and the battery pack, a TALON lead-acid battery pack is used for the tests. This requires that a model of the lead-acid battery be developed and integrated into the optimization routine in place of the BB2590 model. Details of the experimental model fit are presented in Appendix A. An in-depth description of these tests is presented in [105].

5.4.1 Hardware setup

Our hardware setup consisted of five different components: an AMI fuel cell, lead-acid battery pack, programmable load bank, current and voltage logger, and propane fuel canisters. Since the focus of these experiments is the power system for a ground robot, the load bank is used in place of the robot to simulate the power draws required to complete a mission. This simplifies the experiments by removing the need to drive the robot in a simulated mission and allows for repeatable power loads to use in verifying the simulation results. Figure 5.15 shows the different components used in the system. All components either were commercial products with vendor-supplied interfaces or were designed and built by GSPEL Fuel Cell lab technicians. In this setup, we were able to measure the following variables: battery/fuel cell voltage, battery current, fuel cell current, load bank voltage and current, and fuel weight.

All systems were controlled through a LabVIEW interface. The program initializes communication, loads power command data from a file, records data outputs and sends power commands for the duration of the test, then closes communications. An experiment terminates when the final



(a) Fuel cell, battery, and data logger

(b) Load bank and fuel canisters on a scale

Figure 5.15: Hardware-in-the-loop setup

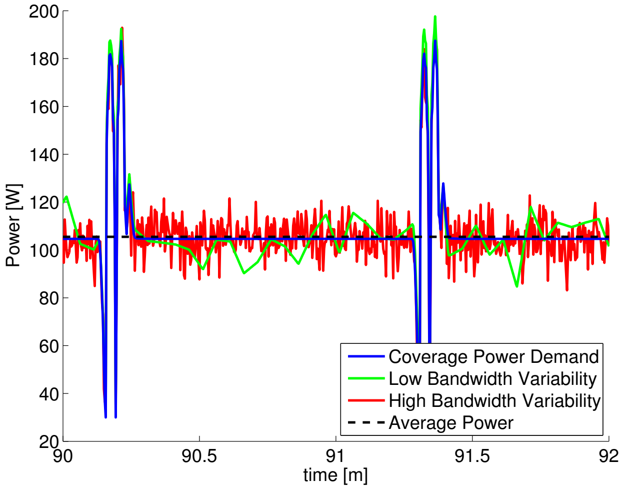


Figure 5.16: Portion of power demands used in tests

time is reached, the stop button is pushed, or an error occurs in any communication stream. While the communication loops occur asynchronously, all recorded data is marked with a synchronized measurement time.

5.4.2 Battery Performance Under Different Power Demands

The first tests compare battery performance under averaged and time varying power demands. One of the main assumptions in our optimization algorithm is that the averaged power demand is a valid approximation of time varying power demand. The simulation models suggested this approximation was valid; the purpose of these tests was to verify that assumption in hardware.

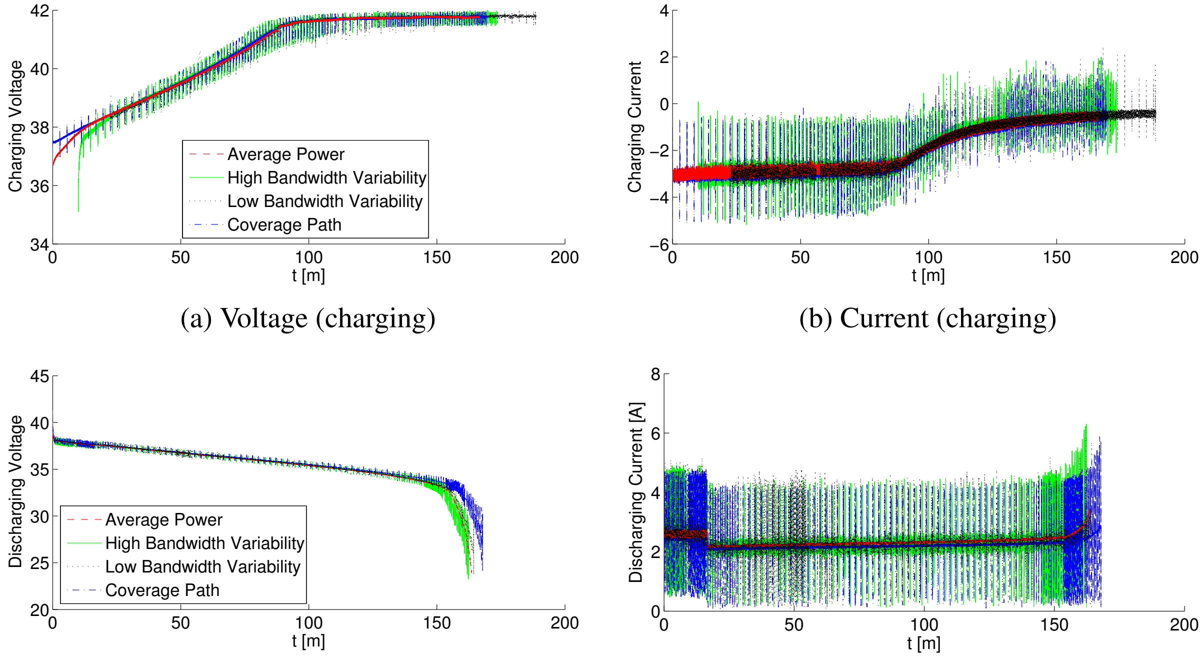
Each test consists of operating the power system (battery and fuel cell) to meet a known power demand time history. The basic power demand for these tests is given by the area coverage trajectory described in Section 5.3.1. To prevent battery charging currents from exceeding the maximum charging current, a 30 W peripheral power demand is added. This power demand is consistent with constant power demands for this type of robot, as discussed in Chapter 3. This power demand, labeled "Coverage Path", obtained through simulation, is smooth; to add more realism, we also introduced different power loads with more variation in the power demand over time. Figure 5.16 shows the different power demand profiles. The variability is a random gaussian sample with the variance taken from tests described in Chapter 3. The "Low Bandwidth Variability" power demand has variability added every 3 seconds and the "High Bandwidth Variability" power demand has variability added every 0.2 seconds. The final power demand tested is the "Average Power".

Tests were run as follows: first, the battery was discharged to about 40% SOC (36.35 V open circuit). The battery was allowed to rest to make sure the desired voltage was achieved. Next, the fuel cell was turned on and allowed to complete its startup sequence. During this time, the power system was under no load and battery output current was recorded. When the fuel cell started producing power, the test was started and the load bank began to draw power according to the desired power profile for the test. Due to the variability of the fuel cell startup time, the battery was at slightly different levels of charge when the tests began. The battery was allowed to charge until power input to battery averaged around 20 W. The fuel cell was then shut off and the test continued until the battery reached the low cut off voltage (18 % SOC).

The tests were divided into charging and discharging portions due to the fact that the fuel cell is shut off at different times. Without this division, the discharge portions of the test could not be readily compared. In each case, we compared the battery voltage and current and the time required to charge/discharge the battery.

Figure 5.17 shows the results from the first set of tests. In these tests, the starting state of charge of the battery was not as uniform at the beginning of the test. Because of this variability, the start times of the "High Bandwidth Variability" and "Low Bandwidth Variability" cases are shifted. This shift was determined manually to counteract the differences in starting state of the battery. After completing this test, a more thorough test procedure was devised to eliminate the need for this manual adjustment.

In these tests, there is close agreement between the voltage and current over time during charge and discharge phases. The total time for discharging the battery is also roughly equal between the different tests. Compared to the "Average Power" case, the "Coverage Path" and "Low Bandwidth Variability" tests take 2.7% and 0.5% longer respectively to discharge and the "High Bandwidth Variability" takes 0.8% less time to discharge.



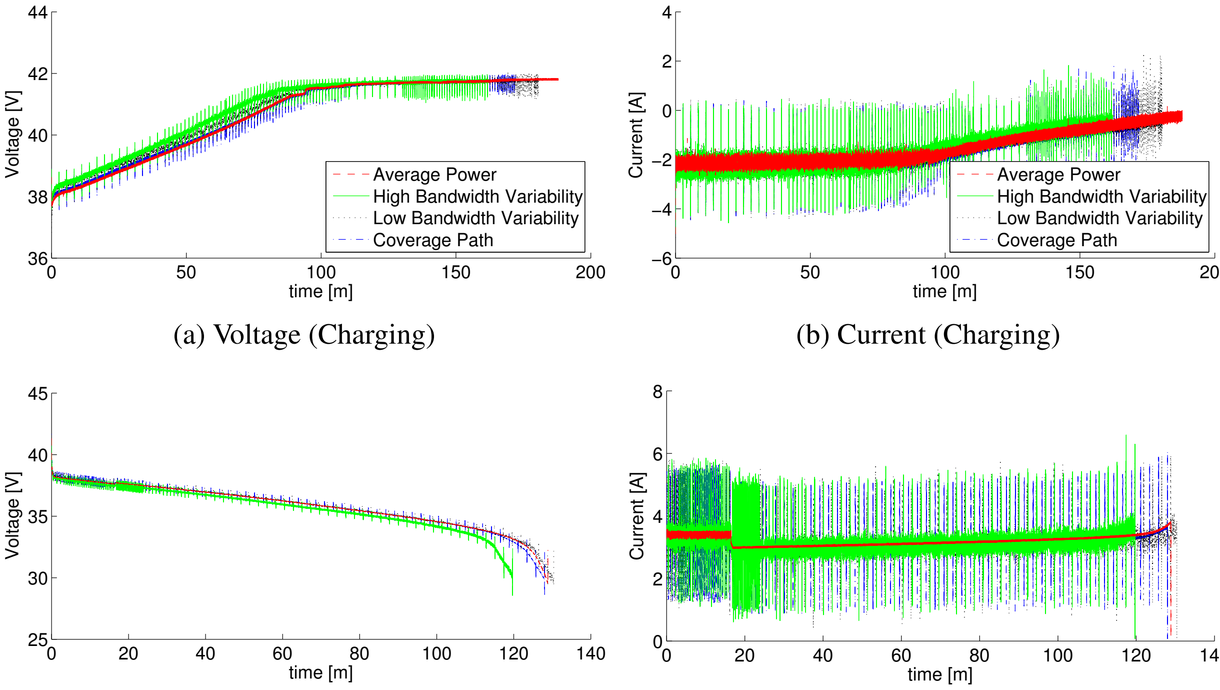
(a) Voltage (charging)

(b) Current (charging)

(c) Voltage (discharging)

(d) Current (discharging)

Figure 5.17: Comparison of averaged and variable power demand- first set



(a) Voltage (Charging)

(b) Current (Charging)

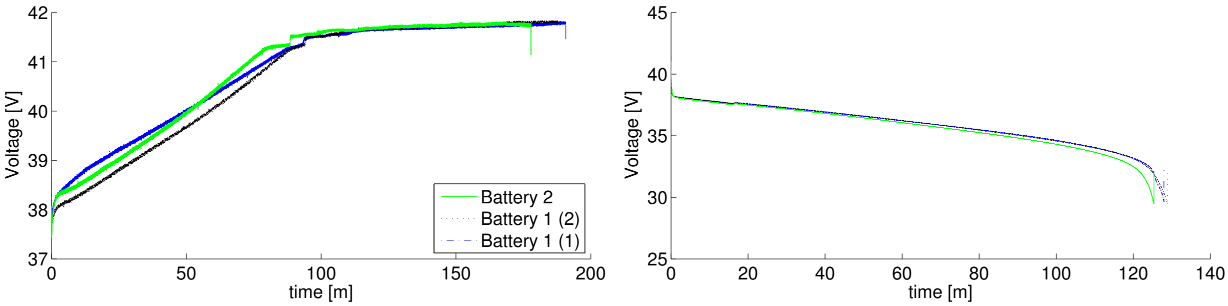
(c) Voltage (Discharging)

(d) Current (Discharging)

Figure 5.18: Comparison of averaged and variable power demand- second set

Table 5.2: Amp-hour during charge and discharge- second set

	Cov. Path	Low Band.	High Band.	Avgerage
Discharge	6.75	6.83	6.37	6.89
Charge	4.86	4.85	4.43	4.81



(a) Coverage path

(b) Average power demand

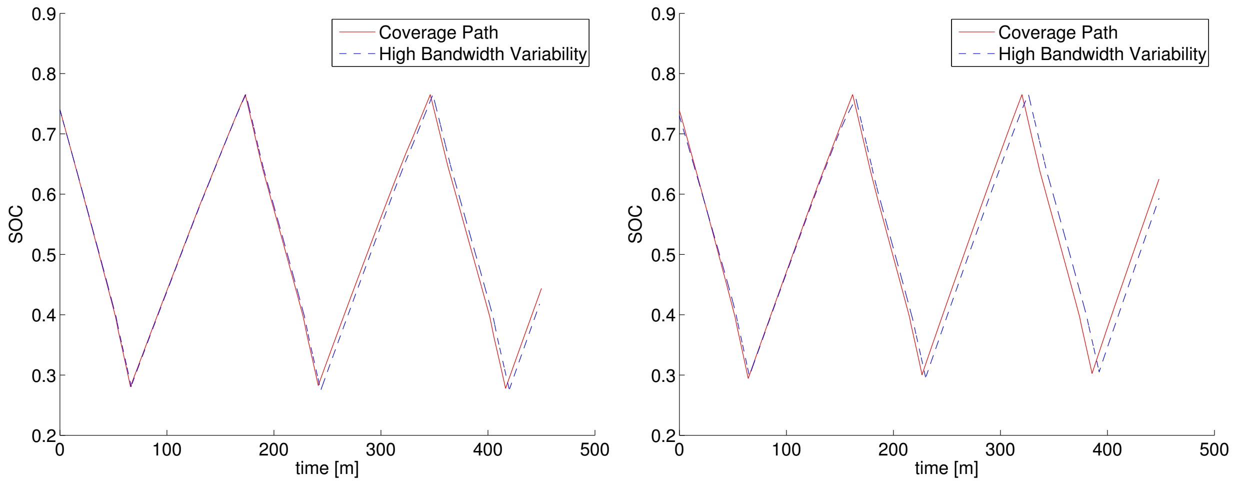
Figure 5.19: Comparing different batteries in same test

The second set of tests were conducted on a different battery pack of the same type. Figure 5.18 shows the voltage and current for the different tests and Table 5.2 shows the total Amp-hours drawn by the load bank during the tests. The values in brackets for the "Average" test case are for an additional test; the current draws are very similar between the different tests on the same battery. Once again, there is close agreement between the different power demands. The main exception is the discharge portion of the "High Bandwidth Variability" test; the battery is depleted faster in this test than the other tests. The discharge time is 7.1% less than the "Average Power" test and uses 7.5% less current before full discharge, but has 7.9% less current during the charging phase, accounting for the discrepancy in discharge time. The "Coverage Path" power demand discharges the battery in 0.66% less time and the "Low Bandwidth Variability" discharges the battery in 1.2% more time than the average power case.

Additional tests were conducted to compare the performance between the two different battery packs used in these tests. Figure 5.19 compares voltage from the "Average Power" tests for the different batteries for the charge and discharge phases and there is close agreement between the different batteries. This test was primarily conducted to verify that the optimization tests, described in the next section, could be run on the different batteries without changing battery parameters.

5.4.3 Validation of Optimization

To validate the optimization, we ran the optimization with the lead acid battery model to determine the optimal fuel cell switching times. This load can represent a constant power load for the



(a) First set

(b) Second set

Figure 5.20: Comparing measured battery SOC for different power demands using the SOC-limit controller

UGV, such as power for computing resources and communications. Figures 5.20 and 5.21 show experimental results from the SOC-limit controller and the optimization method, respectively. For both controllers, two different tests were run with "Coverage path" and "High Bandwidth Variability" power demand profiles. In both cases, we see close agreement between battery SOC for the different power demands.

In the SOC-limit case, we have two different sets of data, depending on which fuel cell was used for the test. The first fuel cell failed after completing the first set of tests and the tests were rerun with the new fuel cell to compare performance between the different fuel cells. The first set (Figure 5.20a) shows close agreement between the two trajectories, while there is more disagreement in the second set (Figure 5.20b). This difference is partially due to differences in initial conditions and a slightly delayed (about 30 s) fuel cell restart. Due to different fuel cell outputs, the data between fuel cells cannot be compared, however.

The fuel cell model also had to be slightly changed based on observations from working with the fuel cell. First, the fuel draws about 6 W while off, presumably to power control electronics on board. Also, the real power output for the fuel cell varies slightly from the published 200 W. One of the fuel cells tested averaged 193 W and the other fuel cell averaged 198 W. The average startup and shutdown times were also updated based on the recorded data. The variation in the fuel cell, both power output and the startup/shutdown time, is an important system variability that cannot currently be accounted for in the optimization model.

Figure 5.22 compares the energy used over the course of the mission. To compute energy usage, we use the weight of fuel consumed, scaled by the energy density of propane and the

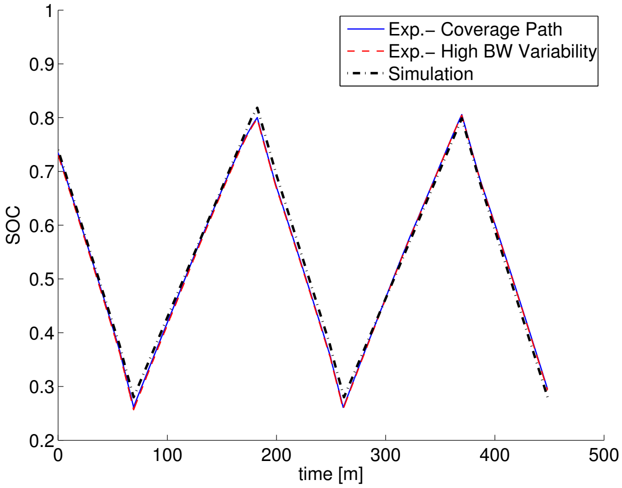


Figure 5.21: Comparing actual battery SOC with optimization model

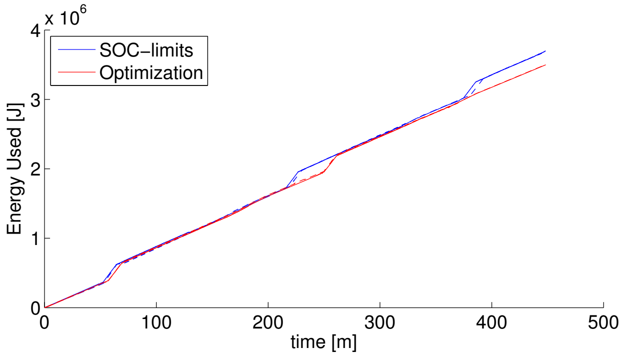


Figure 5.22: Energy usage for the SOC-limits and optimization tests (Solid line- “Coverage Path”, dashed line- “High Bandwidth Variability”

thermal efficiency of the fuel cell (14%), and the change in energy storage in the battery. The energy remaining in the battery for a given SOC is calculated using the battery model. Using the unscaled propane energy would change the scale of the plot would be changed dramatically and hide what the optimization is doing. This plot agrees closely with Figure 5.13b from simulation. This plot was determined using actual fuel consumption and battery SOC, while the simulation plots were calculated by integrating the power outputs from the battery and the fuel cell.

One additional test was run with the optimization controller, with a different power demand than previously considered. The average power demand over the entire mission was the same as the previous test cases, but the average power demand for sections of the mission varied by up to 50 W from the mission average. Figure 5.23 compares the “Varying Average” power demand to the baseline “Coverage Path” power demand. This is similar to a robot that runs on constant terrain for a period, then on a different surface with slightly different characteristics. Due to the large

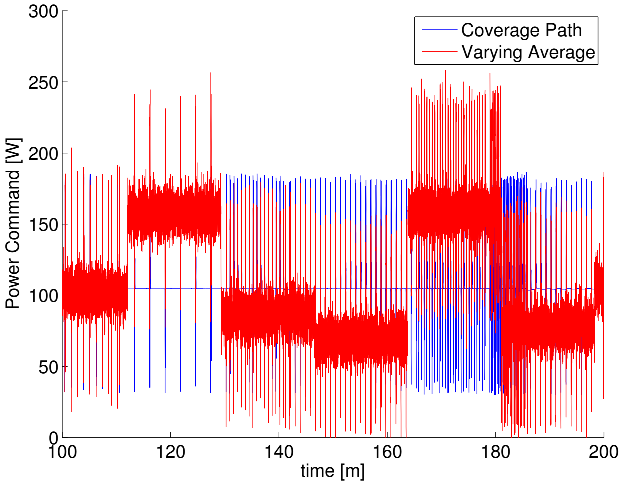


Figure 5.23: “Varying Average” power demand

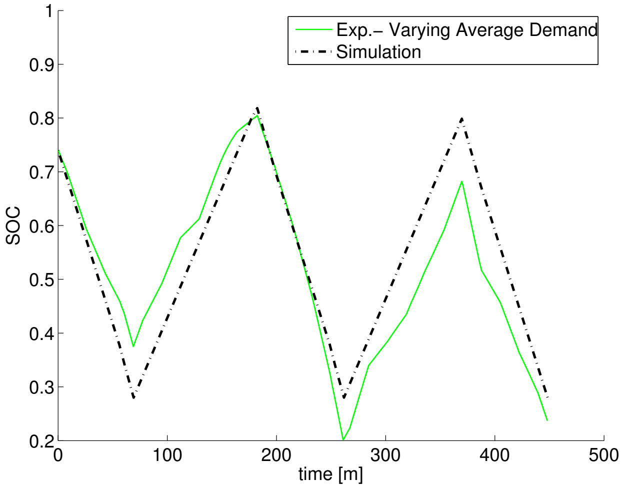


Figure 5.24: Comparing actual battery SOC for “Varying Average” power demand with optimization controller

differences over time, the battery SOC is expected to deviate from the baseline test. This test was an experiment to see how far the averaging assumption can be taken. As shown in Figure 5.24, there is a large discrepancy over time due to the unexpected power demands. Additionally, the battery charging limits are encountered in the first charging cycle (around 180 minutes), leading to a lower SOC at the end of the mission. This test shows the limits of the power demand assumption used in the optimization. If the moving power average differs greatly from the overall average, the battery can reach unsafe or inefficient operating conditions that the optimization routine cannot predict. In this case, the power strategy would need to be adapted during the mission based on measured power usage. This adaptation is left for future work.

5.5 Thermal optimization

Thermal modeling and control is another important part of battery operation. If the internal temperature of the battery gets too high, the battery will age more quickly and could potentially catch fire. In order to prevent this, the BB2590 shuts down and will not operate after the temperature exceeds a threshold. For a ground robot, this is a catastrophic failure as the robot would be inoperable until the battery cools down.

In addition to modeling the battery SOC, the BB2590 model discussed in Section 5.2.1 also tracks the internal temperature state of the battery. This allows us to experiment with different control techniques in simulation as well as augmenting the combined hybrid automaton model to include temperature dynamics. This model is used to prevent battery overheating and the optimization cost function is extended to allow tradeoffs between high temperature operation and energy efficiency.

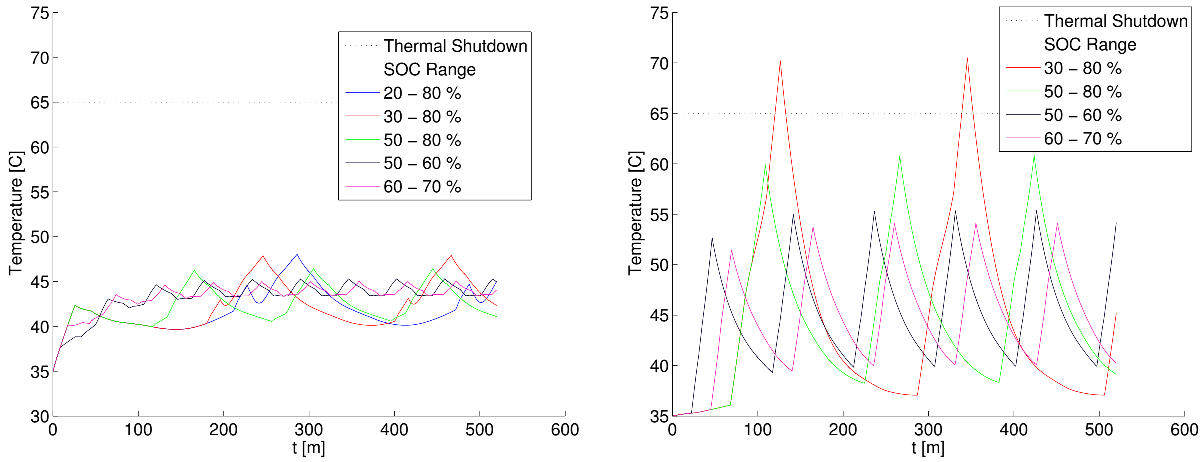
In particular, we are operating under the assumption that the mission power demand is known and cannot be modified. As such, our only control options are when to switch the fuel cell on and off. In contrast, existing algorithms for preventing battery overheating limit the power output of the battery at high temperatures [103]. By leveraging the hybrid power system with multiple components, we can maintain the desired system performance while maintaining the battery temperature in the acceptable range. We leave as future work to modify the mission power demand to control the battery temperature.

5.5.1 Thermal Behavior

First, we explore the thermal response of the battery in several different test cases. We use the constant power demand and SOC-based control for the fuel cell described in Section 5.3.2. The mission duration is also about 8.6 hrs. At the beginning of the mission, the battery temperature is 35° C, which is also the ambient temperature for the mission. Figure 5.25 shows the temperature over time for two different power levels: 40 W and 140 W.

In general, the temperature increases rapidly when the fuel cell is turned off and the battery is the only power supply operating. The battery heats up due to internal resistance and enthalpy effects, which is especially prominent at high currents and low SOC. When the fuel cell starts producing power, the battery heating decreases. For high power demands, the charging current is very small and the battery generates little heat due to operation and cools due to lower ambient air temperature. At low power demands, thermal rates are similar between charging and discharging phases, though charging produces less heat due to enthalpic characteristics.

The size of the peak depends on the power demand and how long the battery is operating under high demand. In the case of low power demand (40 W), there is little difference in maximum



(a) $P = 40$ W

(b) $P = 140$ W

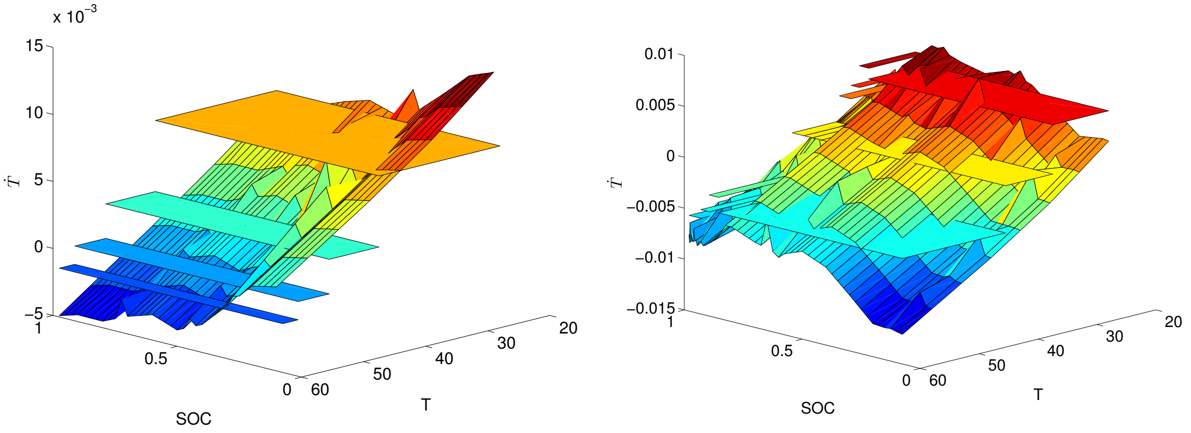
Figure 5.25: Comparing thermal response under different power conditions using a SOC-based control strategy

temperature at the different control strategies. None of the runs exceed the temperature threshold for thermal shutdown. In the case of higher power demand (140 W), the thermal threshold is exceeded when the more energy efficient (larger SOC range) is followed. While there are still large temperature spikes while operating at smaller ranges, the spikes are much smaller, though more frequent.

5.5.2 Thermal Modeling

Like the SOC dynamics, the temperature dynamics depend on current draws and the battery SOC. To adapt the combined model, we need to introduce a similar averaging technique for the temperature dynamics. However, there are several challenges. First, the temperature dynamics depend on both the current temperature and SOC and, second, the temperature dynamics can be both positive and negative in different SOC ranges for a given temperature and battery power draw.

To average the temperature dynamics, we introduce a set of states to represent different temperature levels. There are at least three states for each constant power draw, as shown in Figure 5.26. As explained later, a fourth state may be added based on the penalty temperature range. The "equilibrium" state, when $T_0 < T < T_1$, represents region of operation where the battery temperature is constant. To choose the thresholds T_0 and T_1, we look at the range of temperatures where the temperature dynamics are 0 for some SOC. While this does introduce some errors in the thermal models, it gives us sufficient information to track the temperature of the battery. When T is greater than T_1, the battery is cooling until it reaches T_1, then remains constant until the power level changes.



(a) Fuel cell off

(b) Fuel cell on

Figure 5.27: Temperature rate of change maps for area coverage mission

The main drawback to this model is that the simulated temperature will reach certain plateaus when it reaches the "equilibrium" state, while the actual temperature may still be increasing/decreasing. However, due to heat transfer to the ambient air being one of the largest factors, there is a strong tendency to return to the ambient air temperature over time. This averaged model implicitly uses this fact to make sure that the temperature error does not grow too large.

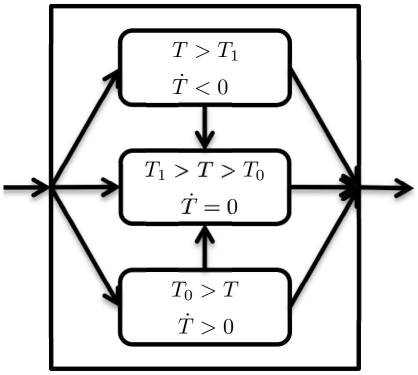


Figure 5.26: Multiple temperature states for a single power level

To adapt the combined model in Figure 5.9 to model temperature dynamics, each state in the model is replaced by a set of three states, differentiated by temperature. This results in a large increase in the number of states required to model the entire system. Figure 5.26 depicts the transitions within one power level. The transitions between different power levels depend on the relative values of T_1 and T_0 for the different states (e.g. fuel cell on).

To create the averaged dynamics for a single power level, first the map $\dot{T} = f(T, soc)$ is created. The possible temperature and SOC values are discretized and the battery model is simulated at each (T, soc) pair in the discretization. It is important to note that the ambient air temperature must be known and that the generated map $\dot{T}$ is dependent on the ambient temperature. In these plots, the ambient temperature is 35 C. Figure 5.27 shows the resulting map for the averaged area coverage mission with the fuel cell on and off. The flat planes show the averaged temperature dynamics and the temperature and SOC range over which those dynamics are used.

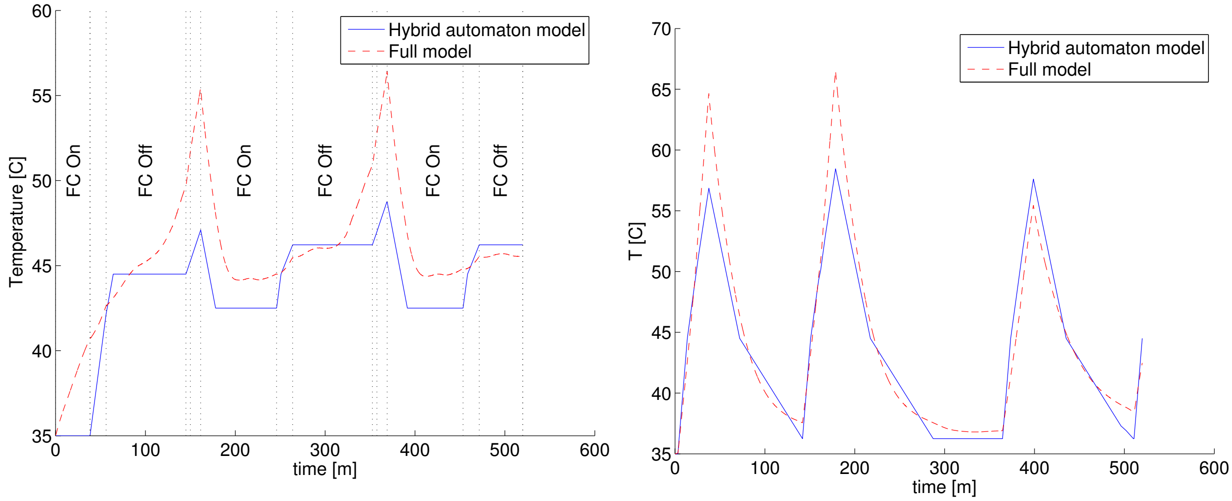
Note that in the discharge case, there are four average planes, representing four different temperature states. The orange plane corresponds to battery heating below the temperature penalty threshold (45 C), discussed in Section 5.5.4, while the next plane corresponds to battery heating above the threshold. For a battery beginning at the ambient temperature of 35 C, the state sequence would include both heating states before reaching the "equilibrium" state. The average dynamics are different in each case, as well as the cost function, which is discussed later. In the charging case, we still have different charging states based on the SOC charging profile, as discussed in Section 5.2.4. This process is repeated for each state in Figure 5.9 to create a full system hybrid automaton model with continuous variables representing temperature and battery SOC.

5.5.3 Simulation Comparison

To judge the effectiveness of the averaged temperature model, we simulate the full hybrid automaton model for two power demands: the area coverage mission with an average power demand of 75 W and a higher, constant power demand of 140 W. The ambient temperature for these tests is 35 C. Figure 5.28 shows the temperature over time for different power demands. To evaluate the performance of the hybrid automaton model, we also simulate the battery using the full nonlinear model. The results of these simulations are also shown in Figure 5.28. In both cases, the general shape of the temperature profile is modeled by the hybrid automaton model. However, there is significant underestimation of the peak temperature, especially in the area coverage mission in Figure 5.28a. Note that the peak occurs when the fuel cell is turned on and the temperature drops dramatically afterwards, mostly due to shedding heat to the ambient air.

There are two main reasons for this discrepancy. First, the temperature rises faster when the battery is at a lower SOC. This can be seen in Figure 5.27a. In the case when the fuel cell is turning on, it will usually be at a lower SOC; however, the model averages over all SOC. Second, the plateaus model constant dynamics. While there are times that approximation is valid (for example, at time 300m in Figure 5.28a), as the battery discharges, the battery begins to heat up faster. This effect cannot be modeled without introducing a new state division in the battery model based on SOC. Table 5.3 shows that the average temperature rate of change for the two different models. The startup states show the greatest underestimation of the temperature dynamics, though the fuel cell off state also underestimates the rate of change. However, the shutdown state overestimates battery heating, partially offsetting the underestimation of the off state.

Figure 5.28b shows the temperature modeling for a higher power demand mission. In this case, there is closer agreement between the different models. The main reason is that during discharge, the battery is always heating and there are no temperature plateaus. In the charging case, the temperature plateau only occurs at approximately the ambient temperature (35 C), matching the



(a) Area Coverage Path (75 W average), dotted lines denote change in fuel cell state

(b) Constant Power Draw (140 W)

Figure 5.28: Comparing optimization model and full battery thermal model for two different power demands

Table 5.3: Comparing average $\dot{T}$ in different states

State	FC On	FC off	FC Shutdown	FC startup 1	FC startup 2
Opt.	-0.05	0.01	0.27	0.15	0.16
Full Sim.	-0.08	0.06	0.07	0.38	0.32

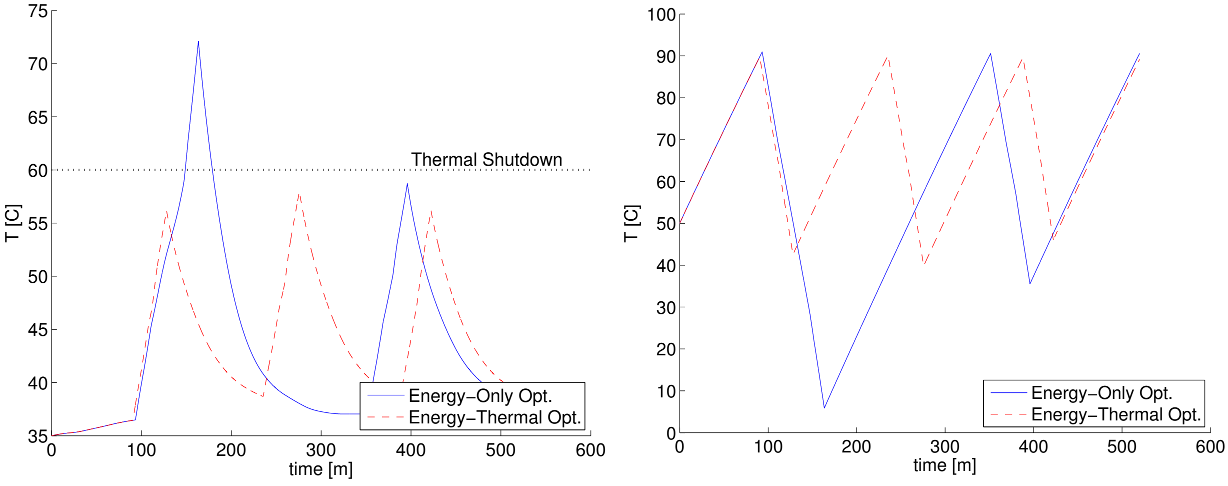
full battery model very well. Note that the underestimation of the peak temperature at time 50 minutes and 200 minutes is due to the battery being close to fully depleted at those times. The peak at time 400 minutes shows much closer agreement due to the battery not being as depleted.

5.5.4 Optimization Results

Using the above models, we can frame the optimization problem just as we did with the energy-only case. We expand the hybrid automaton model to include the continuous temperature state and the additional discrete states for distinguishing between different temperature dynamics. We update equation (5.1) to include temperature penalties as

$$J = \int_{t_0}^{t_f} J_c \, \mathrm{d}t = \int_{t_0}^{t_f} J_{fc} + J'_{batt} \, \mathrm{d}t \tag{5.7}$$

where $J'_{batt} = J_{batt} + P_{temp}$. J_{fc} and J_{batt} were defined in Sections 5.2.2 and 5.2.1 respectively. P_{temp} is a penalty on the temperature when the battery temperature exceeds a threshold and is



(a) Temperature

(b) Battery SOC

Figure 5.29: Comparing Energy-only and Energy-Thermal optimizations

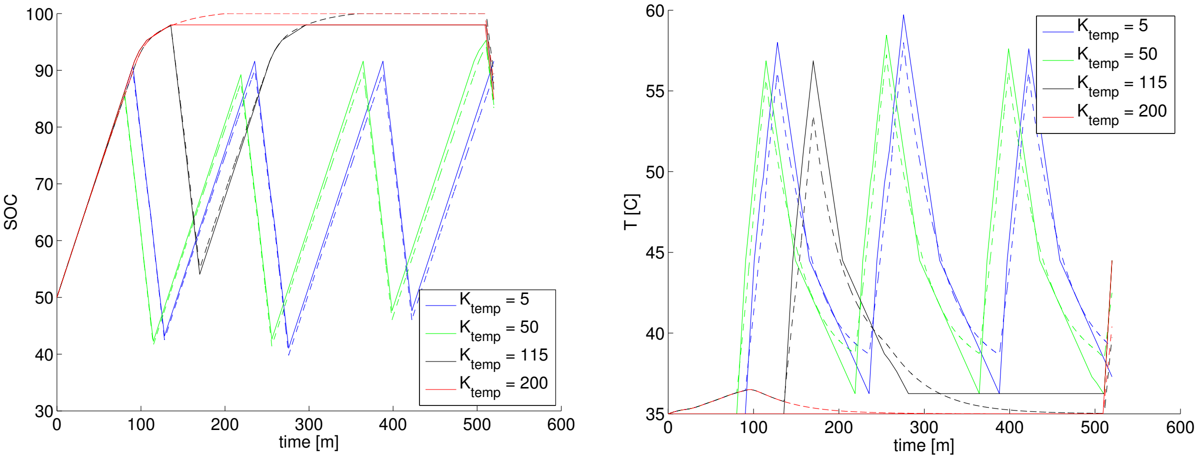
written as

$$P_{temp} = \begin{cases} K_{temp} & T_{batt} \geq T_h \\ 0 & T_{batt} < T_h \end{cases} . \tag{5.8}$$

Note that the states in Figure 5.26 might be further divided based on T_h to separate penalized and unpenalized temperatures. In this chapter, we used the value $T_h = 45$ C. This is taken from the battery specifications as the highest temperature where guarantees of battery cycles will be met. Exceeding this temperature will result in premature loss of capacity for the battery.

As before, the only control option that the optimization considers is when to switch the fuel cell on and off. This differs from previous battery thermal controls that impose limits on the current to prevent battery overheating [103]. By having the hybrid power system, we can leverage the different power components to complete the mission at the desired velocity without having to slow the UGV due to limited battery currents.

The resulting optimization problem is solved using the method described in Section 5.1.2.2 with some adjustments to speed up the process. Because the additional discrete states leads to an exponential increase in the number of sequences to test, the optimization takes significantly longer. As such, the full optimization algorithm from Section 5.1.2.2 was run once for $K_{temp} = 1$ and the discrete state sequences were recorded. Since many of the paths result in very similar cost values, state sequences were grouped with trajectories that differed by less than 0.001% in cost. Other values of K_{temp} were tested on one sequence from each group of trajectories. For several values of K_{temp}, additional sequences that were similar to the optimal or near optimal sequences were tested and did not improve the optimal cost for the mission.



(a) Battery SOC

(b) Temperature

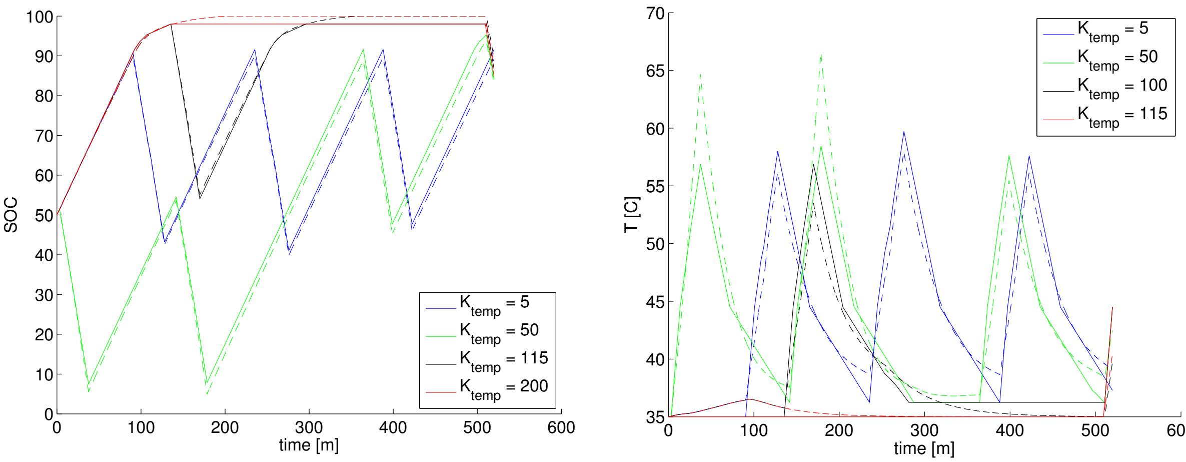
Figure 5.30: Comparing different values of K_{temp}, $P_d = 140$ (solid line- opt. model, dashed line- full model)

Figure 5.29 compares the results of the optimization for the $P_d = 140$ W case. The energy-only optimization only turns the fuel cell on two times, compared to the three times for the energy-thermal optimization, but results in a temperature that exceeds the thermal shutdown limit. Adding in the thermal model in this case results in extending mission life by preventing battery shutdown.

Operating the battery at high temperature leads to faster battery degradation. The previous example looked exclusively at preventing battery shutdown; here we look at preventing high-temperature operation of the battery by varying the cost parameter K_{temp}. We use the same power demands as used previously in this section (area coverage and constant 140 W demand curves).

Figures 5.30 and 5.31 show the battery SOC and temperature over time for the different values of K_{temp} for the 140 W power demand. In both of these figures, the discrete state sequences are the same and the difference is only in the timing of the controlled transitions. Additionally, the simplified model shows that the timings in both models are optimal for the given value of K_{temp}. However, in Figure 5.31 there are temperature spikes that cross the shutdown threshold of 60 C. The major differences occur in the $K_{temp} = 50$ case. The reason for the discrepancy between the optimization model and actual simulation was described on page 90 and is due to the lower battery SOC in that particular test. In this case, after the optimization is run, a simple test can compare several different optimal timings for the same sequence to determine which is optimal and conforms to the system constraints.

It is interesting to note that in some cases, the optimal response is to just leave the fuel cell on and use less than the full fuel cell power output. At this point, the cost of turning off the fuel cell, and spending some time in a high temperature region, is more than constant 60 W cost of



(a) Battery SOC

(b) Temperature

Figure 5.31: Comparing different values of K_{temp}, $P_d = 140$, also optimal timings, (solid line- opt. model, dashed line- full model)

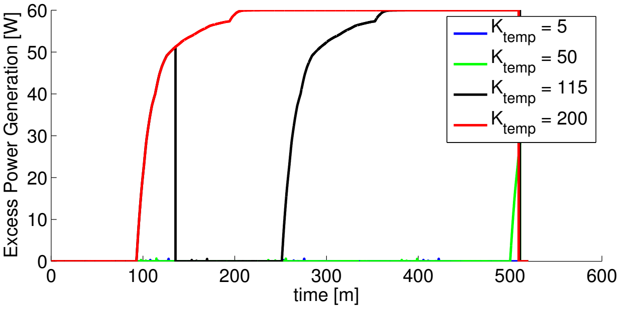
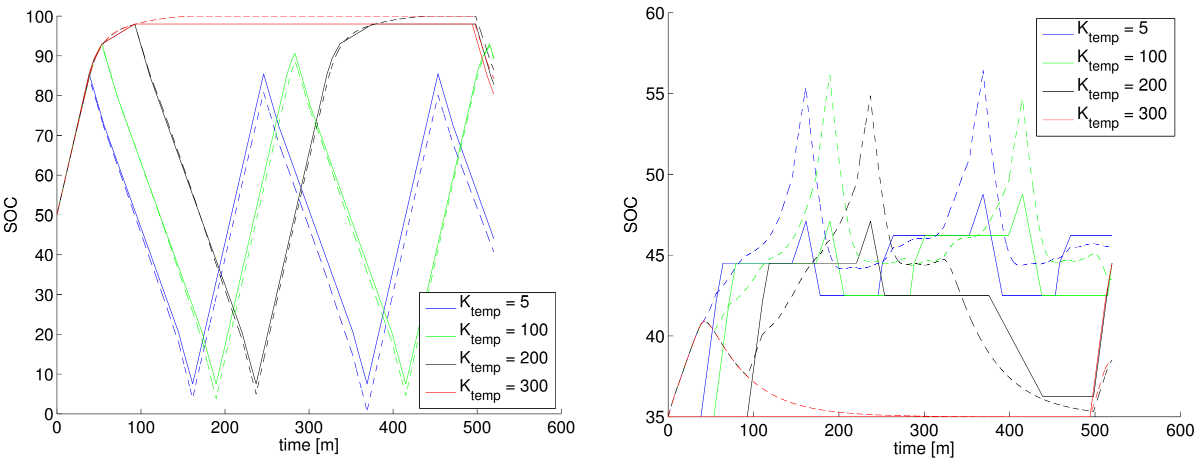


Figure 5.32: Excess power produced by the fuel cell, $P_d = 140$, for different values of K_{temp}

leaving the fuel cell on with the battery charged. Figure 5.32 shows the amount of fuel cell power output that is unused during the mission. As the value of K_{temp} increases, the amount of excess power produced increases. The most interesting case is the $K_{temp} = 115$. In this case, there is one switch off of the fuel cell. Due to the "equilibrium" thermal state, where the thermal state is approximated as constant, the estimated temperature plateaus at about 37 C, slightly higher than the initial battery temperature of 35 C. That difference is enough to prevent a second cycle of the fuel cell, even though there is enough time for a second cycle to be completed.

Figure 5.33 shows the results for the coverage demand profile. In this case, we get some of the same trends as previously. As K_{temp} increases, the battery is charged to a higher state before the fuel cell is switched off. Eventually, the fuel cell is left on when the penalty reaches a higher value.

It is important to note that there are large discrepancies between the optimization model and the full model. Because of the lower power demand in this mission, there is a very small chance



(a) Battery SOC

(b) Temperature

Figure 5.33: Comparing different values of K_{temp} using the coverage power demand (solid line- opt. model, dashed line- full model)

Table 5.4: Comparing time spent when $T > T_H$ for area coverage mission

K_{temp}	5	100	200	300
Opt Model	1.9e+04	1.7e+04	8.1e+03	0
Full Model	2.1e+04	2.2e+04	7.3e+03	0
Difference	-2.3e+03	-5.7e+03	8.1e+02	0

of reaching the shutdown temperature. While these large deviations are not ideal, they do not incapacitate the robot but can contribute to degraded battery performance over the lifetime of the battery.

Since the thermal penalty is engaged at the threshold T_H, it is important to quantify how accurately the model predicts temperatures in excess of the threshold. Table 5.4 shows the amount of time that the different models show the battery above the threshold. The optimization model is with 25% of the full model, though increasing K_{temp} from 5 to 100 results in an increase of time indicated by the full model. While this does case doubt as to the overall optimality of the solution when applied to the full model, the average temperature of the battery over the mission decrease from 45.7 C to 45.2 C when K_{temp} increases from 5 to 100, which accomplishes the goal of lowering the battery operating temperature.

5.6 Contributions

This chapter introduces a novel hybrid modeling framework for managing multiple power sources on a small ground robot (Innovation 3). A simplified model of power use over a mission can be used to optimize power source use. Application of this framework to a fuel cell/battery hybrid power system for a Packbot comprises Contribution 3. To optimize power generation, we introduce a simplification of the hybrid model of the different power system modes, with dynamics averaged in each discrete state. Through simulation, we demonstrate that the simplified model matches the full model closely.

Using the optimization algorithm, we plan operation of the fuel cell for a simulated coverage mission. Comparison to a simple, state-of-charge-limit controller is conducted showing that the optimal trajectory is able to complete the mission with greater energy reserves remaining in all test cases. The framework is also extended to model battery temperature during operation. The system is constrained to remain within safe operating temperatures and, through including temperature penalties in the optimization cost function, tradeoffs between energy loss and operation at high temperature are explored.

Optimization models for energy efficiency were validated with a hardware-in-the-loop system (Contribution 4). We compared battery performance under constant and variable power demands. Experimental results agreed closely with simulation results, confirming our averaging assumptions underpinning our hybrid system model used for optimization. Additionally, experimental tests using the optimization controller showed similar efficiency improvements compared to the baseline controller from the simulation results.

While our average power assumption was sufficient for the mission considered in this chapter, other missions may require a model allowing varying levels of power demand over the entire mission. This would allow missions that involve different modes of operation (periods of travel interrupted by periods of silent watching, for example) or different terrains that require different power demands (uphill vs. downhill, for example). We plan to analyze the performance of this method in missions with different durations and power demands. Additionally, the cost function can be expanded to include a desired state bounds both during and at the end of the mission. Future work is also required to incorporate fuel levels and tradeoffs between fuel consumed and ending state of charge.

CHAPTER 6

Modeling and Control of a Slipping UGV

Unmanned ground vehicles (UGVs) operate in many different, unstructured environments. They are widely used inside buildings, on hard road surfaces, as well as on softer off-road surfaces. Some environments are easy for the UGVs to traverse, asphalt roads for example, while some terrains pose significant hazards, such as frozen lakes, sand dunes, and slippery hillsides. In addition, UGVs can suffer from non-catastrophic failures that limit mobility and operation, such as manipulator arm failure [4]. Since many UGV operations take place where humans cannot safely and easily reach, simple failures can become catastrophic without ways to address the failures.

When operating in slippery conditions, large motor torques and turns at high velocity can introduce excessive wheel slip. Since longitudinal slip reduces the lateral friction (see [58] for a discussion of this topic), this slip can reduce maneuverability for the UGV. Also, the slipping UGV can execute a longer path, possibly colliding with obstacles the operator is trying to avoid.

This chapter presents two different controllers to limit wheel slip. First, we develop a dynamic model of a UGV with longitudinal and lateral wheel slip. This model is used in the development of two control schemes: a Supervisory Traction Controller based on trajectory planning and a sliding mode controller based on similar techniques for automobiles. The Supervisory Traction Controller is a switching controller that plans a trajectory to follow when excessive wheel slip is introduced using an optimization framework. This switching controller and the optimization cost function comprise Innovation 4. The sliding mode controller only considers longitudinal wheel slip, similar to automotive applications. If significant lateral slip occurs, this controller is not able to control the wheel slip, leading to loss of control. In simulation, we compare these controllers and demonstrate the ability to prevent loss of control due to wheel slip (Contribution 5). In this chapter, we assume that the command inputs are velocity commands coming from either a human teleoperator or an autonomous controller. While we are interested in command tracking, the motivation for this research is to prevent catastrophic situations for UGVs.

Related work on wheel slip and control is presented in Section 2.6. The slipping UGV model is derived in Section 6.1. Section 6.2 presents the supervisory traction control algorithm, including

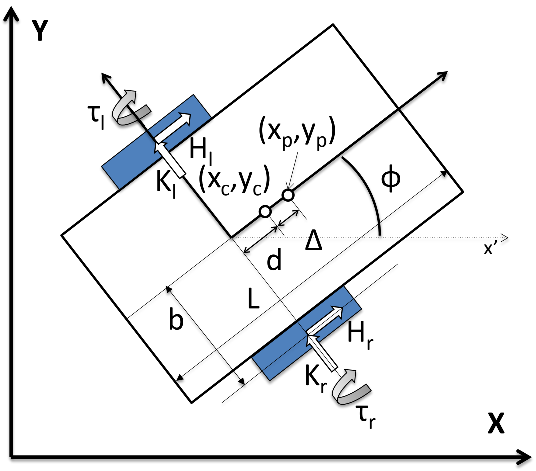


Figure 6.1: Free body diagram of a slipping UGV

the switching conditions, and the trajectory planning method. Section 6.3 derives the sliding mode controller for the slipping UGV. These two controllers are compared in Section 6.4 and conclusions are presented in Section 6.5.

6.1 Slipping UGV Model

For this research we consider a differentially-driven UGV, as shown in Figure 6.1. We use the following variables as states of the UGV, taken from [64]: (x_c, y_c) denotes the position of the center of mass of the UGV in the plane, ϕ denotes the heading angle of the UGV, $\theta_{r,l}$ denote the rotational angles of the wheels, $\delta_{r,l}$ denote the total distances the wheels have slipped in the longitudinal direction, and ζ denotes the total distance the wheels have slipped in the lateral direction. We assume the wheels are rigidly connected and both have a lateral slip velocity of $\dot{\zeta}$. The UGV is actuated through motors attached to the wheels; the wheel torques are denoted $\tau_{r,l}$. Figure 6.1 also shows (x_p, y_p), which is the center of mass of the platform, without the wheels.

We combine these UGV states into one state vector $q = [x_c, y_c, \phi, \theta_r, \theta_l, \delta_r, \delta_l, \zeta]^T$ and define the model input $u = [\tau_r, \tau_l]$. The ground friction forces are decomposed into $H_{r,l}$ and $K_{r,l}$, where H is the force in the longitudinal direction and K is in the lateral direction. The system parameters include the distance Δ from the center of mass of the platform to the center of mass of the UGV, the distance d from the axle to the center of mass of the UGV, the distance $2b$ between the two wheels, the height h of the center of mass of the UGV from the ground, and masses and inertias of the different bodies.

6.1.1 Dynamics

To derive the equations of motion, we use the Euler-Lagrange method. We model the UGV as three different rigid bodies: the platform and two wheels. Additionally we assume that the UGV is operating on flat ground. We can form the Lagrangian from the kinetic energy:

$$\begin{aligned} L = \;& 1/2m_p \left(\dot{x}_c^2 + \dot{y}_c^2 + \Delta^2\dot{\phi}^2 - 2\Delta\dot{\phi}\left(\dot{x}_c \sin\phi - \dot{y}_c \cos\phi\right)\right) \\ & + 1/2\left(I_p + 2I_m\right)\dot{\phi}^2 + 1/2I_w\left(\dot{\theta}_l^2 + \dot{\theta}_r^2\right) \\ & + 1/2m_w\left(\left(r\dot{\theta}_l - \dot{\delta}_l\right)^2 + \left(r\dot{\theta}_r - \dot{\delta}_r\right)^2 + 2\dot{\zeta}^2\right). \end{aligned} \tag{6.1}$$

There are 6 different forces acting on the system: lateral and longitudinal friction at each wheel and motor torques acting on each wheel. Since the lateral slip is in the same direction for both wheels, the lateral friction forces add to one lateral force. We define the generalized forces as

$$Q = \left[0, 0, 0, \tau_r, \tau_l, -H_r, -H_l, K_r + K_l\right]^T. \tag{6.2}$$

Using the Euler-Lagrange method, we can write the system dynamics in the form $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q,\dot{q}) = B(q)u$. Since the wheel friction terms are dependent on the velocity and not model inputs, the friction terms are included in $G(q,\dot{q})$. In addition, we must take into account the state constraints at the wheel contact points. Under the assumption of no slipping, the constraints relate the wheel velocities $\dot{\theta}_{r,l}$ to the UGV platform velocities $(\dot{x}_c, \dot{y}_c, \dot{\phi})$. By introducing slip, we now formulate the constraints as:

$$\dot{x}_c \cos\phi + \dot{y}_c \sin\phi + b\dot{\phi} = r\dot{\theta}_r - \dot{\delta}_r \tag{6.3}$$

$$\dot{x}_c \cos\phi + \dot{y}_c \sin\phi - b\dot{\phi} = r\dot{\theta}_l - \dot{\delta}_l \tag{6.4}$$

$$\dot{y}_c \cos\phi - \dot{x}_c \sin\phi - \dot{\phi}d = \dot{\zeta}. \tag{6.5}$$

Equations (6.3) and (6.4) connect the longitudinal wheel slip with the motion of the UGV. Equation (6.5) couples the lateral wheel slip with the motion of the robot. The first two constraints can be combined and integrated to form the holonomic constraint that couples the relative motion of the wheels to the change in heading of the UGV:

$$2b\phi - r(\theta_r - \theta_l) + \delta_r - \delta_l = C_1 \tag{6.6}$$

We can combine the constraints into one equation $A(q)\dot{q} = a(q)$. Using the method of [65], we create a vector of generalized velocities ν and create the state vector $x = [q, \nu]^T$. The dynamics of this system are:

$$\begin{aligned}\dot{x} &= \left[\begin{array}{c} S\nu + \eta \\ -(S^T M S)^{-1}(S^T M \gamma + S^T C \dot{q} + S^T G) \end{array}\right] \\ &\quad + \left[\begin{array}{c} 0 \\ (S^T M S)^{-1} S^T B \end{array}\right] u \end{aligned} \tag{6.7}$$

where $\eta(q)$ is a particular solution for the constraint equation $A(q)\dot{q} = a(q)$, $S(q)$ is a matrix whose columns span the null space of $A(q)$, and $\gamma = \dot{S}(q)\nu + \dot{\nu}(q)$.

6.1.2 Traction Model

The values of $H_{r,l}$ and $K_{r,l}$, the wheel friction forces, are calculated using the method outlined in [58]. The derivations for the right and left wheels are identical and the subscripts are dropped except where needed. This method assumes a pneumatic tire. The frictional force is a function of both the lateral (s_S) and longitudinal (s_L) wheel slip ratios [58]. To define the slip ratio, we first introduce the radial velocity of the wheel v_R and the inertial velocity of the wheel v_W. The radial velocity is based on how fast the wheel is spinning and is written as $v_R = r\dot{\theta}$. The inertial velocity is how fast the wheel is actually moving based on the velocity of the UGV platform and is written as $v_W = \sqrt{\dot{\zeta}^2 + (r\dot{\theta} - \dot{\delta})^2}$. We also define the side slip angle as $\alpha = \operatorname{atan}\left(\frac{\dot{\zeta}}{r\dot{\theta} - \dot{\delta}}\right)$. These equations convert the body-centric velocities to the velocity at each of the wheel-ground interaction points, allowing us to write the slip ratio for each wheel as:

$$s_L = \begin{cases} \frac{v_R \cos\alpha - v_W}{v_W}, v_R \cos\alpha \leq v_W \\ \frac{v_R \cos\alpha - v_W}{v_R \cos\alpha}, v_R \cos\alpha > v_W \end{cases} \tag{6.8}$$

$$s_S = \begin{cases} \frac{v_R \sin\alpha}{v_W}, v_R \cos\alpha \leq v_W \\ \tan\alpha, v_R \cos\alpha > v_W \end{cases} \tag{6.9}$$

The coefficient of friction μ is found using Burckhardt's formula as a function of the slip ratio, using values for different surfaces (see [58]). Figure 6.2 shows the friction curve for a wheel on asphalt. In this chapter, we will assume that the friction parameters are known.The main goal of traction control algorithms is to keep the wheel slip in the acceptable region. The values s_1^* and s_2^* denote the maximum and minimum wheel slip that produce the largest tractive forces. Moving out of this range leads to less traction and possible loss of control of the vehicle.

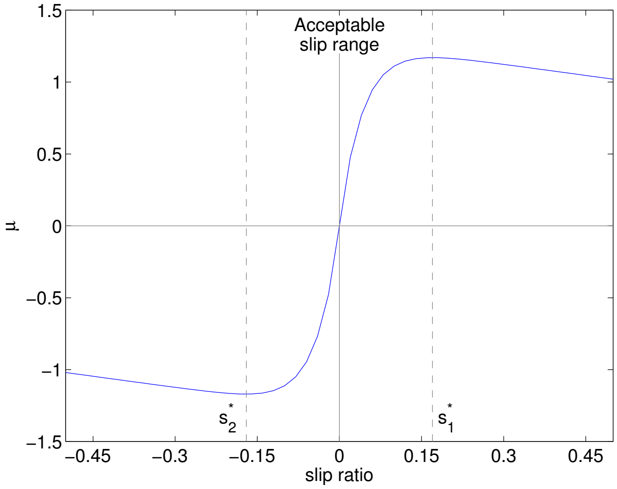


Figure 6.2: Friction coefficient μ as a function of wheel slip

The lateral and longitudinal forces are converted to the body frame as H and K. Since the friction also depends on the normal force N at the point of contact, we derive

$$N_l = \frac{mg\left(1 + \frac{h}{b}\mu_{K_r}\right)}{2 + \frac{h}{b}\left(\mu_{K_r} - \mu_{K_l}\right)} \tag{6.10}$$

$$N_r = \frac{mg\left(1 - \frac{h}{b}\mu_{K_l}\right)}{2 + \frac{h}{b}\left(\mu_{K_r} - \mu_{K_l}\right)}, \tag{6.11}$$

where $\mu_{K_{r,l}}$ is the coefficient of friction in the lateral direction for each wheel. The complete UGV model is coded up as a series of MATLAB functions to provide a simulation platform for a slipping UGV.

6.2 Supervisory Traction Control

To control both lateral and longitudinal wheel slip, we present the Supervisory Traction Controller. This controller is a switching controller that identifies situations where the UGV is or will be slipping. When wheel slip is detected, the controller switches from the nominal PID controller to a trajectory planning mode. The trajectory is planned using the slipping UGV model and is based on an optimization problem, combining wheel slip, commanded velocities, and a desired position derived from the commands.

We include the position in our controller, even though the commands are velocities, because fundamentally, all ground robot control is focused on positioning the robot; the position control loop is either handled in by a human (in teleoperation mode) or by an autonomous planner. This

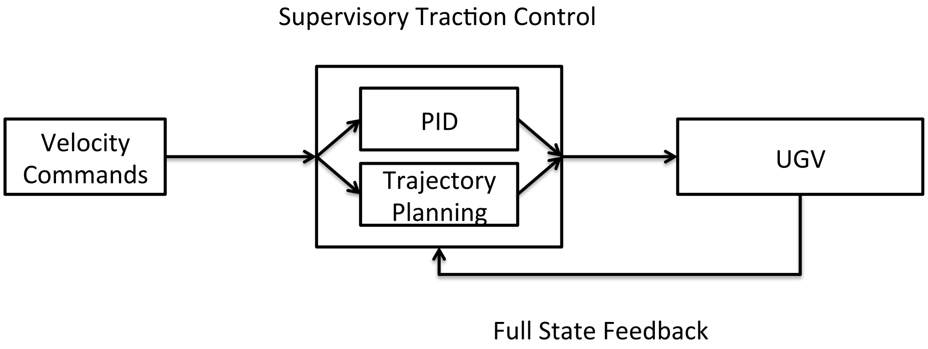


Figure 6.3: Block diagram of the supervisory control system

supervisory traction controller will improve the velocity following by adding a sense of position, reducing disturbances to higher controllers.

While we are taking UGV position into account while planning the trajectory, we only address the stability of the system with regard to velocity tracking and leave position tracking to higher level controllers. This controller will limit position errors caused by slipping, but will not track a path.

6.2.1 Switching Criteria

We define the following switching criteria to enter the trajectory planning mode:

- Lateral slip velocity $\dot{\zeta}$ exceeds some threshold: $|\dot{\zeta}| > \dot{\zeta}_{max}$
- Longitudinal slip velocity $\dot{\delta}$ exceeds some threshold: $|\dot{\delta}_{r,l}| > \dot{\delta}_{max}$
- Control input would exceed the ground reaction force: $|u_{r,l}| > \mu_{max} N_{r,l}$

When operating in PID mode, if any of these criteria are met, the controller switches to the trajectory planning mode. A full trajectory is planned and executed, then the switching criteria are reevaluated. If any of the criteria for slip are still met, the controller plans another trajectory. The controller returns to PID mode when the switching criteria are not met. Algorithm 6.1 shows the calculation of the control input.

Currently, while the controller is in trajectory planning mode, it ignores changes in the commanded velocities. When a new trajectory is planned, the controller uses the current velocity commands. Future work is required to determine when to replan the trajectory due to changes in commands.

Algorithm 6.1 Supervisory Traction Controller

```
$k = 0$
**while** $k \geq 0$ **do**
    Read states $x$
    Define $e = [v_d, \omega_d] - [v, \omega]$
    **if** None of the Switching conditions from Section 6.2.1 are met **then**
        Calculate PID command
        $v = K_P e + K_D \dot{e} + K_I \int_{t_o}^{t_k} e \, \mathrm{d}t$
        $u = [v_1 + v_2; v_1 - v_2]$
        (Convert from velocity to motor commands)
        Apply input $u$
        $k = k + 1$
    **else** *Trajectory Planning Mode*
        Choose $t_f$ and $(x_f, y_f)$
        Plan trajectory based on cost function (6.12)
        Determine inputs $u_k$ to $u_{k+t_f}$
        **for** $i = k \rightarrow k + t_f$ **do**
            Apply input $u_i$
        **end for**
        $k = k + t_f$
    **end if**
**end while**
```

6.2.2 Trajectory Planning

The planned trajectory must be feasible as well as limiting slip, converging to the velocity commands and limiting distance errors to the forecasted desired position (x_d, y_d) described below. This trajectory is planned over a designated time range $[t_0, t_f]$, with the time t_f defined below. We use the following cost function to penalize slip over the entire trajectory and errors in position (x_f, y_f), heading ϕ_f and velocity (v_f, ω_f) at $t = t_f$ from the corresponding desired values:

$$\begin{aligned} J = \; & K_\phi (\phi_f - \phi_d)^2 + K_d \left[(x_f - x_d)^2 + (y_f - y_d)^2 \right] \\ & + K_v (v_f - v_d)^2 + K_\omega (\omega_f - \omega_d)^2 + \int_{t_0}^{t_f} \nu^T P \nu \, \mathrm{d}t \end{aligned} \tag{6.12}$$

We can formulate P as $S^T \tilde{P} S$, where $\tilde{P}$ is a diagonal matrix that penalizes the slip states $\dot{\delta}_{r,l}$ and $\dot{\zeta}$.

Using this cost function, we can find an optimal trajectory that the UGV can traverse. We solve this optimization problem using the `fmincon` function from MATLAB, a numerical optimization technique.

In order to use this cost function, we need to determine the desired final position (x_d, y_d), heading ϕ_d and t_f. The final position is based on the commanded velocities by creating a circular arc originating from the starting location with the radius of the circle equal to $r = v_d/\omega_d$. The length of the arc is proportional to the error in velocity; larger velocity errors will have larger arcs. A length L is calculated as follows:

$$\begin{aligned} L_0 &= \max(e_v^0 P_v, e_\omega^0 P_\omega r) \\ L &= \min(L_0, L_{max}, \Psi_{max} r) \end{aligned} \tag{6.13}$$

where $e_v^0 = v_0 - v_d$ and $e_\omega^0 = \omega_0 - \omega_d$, v_0 and ω_0 are the initial velocities, P_V and P_ω are constants, and Ψ_{max} is the maximum angle of the arc. To determine the final time t_f, we calculate the nominal time $t_n = L/v_d$ to traverse the arc; t_f is calculated as

$$t_f = \left(L^2/P_t + 1\right) t_n. \tag{6.14}$$

where P_t is a surface-specific gain. This equation for t_f was determined by trial and error; linear or constant multipliers required more iterations of trajectory planning.

6.2.3 Stability Analysis

For the trajectory planning mode to be useful, the UGV needs to track the desired velocities, even when the controller stays in the trajectory planning mode for several iterations. To guarantee that the velocity errors are bounded, we argue that, with sufficiently large K_v and K_ω, the weighted norm of the velocity errors $||e|| = K_v e_v^2 + K_\omega e_\omega^2 \leq ||e_s||$, where $||e_s||$ is the norm at the beginning of the trajectory, $e_v = v_f - v_d$, and $e_\omega = \omega_f - \omega_d$. We first look at a simpler cost function by setting $K_d = K_\phi = 0$.

First, choose an input $u = u_0$ to the system (6.7) such that $||e_0|| = ||e_s||$. Such a trajectory can be constructed through a linearization of the the input-output system of u to $[e_v, e_\omega]$. The relative degree of this system is two and a full-state feedback can be designed by nonlinear analysis to match the goal condition. The input will not saturate because the only external force acting on the UGV is friction, which can only dissipate energy from the system. If the energy dissipated by friction would cause u to saturate, then the UGV would not be slipping and the controller would remain in PID mode. We can calculate $J_0 = ||e_0|| + \int \nu_0^T C \nu_0 \, \mathrm{d}t$. Since velocity errors are the same at t_0 and t_f, we can calculate $||e_s||$.

Next, suppose there is a different u such that $\int \nu^T C \nu \, \mathrm{d}t < \int \nu_0^T C \nu_0 \, \mathrm{d}t$. If we increase K_v and K_ω sufficiently, any possible increase in $||e||$ will more than off set the decrease in $\int \nu^T C \nu \, \mathrm{d}t$, making $J > J_0$. Thus, this trajectory cannot be optimal. Conversely, if there is another u such that

$\int \nu^T C \nu \,\mathrm{d}t > \int \nu_0^T C \nu_0 \,\mathrm{d}t$ and e_v or e_ω decreases over the interval, by increasing K_v or K_ω, this trajectory will have a smaller J.

The optimal input will be either u_0 or some u such that $||e|| < ||e_s||$. Creating a series of these trajectories one after another will be stable in the sense that $||e||$ will not increase and, if there exists a control that decreases $||e||$, with sufficiently large K_v and K_ω, $||e||$ will tend to 0.

For the original cost function (6.12), a similar analysis holds. Instead of only $\int \nu^T C \nu \,\mathrm{d}t$ being able to decrease, the terms $K_\phi(\theta_f - \theta_d)^2$ and $K_d\left[(x_f - x_d)^2 + (y_f - y_d)^2\right]$ can also decrease. We can make a similar argument that sufficiently large K_v and K_ω will result in $||e|| \leq ||e_0||$. The limiting values of K_v and K_ω will depend on the current friction values as well as t_f. We found that decreasing the values of t_f caused the velocity to converge to the commands faster.

6.2.4 Robustness Analysis

The trajectory planning and the switching criteria depend on knowing the friction coefficient at the current position. In a simulation environment, we know the parameters exactly. In a physical setting, the friction coefficient needs to estimated (see [70] for an example). Underestimating the friction coefficient will cause the trajectory planning mode to be entered when the UGV could actually execute the commanded torques without slipping. From Section 6.2.1, if $\mu_{est} < \mu_{max}$, a smaller torque will satisfy the switching condition. While the resulting trajectory would underestimate the mobility of the UGV and limit the performance, the trajectory would be feasible and the UGV could operate.

Overestimating the friction coefficient would delay triggering the switching conditions until the slip exceeds the threshold. Additionally, the planned trajectory would exceed the UGV's capabilities and would introduce more slip, which is what the controller is attempting to prevent. Therefore, when estimating the friction coefficient in a physical system, it is imperative to not overestimate the friction coefficient.

6.3 Sliding Mode Traction Control

Sliding mode traction control systems have been proposed for automotive applications [66–68]. There are several advantages to using sliding mode: the controller is robust to disturbances and can converge to a stable mode of the system in finite time [104]. Since the wheel-tire dynamics can vary widely over different surfaces and the dynamics of the system are very fast, these advantages of sliding mode controller make them attractive to control wheel slip.

There are several differences between the automotive applications in [66–68] and the UGV setting studied in this chapter. First, the command in an automobile is a torque input from the driver

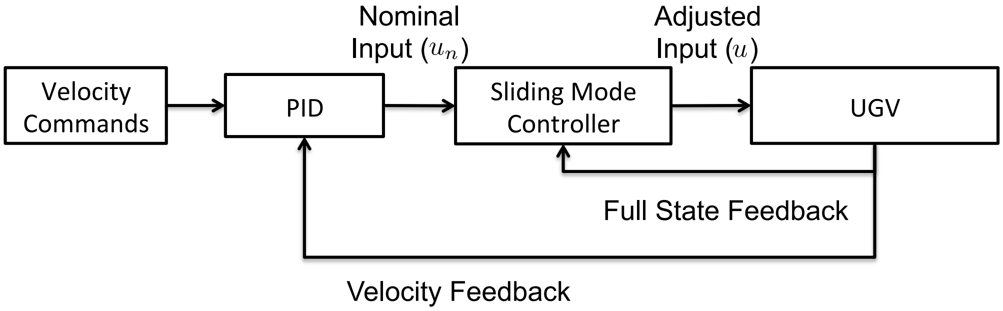


Figure 6.4: Block diagram for sliding mode traction controller

(brake pressure or engine power output). The controller augments this input to ensure traction. In the UGV setting, we are assuming velocity inputs; to bridge this gap, we introduce a PID controller to generate the nominal torque inputs; the sliding mode controller augments this command to ensure traction. Figure 6.4 shows the block diagram for the sliding mode controller to regulate the UGV velocity. For the following derivations, each wheel is treated independently and the subscripts r and l have been omitted. The derivations apply to both wheels.

To formulate the sliding mode controller, we need to define a sliding surface and dynamic equations for the sliding surface. In particular, since we are defining a traction and braking controller, we want to keep the longitudinal wheel slip s_L in a defined range. Figure 6.2 shows the acceptable region for s_L. To simplify the derivation of the sliding mode controller, we will assume that the lateral wheel slip is approximately zero. This follows the assumptions of the automotive traction control literature by only looking at longitudinal slip. Using this simplification, we can write

$$s_L \approx \tilde{s}_L = \dot{\delta}/\max(|r\dot{\theta}|, |r\dot{\theta} - \dot{\delta}|). \tag{6.15}$$

where $\tilde{s}_L$ is the longitudinal wheel slip without any lateral slip. Following the method presented in [68], we define two sliding surfaces L_1 and L_2 to mark the edge of this region:

$$\begin{aligned} L_1 &= s_L - s_1^* = 0 \\ L_2 &= s_L - s_2^* = 0 \end{aligned} \tag{6.16}$$

When $L_1 < 0$ and $L_2 > 0$, no extra control action needs to be taken to prevent wheel slip. When wheel slip exceeds the thresholds for maximum traction, the input is chosen to enforce $\dot{L}_1 < 0$ when $L_1 > 0$ and $\dot{L}_2 > 0$ when $L_2 < 0$.

To find $\dot{\tilde{s}}_L$, we have to consider forward and backwards motion of the wheel in addition to traction and braking cases. The traction case corresponds to acceleration and the braking corresponds to deceleration. Mathematically, these cases are based on the state derivative $\dot{\theta}$ and $\dot{\delta}$ and are written as follows:

1. Forward traction: $r\dot{\theta} \geq 0$ and $|r\dot{\theta}| \geq |r\dot{\theta} - \dot{\delta}|$
2. Backward traction: $r\dot{\theta} < 0$ and $|r\dot{\theta}| \geq |r\dot{\theta} - \dot{\delta}|$
3. Forward braking: $r\dot{\theta} - \dot{\delta} \geq 0$ and $|r\dot{\theta} - \dot{\delta}| > r\dot{\theta}$
4. Backward braking: $r\dot{\theta} - \dot{\delta} < 0$ and $|r\dot{\theta} - \dot{\delta}| > r\dot{\theta}$

For each of the four cases, we can derive $\dot{\tilde{s}}_L = f_i(\tilde{s}) + b_i(\tilde{s})u$. For the forward traction case ($i = 1$, $\tilde{s}_L = \frac{\dot{\delta}}{r\dot{\theta}}$), we get

$$\begin{aligned}\dot{\tilde{s}}_L = &\frac{\left(r\tilde{s}_L S_\theta - S_\delta\right)\left(S^T M S\right)^{-1}\left(S^T M\gamma + S^T C\dot{q} + S^T G\right)}{r\dot{\theta}} \\ &- \frac{\left(r\tilde{s}_L S_\theta - S_\delta\right)\left(S^T M S\right)^{-1} S^T B}{r\dot{\theta}} u\end{aligned} \tag{6.17}$$

S_θ and S_δ denote the rows of the matrix S, defined in Section 6.1, corresponding to θ and δ. Similar equations exist for the other cases $i = 2, 3, 4$. The constraint $L_1 > 0$ is possible only in the forward-traction and backward-braking cases; surface $L_2 < 0$ is only possible in the other cases.

Having found $\dot{L}_1 = \dot{L}_2 = \dot{\tilde{s}}_L$, we can choose the input u to converge to the sliding surfaces in finite time. By choosing $u = -b_i^{-1}(v + f_i)$, with i determined by the conditions listed above for the traction and braking cases, and v chosen as

$$v = \begin{cases} k & L_1 > 0 \\ -k & L_2 < 0 \end{cases},$$

wheel slip can be contained in the region $s_2^* \leq s_L \leq s_1^*$. By substituting u back into equation (6.17), we can verify that $\dot{L}_1 = -k$ when $L_1 > 0$ and $\dot{L}_2 = k$ when $L_2 < 0$. By the theory of sliding mode control, the system will converge to the sliding surfaces in finite time, keeping the wheel slip in the desired region [104]. In addition, if $L_1 \geq 0$ and the nominal user input $u_n < u$, the user is commanding a smaller input than is required to prevent excessive wheel slip. As such, the user command takes precedence and $u = u_n$. Similar assignment is made if $L_2 \leq 0$ and $u_n > u$.

Chattering is one drawback of sliding mode control, caused the discrete switching of the control input at the sliding surface. Using the method described in [66], a region of width ψ is defined around the sliding surface. The input v is changed as follows to maintain convergence while

Table 6.1: Simulation parameters

Parameter	mass	wheel radius	b	h
Value	14 kg	.3 m	.3 m	.25 m
Parameter	K_P	K_I	K_D	u_{max}
Value	$[8, 5]^T$	0	0	150 Nm
Parameter	K_V	K_ω	K_d	K_ϕ
Value	8/30[1]	8/20[1]	100/12.5[1]	10/60[1]
Parameter	$\tilde{P}$			
Value	$\mathrm{diag}([0, 0, 0, 0, 0, 40, 40, 65])$			

[1] These values used for large decelerations in angular velocity

avoiding the discrete change in input:

$$v = \begin{cases} k & S_1 > \phi \\ \frac{aS_1}{\psi} + \frac{b}{\psi} \int_0^t S_1 H(S_1) \mathrm{d}t & \psi > S_1 > 0 \\ \frac{aS_2}{\psi} + \frac{b}{\psi} \int_0^t S_2 H(S_2) \mathrm{d}t & 0 > S_2 > \psi \\ -k & S_2 < \psi \end{cases},$$

In this equation, $H(\cdot)$ is the unit step function. Within the region around the sliding surface, the control input takes the form of a PID controller. Note that because ϕ is chosen to be small, the gains for this PID controller are large.

6.4 Simulation Results

For the simulation, we used the parameters of a wheeled robot similar in size to an iRobot Packbot (See Table 6.1 for values), with friction parameters for running on snow [58]. We compare four different tests: longitudinal acceleration, circular trajectory, deceleration into turn, and acceleration out of turn. In all cases, we test the Supervisory Traction Controller of Section 6.2 and the sliding mode controller described in Section 6.3. Additionally, in the circular test, we compare a test with no traction control.

While the longitudinal test is similar to a scenario that would be encountered on an automotive system [66, 68], the other tests demonstrate the high maneuverability of ground robotic systems. In particular, UGVs can make small-radius turns, even turning in place. We consider the other test cases to demonstrate that limiting wheel slip for turns at higher speeds requires controlling both lateral and longitudinal wheel slip.

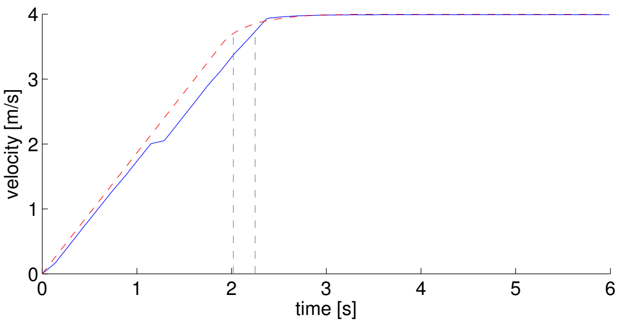


Figure 6.5: Comparing forward velocity for straight acceleration

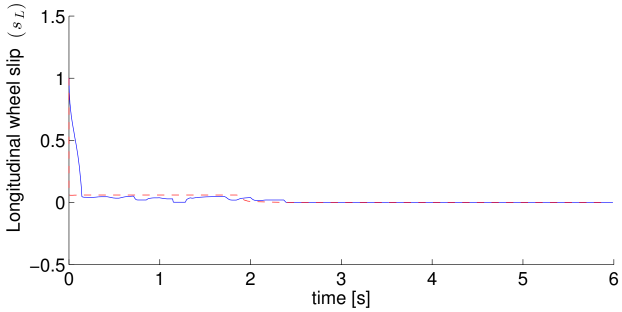


Figure 6.6: Longitudinal wheel slip s_L for straight acceleration

6.4.1 Longitudinal Acceleration

The first test compares the sliding mode controller and the supervisory traction controller in a straight-line acceleration test. The UGV is commanded to drive at a constant desired velocity of $[v_d, \omega_d] = [4\ \mathrm{m/s}, 0\ \mathrm{rad/s}]$ on snow. Initially, both wheels are slipping at 0.045 m/s and the UGV is stationary. This is very similar to applications for automobiles and we expect the sliding mode controller to work very well in this case. Figure 6.5 compares the velocity over time for this case. The dashed vertical lines show when traction controllers are no longer needed to prevent wheel slip in the respective cases: the Supervisory Traction Controller reenters the PID mode at $t = 2.2$ seconds and the sliding mode controller stops augmenting the PID commands to control slip at $t = 2$ seconds. In both cases, the UGV is traveling at 3.7 m/s. In this test the Supervisory Traction Controller takes 10% longer to reach this velocity, compared to the sliding mode controller.

Due to the initial conditions, with the UGV stationary and the wheels slightly turning, $s_L = 1$ at the beginning of the scenario. As shown in Figure 6.6, both controllers are able regulate the wheel slip to the desired range of slip values. For the snow conditions used in this simulation, $s_1^* = 0.06$. The sliding mode controller is able to find the wheel slip with the largest acceleration. Since the UGV is traveling straight, there is no side slip introduced and the approximation for longitudinal wheel slip is exact. While the lookahead parameters of the Supervisory Traction Controller could

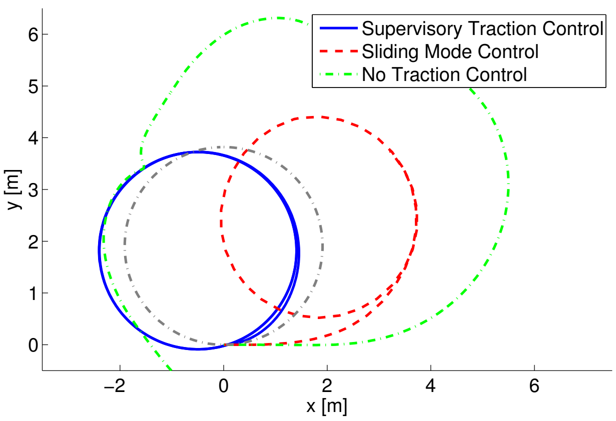


Figure 6.7: Supervisory Traction Controller closest to inferred path (gray circle)

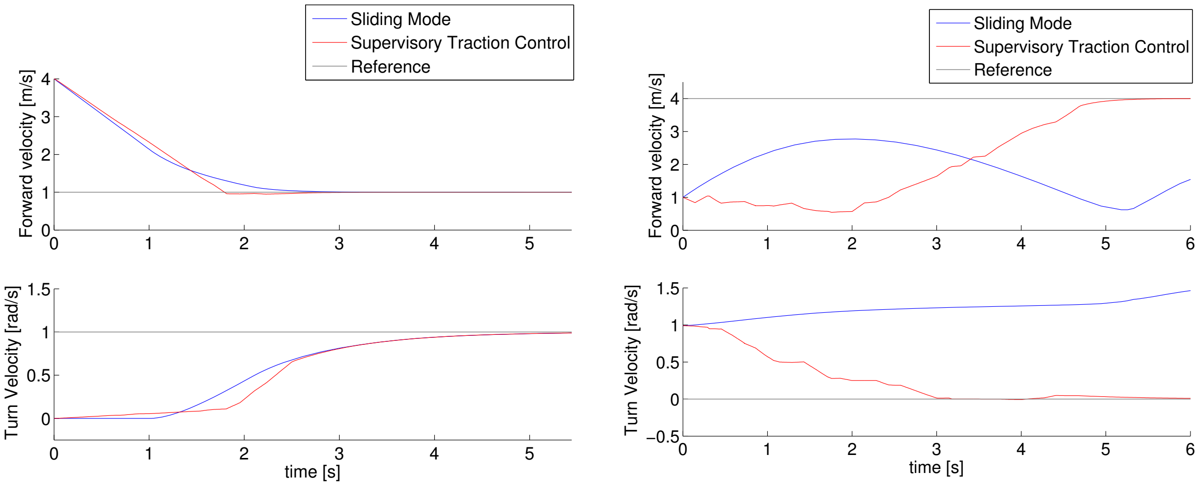
be adjusted so that the performance of the two controllers matched each other, these parameters would be terrain dependent. The sliding mode controller would operate well in all terrains without needing to tune parameters.

6.4.2 Circular Trajectory

In this scenario, the UGV is commanded to drive at a constant desired velocity of $[v_d, \omega_d] = [2\text{ m/s}, \pi/3\text{ rad/s}]$ on snow. Initially, both wheels are slipping $\dot{\delta}_{r,l} = 0.045m/s$ and the UGV is stationary. Figure 6.7 compares the sliding mode controller, the Supervisory Traction Controller, and a baseline without traction control and shows the $x - y$ trajectory for each controller. Without traction control, the UGV loses traction and starts to slide on the surface. Using the sliding mode controller, the UGV completes a circle with moderate displacement away from the desired path from the user input. Using the supervisory traction controller, a circular trajectory is completed near the origin. While the initial slipping was within the switching limits, the initial torques calculated by the PID controller would have exceeded the ground reaction forces and the trajectory planning mode was activated.

If the UGV exactly follows the commanded velocity from the beginning of the simulation, it would complete a circular path that returns to the origin. While the commanded velocity is reached with both controllers, the Supervisory Traction Controller regulates the path of the UGV to follow more closely this inferred path from the user commands. The PID controller in the sliding mode control architecture could be tuned to give a similar response to the traction controller. However, to make the sliding mode controller robust to different surfaces, distinct tunings would be needed for every surface.

In both the Supervisory Traction Controller and the sliding mode controller, the wheel slip is limited and the UGV maintains good control with the ground. For the Supervisory Traction



(a) Deceleration into turn

(b) Acceleration out of turn

Figure 6.8: Velocity and angular velocity for combined change in forward and angular velocities

Controller, this is a direct consequence of the controller design; both lateral and longitudinal slip are considered in the optimization step. For the sliding mode controller, the amount of side slip is limited and does not introduce errors in the wheel slip equations.

6.4.3 Combined Acceleration and Turning

Here we consider two different velocity commands. In the first case, called *deceleration into turn*, the commanded velocity is $[v_d, \omega_d] = [1, 1]$ and the UGV is initially traveling straight forward at 4 m/s with no wheel slip. In the second case, called *acceleration out of turn*, the command and the initial conditions are reversed; the UGV is commanded to travel straight forward at 4 m/s and is initially traveling at 1 m/s and turning at 1 rad/s. Figure 6.8 compares the forward velocity and angular velocity for these different cases.

For the deceleration into turn test, both controllers are able to regulate the velocities to the commanded values in fairly similar manners. The Supervisory Traction Controller decelerates faster before beginning the turn, but the two controllers take about the same amount of time to meet the commands. In the acceleration out of turn test, only the Supervisory Traction Controller is able to drive the UGV to meet the command. The sliding mode controller initially accelerates the UGV while still turning, causing the lateral wheel slip to become significant and resulting in loss of control of the vehicle. Figure 6.9 shows the lateral wheel slip becoming significant for both wheels, which causes the sliding mode controller to not be able to control longitudinal wheel slip.

In comparing these two situations, we see that only the Supervisory Traction Controller is able to complete both operations without introducing significant lateral wheel slip. This is accomplished

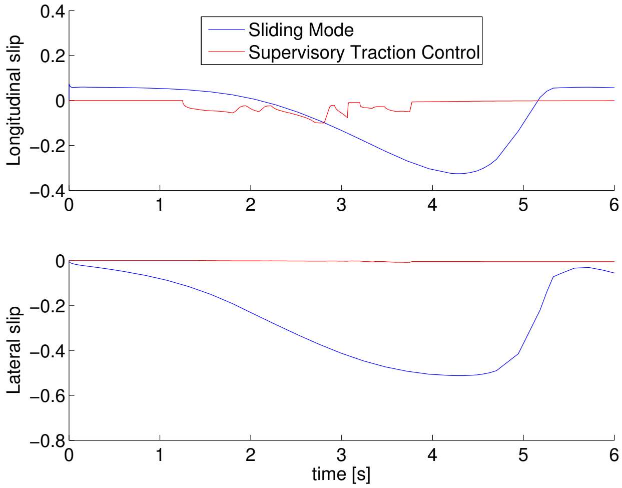


Figure 6.9: Longitudinal and lateral wheel slip for both wheels under sliding mode control for acceleration out of turn

by penalizing both longitudinal and lateral wheel slip in the optimization. With sufficiently large slip penalties in the cost function, the UGV could even take infeasible commands and find the best UGV performance without losing control. The sliding mode controller, however, has to have additional logic to prevent lateral wheel slip. In the case presented here, the PID could be retuned to prevent loss of control on the deceleration case; however, that returning does effect the response for the deceleration into turn case, resulting in lateral slip in that case as well. While the sliding mode controller is easier to implement on an actual UGV, an additional controller needs to be designed to prevent lateral slip.

6.5 Contributions

This chapter developed a dynamic model for a slipping UGV, using the Lagrangian approach. Using this model, we presented two controllers. The Supervisory Traction Controller is proposed which switches from a PID law to a trajectory planning mode when excessive wheel slip is or will be encountered. In this trajectory planning mode, we trade off commanded velocity and the desired position based on velocity commands using a novel cost function for optimization (Innovation 4). A sliding mode controller, similar to controllers used in automotive applications, is derived using the equations of motion for the slipping UGV. These two controllers are compared in simulation on snow. In some cases, the controllers operate comparably, while in other cases, the sliding mode controller is unable to compensate for the lateral slip of the vehicle and is unable to control the system. The slipping UGV model and these two controllers constitute Contribution 5 of this dissertation. The sliding mode controller, while easier to implement in a physical system, requires additional logic to prevent lateral slip. The Supervisory Traction Controller limits both lateral and longitudinal slip, but the trajectory planning step requires additional work to be feasible in a physical system.

In future work, we plan on implementing this controller on a physical system. To model slopes and other terrain features, a gravity vector will be added to the model. In addition, we can design a path planner that utilizes the predicted location information provided by the supervisory traction controller. For the sliding mode controller, we can look into a more complex controller to translate the velocity commands into torque commands. This controller would be responsible for limiting lateral wheel slip, allowing the sliding mode controller to operate as designed to control longitudinal wheel slip.

With the Supervisory Traction Controller designed as a switching controller, additional modules can be added to deal with different situations. For example, if the UGV becomes immobilized due to the wheels digging into the ground, a different controller can be activated that will try different techniques to extricate the UGV from the situation. Other controllers for different mobility faults, such as motor or track failure, can also be designed.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation, we discussed three control methods that increase the operational capacity of unmanned ground vehicles (UGVs). The first two methods increase the energy efficiency of the UGVs, allowing for longer missions for a given onboard energy storage. The third method focuses on preventing loss of traction and maintaining maneuverability of the UGV. These methods can be integrated together to foram a more robust overall system for planning and regulating energy and mobility systems.

7.1.1 Energy Characterization Contributions

Understanding energy use is important for being able to plan energy efficient operation of the robot. In Chapter 3, we presented a methodology for characterizing energy usage for a mobile ground robot. No such standardized methodology existed previously. This methodology entails running the robot in a straight line or rounded box path at different speeds and on different terrains. From these tests, we record the energy used per distance traveled and the electrical-to-mechanical efficiency. This data can be used to produce more efficient path plans and better estimate remaining transversal range.

We demonstrated this methodology on an iRobot Packbot. We conducted tests at five different velocities on asphalt and soft grass terrains. We found that the robot required less energy when traveling on asphalt than on soft grass. On the soft grass, the energy requirements decreased with increasing speed, while the same trend was not present on asphalt. The electrical-to-mechanical efficiency increased with speed up to the maximum speed allowed by the Packbot, but did not depend on the terrain. Additionally, we ran both rounded box and straight-line tests for a hard grass terrain and found similar results to the asphalt tests over multiple traversal speeds.

7.1.2 Area Coverage Contributions

Using this energy usage characterization, we extended existing path planning techniques for an area coverage mission, adding a trajectory generation step that optimizes the path and traversal speeds over the coverage path. First, we use the Boustrophedon Decomposition path planning method to find a set of waypoints and an obstacle free path that would cover the region. We then used an optimal control framework to plan a trajectory that would follow the path, trading off coverage mission time for energy usage. Augmentation of coverage planning with an energy and time-aware optimization represents an innovation of this dissertation.

By varying the values of the gains in the cost function, we are able to quantify tradeoffs between the area covered, force inputs, and most efficient motor operation. This is demonstrated in three different cases: motor efficiency excluded from cost function, motor efficiency peak at the maximum velocity, and motor efficiency peak below the maximum velocity. Simulation results indicated that up to 25% of the energy required could be saved by lengthening the time required to travel the coverage path distance in the case where motor efficiency peaks below the maximum velocity of the UGV. We also introduced a moving turn modification to the coverage trajectory that reduced the time and energy required to traverse the coverage path.

7.1.3 Hybrid Power Modeling Contributions

To integrate multiple power sources on a UGV, we presented a novel hybrid modeling framework for managing multiple power sources on a small ground robot. This framework combines hybrid models with continuous and discrete dynamics of each component to model the complete power system. By assuming that power use over the mission can be approximated by the average power draw, a simplified model is used for optimization. The specification framework and model simplification contributions of this ware also represent innovations.

Using the optimization algorithm, we schedule operation for a battery/fuel cell hybrid power source. Comparison to a simple, state-of-charge-limit controller is conducted showing that the optimal trajectory is able to complete the mission with greater energy reserves remaining in all test cases. The framework is also extended to model battery temperature during operation. The system is constrained to remain within safe operating temperatures and, through including temperature penalties in the optimization cost function, tradeoffs between energy loss and operation at high temperature are explored. The development of the battery/fuel cell power system model and the accompanying simulation results represents a contribution of this dissertation.

These simulation results were extended by validating the proposed framework in a hardware-in-the-loop experimental setting. Battery performance under constant and variable power demands showed little difference, validating the averaging assumption used in the simplification process.

Implementing the optimization controller in the hardware system yielded the same 5% decrease in energy usage compared to the baseline controller, as reported in simulation. This suite of hardware-in-the-loop validation experiments represents another contribution of this work to the UGV power systems community.

7.1.4 Slipping UGV Contributions

To prevent loss of traction, we presented two different control algorithms to control wheel slip. These controllers are based on a dynamic model for a slipping UGV, derived using the Lagrangian approach. The Supervisory Traction Controller is proposed which switches from a PID law to a trajectory planning mode when excessive wheel slip is or will be encountered. In this trajectory planning mode, we trade off commanded velocity and the desired position based on velocity commands using a novel cost function for optimization. The switching control structure for slip control with associated optimization cost function represent an innovation of this work. A sliding mode controller, similar to controllers used in automotive applications, is derived using the equations of motion for a slipping UGV. This controller is the baseline comparison for the Supervisory Traction Control algorithm.

These two controllers are compared in simulation on snow. In some cases, the controllers operate comparably, while in other cases, the sliding mode controller is unable to compensate for the lateral slip of the vehicle and is unable to control the system. The slipping UGV model and these two controllers constitute a contribution of this dissertation.

7.2 Future Work

The contributions and innovations of this dissertation are only the beginning for improving reliability and expanding the operational capacity for autonomous and teleoperated UGVs. There are many other topics that need to be addressed before autonomous UGVs become consistent fixtures in our lives. Faults and failures will certainly occur. As humans, we can reason and find ways to compensate for these faults. UGVs need to have similar reasoning capabilities. For example, if a UGV starts to slip and digs itself into the terrain, the UGV needs to find maneuvers to enable an escape. Humans regularly perform these tasks for cars and other machinery that become stranded or stuck.

Area coverage planning in this work still depends on an initial path planning stage to form the basic coverage path. We are only able to introduce one small improvement on the path through a moving turn in addition to selecting traversal speeds. There are many different ways that the path could changed to increase energy efficiency; limiting the number of turns, planning paths with gentle turns, and prioritizing areas to cover the largest contiguous areas first. Additional analytical and experiment experimental work is needed to compare actual energy usage for the coverage mission for the different tradeoffs discussed in Chapter 4.

Hybrid power sources for UGVs are still very limited. The framework presented here is one of the first attempts to control power sources scaled for small UGVs. There are still many ways that this framework can be expanded. Here, we only consider missions with similar power demands throughout the entire mission. Other missions have regions with dissimilar power demands, such as a persistent stare mission, and the framework can be easily extended to these types of missions. Additional power sources can be added to the framework, giving more control options for the optimization algorithm.

While the methods in this dissertation work well for the battery/fuel cell system, the limitations of the fuel cell motivated many of the assumptions and simplifications used in this dissertation. As new technologies for powering UGVs emerge, they will pose new and different challenges. Some systems will conform well to the optimization presented in Chapter 5 while others will require new approaches if averaging assumptions do not reflect the actual operation of the devices. To understand where this framework is most applicable, systems with more continuous inputs need to be studied to see how closely these approximations represent actual system performance. Other possibilities include controlling a subset of the power sources, such as a battery and ultracapacitor, using higher fidelity models, then adding the controlled system as a power component in the overall power management framework. All of these techniques add promise to greatly increase UGV range and mission life.

For UGV wheel slip, future work includes validation of the slipping UGV model used to derive the control algorithms. This involves instrumenting a robot to acquire accurate ground velocity measurements in order to calculate wheel slip. This platform can also be used to implement the proposed controllers. Currently, the optimization step of the Supervisory Traction Controller takes an infeasible amount of time to run. In order to implement this controller, a look up table of planned trajectories is required to achieve real-time performance. For the sliding mode controller, a more complex intermediate controller is required to account for lateral wheel slip. This controller would prevent high-speed turns that would introduce lateral slip and allow the sliding mode controller to maintain wheel traction.

One major enhancement to this work would be the real time updates for trajectory and power planning. Currently, these algorithms create a plan for an entire mission. In reality, during operation of the system, new information is acquired, prompting the need for plan updates as performance or the mission deviates from average or nominal trends. This requires developing online versions of these algorithms capable of incorporating this new information into plan updates. This also includes replanning the desired mission plan, e.g., coverage path, if, for instance, the online feedback shows that the energy stores are insufficient for the current plan.

With continued research, autonomous ground robots will continue to advance and develop. The topics discussed in this dissertation address a few of the current challenges that limit the operation capacity; expanding this ideas of energy and mobility management will lead to platforms with longer mission duration and more reliable operation on slick and dangerous surfaces. This will allow human operators to exploit capable autonomous systems for dull, dirty and dangerous tasks, saving and enriching human lives.

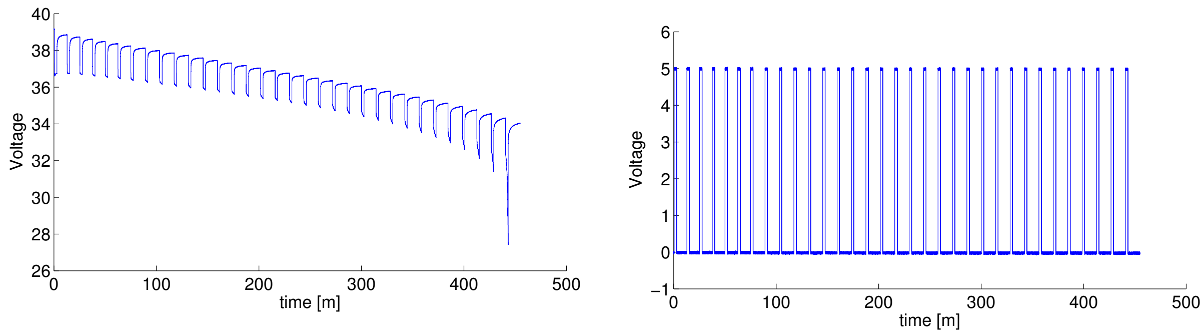
APPENDIX A

Lead-Acid Battery Model

This Appendix presents the experimental tests used to create the lead-acid battery model discussed in Section 5.2.1. This model was created by tuning the lead-acid battery model presented in [102] to our battery pack. There are two main steps to finding the battery model: finding the open circuit voltage (OCV) as a function of state of charge and fitting the resistance/capacitance values to match battery performance. A full discussion of these tests is presented in [105].

A.1 Pulse-Relaxation tests

To find the OCV, we use a pulse-relaxation test. The battery was initially charged and then a constant current was drawn from the battery for a set time. The SOC is computed by integrating the current draw. The battery is allowed to rest for a period of time while the voltage rebounds. The OCV is measured once the voltage has stabilized and the process is repeated. Figure A.1 shows the voltage and current profiles for one such test. From this test, the resulting OCV data is shown in Figure A.2.



(a) Voltage over time, showing rapid discharges followed by voltage relaxation

(b) Repetitive current profile

Figure A.1: Example pulse-relaxation test

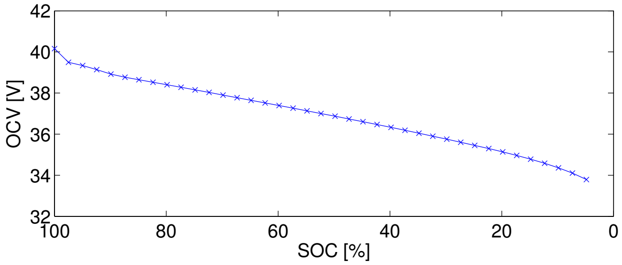


Figure A.2: OCV as a function of SOC

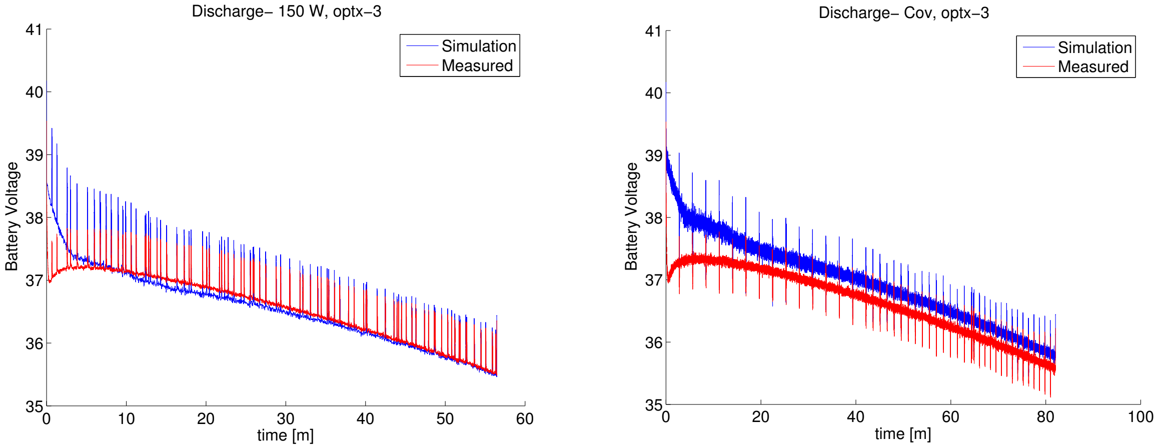


Figure A.3: Comparing battery model and measured voltage for two different power demands

A.2 Battery model

The OCV-R-RC battery model from [102] presumes an OCV source in series with a resistor and an RC circuit. With the OCV known, the other parameters of the model must be found. The primary resistance R is parameterized by soc s as $R = as^2 + bs + c$. The RC circuit parameters are assumed to be constant over all SOC values. The quality of our curve fit was determined by the squared difference between the simulated voltage and the measured voltage. Different parameters were found for charging and discharging.

A series of experiments were conducted to tune the battery parameters, including pulse-relax and discharge/charge tests. The parameters varied slightly between the different tests. Figures A.3 and A.4 compare the battery simulation and the experimental measurements for the discharge and charge profiles respectively. Model parameters were chosen by averaging the best fit parameters for the different test cases. In the discharge tests, there is an interesting rebound in the measured voltage at the beginning of the test, which is not accounted for in the simulation model. This rebound is possibly due to battery heating. It is unclear if this is a common phenomenon in lead-acid batteries; however, it is very common in our tests. The tail portion of the simulation matches

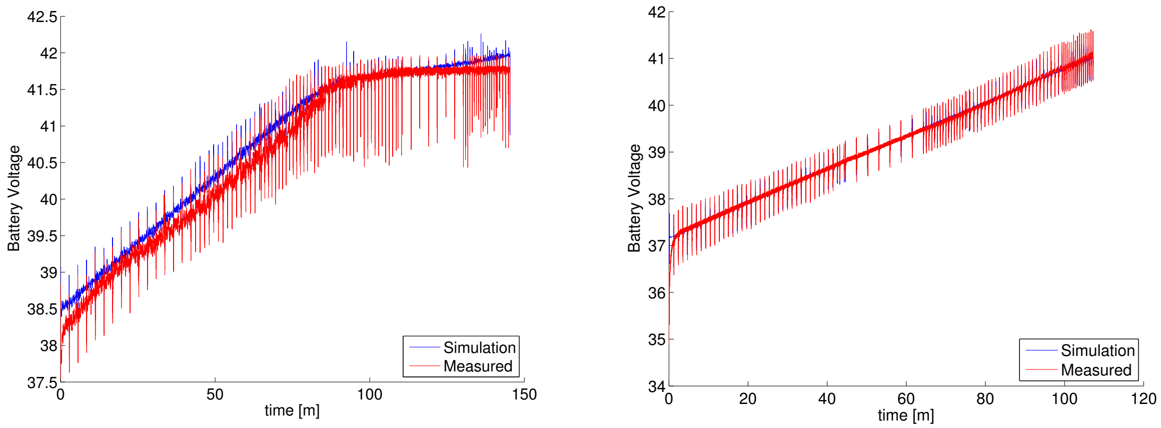
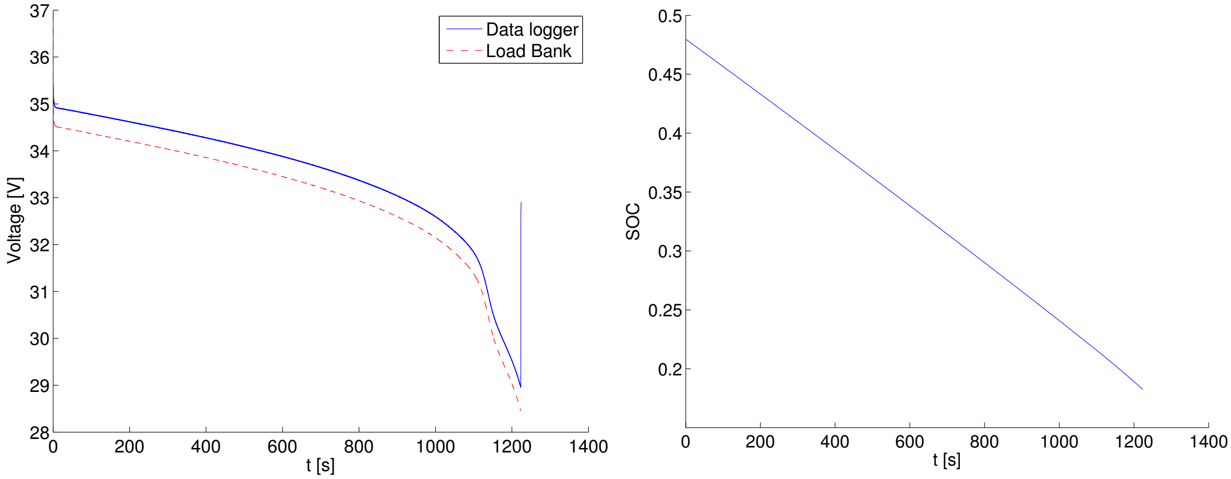


Figure A.4: Comparing battery model and measured voltage while charging



(a) Voltage

(b) State of charge

Figure A.5: Battery reaches cutoff voltage before depletion

the experimental results quite well. For the charging case, the model matches the acquired data closely.

A.3 Other Battery Considerations

For power system optimization, the other key components of the model are the low SOC cutoff and the charging characteristics. The SOC thresholds for the baseline control scheme depend on these battery parameters.

In one test shown in Figure A.5, the battery voltage reached the lower voltage cutoff of 29 V at $\tilde{1}8\%$ state of charge. The voltage at the load bank is slightly lower then the data logger due to resistance in the wires between the logger and the load bank. The power demand for this test was a constant 230 W. As such, we define the low limit on the state of charge as 18% for our model. For optimization, we also introduce a safety margin to prevent the battery voltage from getting too close to the lower SOC limit.

To determine charging characteristics of the combined fuel cell/battery system, the fuel cell was started with the battery at a lower state of charge and the battery was allowed to charge until full. As the battery charges, the system voltage increases until the voltage converters in the fuel cell reach an upper limit. At this point the charging current decreases as the battery state of charge continues to increase. From experimental data, we pull out the charging curve, shown in Figure A.6. This is used to determine the SOC dynamics for the *Battery Mostly Charged* states in the combined automata model (Figure 5.9).

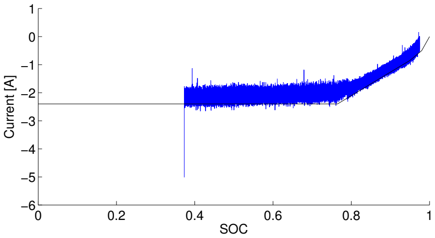


Figure A.6: Current charging profile for lead-acid battery

BIBLIOGRAPHY

[1] R. R. Murphy, *Disaster robotics*. MIT Press, 2014.

[2] S. Thrun, “Toward robotic cars,” *Commun. ACM*, vol. 53, no. 4, pp. 99–106, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721679

[3] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, “Towards fully autonomous driving: Systems and algorithms,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 163–168.

[4] J. Carlson and R. R. Murphy, “How UGVs physically fail in the field,” *IEEE Transactions on Robotics*, vol. 21, 2005.

[5] D. Tilbury and A. G. Ulsoy, “Reliable operations of unmanned ground vehicles: Research at the Ground Robotics Reliability Center,” in *Proceedings of IARP Workshop on Technical Challenges for Dependable Robots in Human Environments*, 2010, pp. 27–32.

[6] J. A. Broderick, D. M. Tilbury, and E. M. Atkins, “Characterizing energy usage of a commercially available ground robot: Method and results,” *Journal of Field Robotics*, vol. 31, no. 3, pp. 441–454, 2014. [Online]. Available: http://dx.doi.org/10.1002/rob.21507

[7] J. Broderick, D. Tilbury, and E. Atkins, “Energy usage for UGVs executing coverage tasks,” in *Proc. SPIE 8387*, April 2012.

[8] J. A. Broderick, D. M. Tilbury, and E. M. Atkins, “Optimal coverage trajectories for a UGV with tradeoffs for energy and time,” *Autonomous Robots*, vol. 36, no. 3, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10514-013-9348-x

[9] J. Broderick, D. Tilbury, and E. Atkins, “Maximizing coverage for mobile robots while conserving energy,” in *Proceedings of the ASME 2012 IDETC/CIE*, August 2012.

[10] J. A. Broderick, D. M. Tilbury, and E. M. Atkins, “Modeling and scheduling of multiple power sources for a ground robot,” in *Proceedings of the ASME DSCC 2014*, 2014.

[11] ——, “Supervisory traction control for a slipping UGV,” in *American Control Conference (ACC), 2013*, June 2013.

[12] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack, “Experiences with an architecture for intelligent, reactive agents,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 237–256, 1997.

[13] S. Bensalem, M. Gallien, F. Ingrand, I. Kahloul, and N. Thanh-Hung, “Designing autonomous robots,” *Robotics Automation Magazine, IEEE*, vol. 16, no. 1, pp. 67–77, 2009.

[14] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das, “The claraty architecture for robotic autonomy,” in *Aerospace Conference, 2001, IEEE Proceedings.*, vol. 1. IEEE, 2001, pp. 1–121.

[15] J. Wong and C. Chiang, “A general theory for skid steering of tracked vehicles on firm ground,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 215, no. 3, pp. 343–355, 2001.

[16] Y. Mei, Y.-H. Lu, Y. Hu, and C. Lee, “Energy-efficient motion planning for mobile robots,” in *Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on*, vol. 5, april-1 may 2004, pp. 4344 – 4349 Vol.5.

[17] ——, “A case study of mobile robot’s energy consumption and conservation techniques,” in *Advanced Robotics, 2005. ICAR ’05. Proceedings., 12th International Conference on*, july 2005, pp. 492 –497.

[18] K. Boice, A. Leo, J. Lee, J. Paulson Jr., M. Skalny, and T. Valascho, “Baseline field testing of bb-2590 lithium-ion batteries using an iRobot FasTac 510 robot,” TARDEC, Tech. Rep., September 2010.

[19] P. W. Richmond, G. L. Mason, B. A. Coutermarsh, J. Pusey, and V. D. Moore, “Mobility performance algorithms for small unmanned ground vehicles,” DTIC Document, Tech. Rep., 2009.

[20] J. Morales, J. Martinez, A. Mandow, A. Garcia-Cerezo, J. Gomez-Gabriel, and S. Pedraza, “Power analysis for a skid-steered tracked mobile robot,” in *Mechatronics, 2006 IEEE International Conference on*, july 2006, pp. 420 –425.

[21] J. Morales, J. Martinez, A. Mandow, A. Garcia-Cerezo, and S. Pedraza, “Power consumption modeling of skid-steer tracked mobile robots on rigid terrain,” *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1098 –1108, oct. 2009.

[22] S. Campanari, G. Manzolini, and F. G. de la Iglesia, “Energy analysis of electric vehicles using batteries or fuel cells through well-to-wheel driving cycle simulations,” *Journal of Power Sources*, vol. 186, no. 2, pp. 464 – 477, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775308018934

[23] H. A. Rakha, K. Ahn, K. Moran, B. Saerens, and E. V. den Bulck, “Virginia tech comprehensive power-based fuel consumption model: Model development and testing,” *Transportation Research Part D: Transport and Environment*, vol. 16, no. 7, pp. 492 – 503, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361920911000782

[24] H. Wang, L. Fu, Y. Zhou, and H. Li, “Modelling of the fuel consumption for passenger cars regarding driving characteristics,” *Transportation Research Part D:*

Transport and Environment, vol. 13, no. 7, pp. 479 – 482, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361920908001041

[25] I. El-Shawarby, K. Ahn, and H. Rakha, "Comparative field evaluation of vehicle cruise speed and acceleration level impacts on hot stabilized emissions," *Transportation Research Part D: Transport and Environment*, vol. 10, no. 1, pp. 13 – 30, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361920904000604

[26] J. Wong, *Theory of Ground Vehicles*. Wiley, 2008.

[27] B. Saha, K. Goebel, S. Poll, and J. Christophersen, "An integrated approach to battery health monitoring using bayesian regression and state estimation," in *Autotestcon, 2007 IEEE*, sept. 2007, pp. 646 –653.

[28] A. Sadrpour, J. Jin, and A. Ulsoy, "Mission energy prediction for unmanned ground vehicles," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 2229 –2234.

[29] M. Ceraolo and G. Pede, "Techniques for estimating the residual range of an electric vehicle," *Vehicular Technology, IEEE Transactions on*, vol. 50, no. 1, pp. 109 –115, jan 2001.

[30] Y. Zhang, W. Wang, Y. Kobayashi, and K. Shirai, "Remaining driving range estimation of electric vehicle," in *Electric Vehicle Conference (IEVC), 2012 IEEE International*, march 2012, pp. 1 –7.

[31] V. Berenz, F. Tanaka, and K. Suzuki, "Autonomous battery management for mobile robots based on risk and gain assessment," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 217–237, 2012.

[32] C. Ooi and C. Schindelhauer, "Minimal energy path planning for wireless robots," *Mobile Networks and Applications*, vol. 14, pp. 309–321, 2009, 10.1007/s11036-008-0150-5. [Online]. Available: http://dx.doi.org/10.1007/s11036-008-0150-5

[33] A. Barili, M. Ceresa, and C. Parisi, "Energy-saving motion control for an autonomous mobile robot," in *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on*, vol. 2, 10-14 1995, pp. 674 –676 vol.2.

[34] M. G. Bekker, *Introduction to terrain-vehicle systems*. University of Michigan Press Ann Arbor, MI, 1969.

[35] M. Grahn, "Prediction of sinkage and rolling resistance for off-the-road vehicles considering penetration velocity," *Journal of Terramechanics*, vol. 28, no. 4, pp. 339 – 347, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002248989190015X

[36] R. Pope, "The effect of wheel speed on rolling resistance," *Journal of Terramechanics*, vol. 8, no. 1, pp. 51 – 58, 1971. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022489871900759

[37] B. Crenshaw, *Soil/wheel interaction at high speed*. Society of Automotive Engineers, 1971.

[38] I. Shmulevich, U. Mussel, and D. Wolf, "The effect of velocity on rigid wheel performance," *Journal of Terramechanics*, vol. 35, no. 3, pp. 189 – 207, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022489898000226

[39] T. Guo and H. Peng, "A simplified skid-steering model for torque and power analysis of tracked small unmanned ground vehicles," in *American Control Conference (ACC)*, June 2013, pp. 1106–1111.

[40] H. Choset, "Coverage for robotics – a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[41] ——, "Coverage of known spaces: The Boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[42] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 77–98, 2001.

[43] ——, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry*, vol. 24, pp. 197–224, 2003.

[44] N. Hazon and G. A. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 735–741.

[45] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient Boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, pp. 109–142, 2008.

[46] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2001, pp. 27–32.

[47] I. Hussein and A. Bloch, "Dynamic coverage optimal control for multiple spacecraft," *Journal of Dynamical and Control Systems*, vol. 13, pp. 69–93, 2007.

[48] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, vol. 37, no. 3, pp. 279–300, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10514-014-9390-3

[49] C. Chan, "The state of the art of electric, hybrid, and fuel cell vehicles," *Proceedings of the IEEE*, vol. 95, no. 4, pp. 704–718, 2007.

[50] M. Ceraolo, A. Di Donato, and G. Franceschi, "A general approach to energy optimization of hybrid electric vehicles," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 3, pp. 1433–1441, 2008.

[51] S. Barsali, C. Miulli, and A. Possenti, "A control strategy to minimize fuel consumption of series hybrid electric vehicles," *Energy Conversion, IEEE Transactions on*, vol. 19, no. 1, pp. 187–195, 2004.

[52] C.-C. Lin, H. Peng, J. Grizzle, and J.-M. Kang, “Power management strategy for a parallel hybrid electric truck,” *Control Systems Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 839–849, 2003.

[53] P. Rodatz, G. Paganelli, A. Sciarretta, and L. Guzzella, “Optimal power management of an experimental fuel cell/supercapacitor-powered hybrid vehicle,” *Control Engineering Practice*, vol. 13, no. 1, pp. 41–53, 2005.

[54] Y. L. Murphey, Z. Chen, L. Kiliaris, and M. A. Masrur, “Intelligent power management in a vehicular system with multiple power sources,” *Journal of Power Sources*, vol. 196, no. 2, pp. 835 – 846, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775310012759

[55] A. N. Wilhelm, B. W. Surgenor, and J. G. Pharoah, “Design and evaluation of a micro-fuel-cell-based power system for a mobile robot,” *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 4, pp. 471–476, 2006.

[56] H.-I. Joh, T. J. Ha, S. Y. Hwang, J.-H. Kim, S.-H. Chae, J. H. Cho, J. Prabhuram, S.-K. Kim, T.-H. Lim, B.-K. Cho, J.-H. Oh, S. H. Moon, and H. Y. Ha, “A direct methanol fuel cell system to power a humanoid robot,” *Journal of Power Sources*, vol. 195, no. 1, pp. 293 – 298, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037877530901194X

[57] L. Li, F.-Y. Wang, and Q. Zhou, “Integrated longitudinal and lateral tire/road friction modeling and monitoring for vehicle motion control,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 1 –19, Mar 2006.

[58] U. Kiencke and L. Nielsen, *Automotive Control Systems: For Engine, Driveline, and Vehicle*. Springer, 2005.

[59] R. Balakrishna and A. Ghosal, “Modeling of slip for wheeled mobile robots,” *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 1, pp. 126 –132, Feb 1995.

[60] R. L. Williams, II, B. Carter, P. Gallina, and G. Rosati, “Dynamic model with slip for wheeled omnidirectional robots,” *Robotics and Automation, IEEE Transactions on*, vol. 18, pp. 285 –293, Jun 2002.

[61] N. Sarkar and X. Yun, “Traction control of wheeled vehicles using dynamic feedback approach,” in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, Oct 1998, pp. 413 –418 vol.1.

[62] J. G. Iossaqui, J. F. Camino, and D. E. Zampieri, “A nonlinear control design for tracked robots with longitudinal slip,” in *18th IFAC World Congress*, 2011, pp. 5932–5937.

[63] N. Sidek and N. Sarkar, “Dynamic modeling and control of nonholonomic mobile robot with lateral slip,” in *Third International Conference on Systems*, 2008, pp. 35–40.

[64] S. Nandy, S. N. Shome, R. Somani, T. Tanmay, G. Chakraborty, and C. S. Kumar, "Detailed slip dynamics for nonholonomic mobile robotic system," in *Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation*, 2011, pp. 519–524.

[65] X. Yun and N. Sarkar, "Unified formulation of robotic systems with holonomic and non-holonomic constraints," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 4, pp. 640 –650, Aug 1998.

[66] C. Unsal and P. Kachroo, "Sliding mode measurement feedback control for antilock braking systems," *Control Systems Technology, IEEE Transactions on*, vol. 7, no. 2, pp. 271 –281, Mar 1999.

[67] J. Song and K. Boo, "Performance evaluation of traction control systems using a vehicle dynamic model," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 218, no. 7, pp. 685–696, 2004.

[68] H.-S. Tan and Y.-K. Chin, "Vehicle traction control: Variable-structure control approach," *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, no. 2, pp. 223–230, 06 1991. [Online]. Available: http://dx.doi.org/10.1115/1.2896369

[69] T. Pilutti, G. Ulsoy, and D. Hrovat, "Vehicle steering intervention through differential braking," in *American Control Conference, 1995. Proceedings of the*, vol. 3, Jun 1995, pp. 1667 –1671 vol.3.

[70] G. Baffet, A. Charara, and J. Stephant, "Sideslip angle, lateral tire force and road friction estimation in simulations and experiments," in *2006 IEEE International Conference on Control Applications*, Oct 2006, pp. 903 –908.

[71] T. Dar and R. Longoria, "Slip estimation for small-scale robotic tracked vehicles," in *American Control Conference (ACC), 2010*, Jul 2010, pp. 6816 –6821.

[72] C. C. Ward and K. Iagnemma, "A dynamic-model-based wheel slip detector for mobile robots on outdoor terrain," *Robotics, IEEE Transactions on*, vol. 24, no. 4, pp. 821 –831, Aug 2008.

[73] K. Iagnemma and C. Ward, "Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain," *Autonomous Robots*, vol. 26, pp. 33–46, 2009.

[74] D. Kirk, *Optimal Control Theory*. Prentice Hall, 1970.

[75] A. Antoniou and W.-S. Lu, *Practical optimization: algorithms and engineering applications*. Springer, 2007.

[76] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *Control Systems*, vol. 29, no. 2, pp. 28–93, 2009.

[77] M. S. Branicky, "Introduction to hybrid systems," in *Handbook of Networked and Embedded Control Systems*, ser. Control Engineering. Birkhuser Boston, 2005, pp. 91–116. [Online]. Available: http://dx.doi.org/10.1007/0-8176-4404-0_5

[78] M. S. Branicky, V. S. Borkar, and S. K. Mitter, “A unified framework for hybrid control: Model and optimal control theory,” *Automatic Control, IEEE Trans.*, vol. 43, no. 1, pp. 31–45, 1998.

[79] P. I. Barton, C. K. Lee, and M. Yunt, “Optimization of hybrid systems,” *Computers & Chemical Engineering*, vol. 30, no. 1012, pp. 1576 – 1589, 2006, papers form Chemical Process Control {VII} {CPC} {VII} Seventh international conference in the Series. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135406001414

[80] X. Xu and P. J. Antsaklis, “Optimal control of switched systems based on parameterization of the switching instants,” *Automatic Control, IEEE Transactions on*, vol. 49, no. 1, pp. 2–16, 2004.

[81] Q. Lin, R. Loxton, and K. Teo, “Optimal control of nonlinear switched systems: Computational methods and applications,” *Journal of the Operations Research Society of China*, vol. 1, no. 3, pp. 275–311, 2013. [Online]. Available: http://dx.doi.org/10.1007/s40305-013-0021-z

[82] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, “The algorithmic analysis of hybrid systems,” *Theoretical Computer Science*, vol. 138, no. 1, pp. 3 – 34, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/030439759400202T

[83] R. Alur, S. La Torre, and G. J. Pappas, “Optimal paths in weighted timed automata,” in *Hybrid systems: computation and control*. Springer, 2001, pp. 49–62.

[84] ASTM Standard E2829, “Standard test method for evaluating emergency response robot capabilities: Mobility: Maneuvering tasks: Sustained speed,” 2011.

[85] ASTM Standard E2826, “Standard test method for evaluating emergency response robot capabilities: Mobility: Confined area terrains: Continuous pitch/roll ramps,” 2011.

[86] ASTM Standard E2801, “Standard test method for evaluating emergency response robot capabilities: Mobility: Confined area obstacles: Gap,” 2011.

[87] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers,” *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[88] SAE International, “Electric vehicle energy consumption and range test procedure,” 2012, standard J1634.

[89] A. Jacoff, R. Sheh, A.-M. Virts, T. Kimura, J. Pellenz, S. Schwertfeger, and J. Suthakorn, “Using competitions to advance the development of standard test methods for response robots,” in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, ser. PerMIS ’12. New York, NY, USA: ACM, 2012, pp. 182–189. [Online]. Available: http://doi.acm.org/10.1145/2393091.2393126

[90] A. N. Tikhonov and V. Y. Arsenin, *Solutions of ill-posed problems*. Washington, DC: Winston, 1977.

[91] Microrobotics, Inc., “MIDG IIC specifications,” 2011, accessed March 23, 2012. [Online]. Available: http://microboticsinc.com/midg_files/MIDG_IIC_Specifications_20111215.pdf

[92] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 1990, pp. 384–389.

[93] D. G. Logan, J. Pentzer, S. N. Brennan, and K. Reichard, “Comparing batteries to generators as power sources for use with mobile robotics,” *Journal of Power Sources*, vol. 212, no. 0, pp. 130 – 138, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775312006611

[94] M. G. Say and E. O. Taylor, *Direct Current Machines*, 2nd ed. London: Pitman, 1986.

[95] Y. Kim, J. Siegel, and A. Stefanopoulou, “A computationally efficient thermal model of cylindrical battery cells for the estimation of radially distributed temperatures,” in *American Control Conference (ACC), 2013*, June 2013, pp. 698–703.

[96] T. Ersal, Y. Kim, J. Broderick, T. Guo, A. Sadrpour, A. Stefanopoulou, J. Siegel, D. Tilbury, H. Peng, J. Jin, and A. G. Ulsoy, “Keeping ground robots on the move through battery & mission management,” *ASME Dynamic Systems and Control Magazine*, 2014.

[97] Y. Pang, M. Spathopoulos, and H. Xia, “Reachability and optimal control for linear hybrid automata: A quantifier elimination approach,” *International Journal of Control*, vol. 80, no. 5, pp. 731–748, 2007.

[98] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer, 2008.

[99] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, no. 3, pp. 407 – 427, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109898001782

[100] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-Parametric Toolbox 3.0,” in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510.

[101] J. Broderick, J. Hartner, D. Tilbury, and E. Atkins, “Modeling and simulation of an unmanned ground vehicle power system,” in *Proc. SPIE*, vol. 9084, 2014. [Online]. Available: http://dx.doi.org/10.1117/12.2050483

[102] Y. Parvini, E. G. Urdaneta, J. B. Siegel, S. Choi, A. Vahidi, and L. Thompson, “Range extension study of a hybridized lead-acid battery using ultracapacitors via simulation and experimental results from a 12 volt actively controlled hardware in the loop test bench,” 2014, in preparation.

[103] Y. Kim, S. Mohan, J. B. Siegel, and A. G. Stefanopoulou, “Maximum power estimation of lithium-ion batteries accounting for thermal and electrical constraints,” in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013, pp. V002T23A003–V002T23A003.

[104] H. K. Khalil, *Nonlinear systems*. Prentice Hall, Upper Saddle River, 2002, vol. 3.

[105] J. A. Broderick, D. M. Tilbury, and E. M. Atkins, “Control of a lead-acid battery/fuel cell hybrid power system for a UGV: Experimental report,” Automotive Research Center, University of Michigan, Tech. Rep., 2014. [Online]. Available: http://deepblue.lib.umich.edu/handle/2027.42/109735