2021

UNMANNED GROUND VEHICLE (UGV) DOCKING, CONNECTION, AND CABLING FOR ELECTRICAL POWER TRANSMISSION IN AUTONOMOUS MOBILE MICROGRIDS

John Naglak
Michigan Technological University, jenaglak@mtu.edu

UNMANNED GROUND VEHICLE (UGV) DOCKING, CONNECTION, AND CABLING FOR ELECTRICAL POWER TRANSMISSION IN AUTONOMOUS MOBILE MICROGRIDS

By

John E. Naglak

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Mechanical Engineering - Engineering Mechanics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2021

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Mechanical Engineering - Engineering Mechanics.

Department of Mechanical Engineering - Engineering Mechanics

Dissertation Co-advisor: *Dr. Wayne W. Weaver*

Dissertation Co-advisor: *Dr. Jeremy P. Bos*

Committee Member: *Dr. Jung Yun Bae*

Committee Member: *Dr. Laura E. Brown*

Committee Member: *Dr. William J. Endres*

Department Chair: *Dr. William W. Predebon*

Dedication

Silas and Daniel

The most wonderful outcome of this PhD!

Contents

List of Figures . . . . . **xiii**

List of Tables . . . . . **xxi**

Author Contribution Statement . . . . . **xxiii**

Acknowledgments . . . . . **xxv**

List of Abbreviations . . . . . **xxvii**

Abstract . . . . . **xxix**

1 Introduction to Autonomous Mobile Microgrids **1**

1.1 Motivation and Applications 1

1.2 Mobile Microgrid State-of-the-Art 11

1.3 Contribution: A method for UGV docking, connection, and cabling for electrical power transmission 15

2 Autonomous Power Grid Formation for Surface Assets Using Multiple UGVs **17**

2.1 Introduction 17
2.2 Robotic Platform 20
2.2.1 Microgrid Control States 22
2.2.2 Waypoint Controller 23
2.2.3 Inter-Agent Communication 24
2.2.4 Heterogeneous Power Sources 25
2.3 Docking Controller 25
2.4 Power Coupling 27
2.4.1 Electrical Connection Coupling 29
2.4.2 Near-field wireless power transmission 30
2.5 Cable Deployment 31
2.6 Validation Test Method 33
2.6.1 Subsystem Test Method 34
2.6.2 ROS move_base configuration 36
2.6.3 Proposed Full System Test Method 37
2.7 Results 40
2.8 Conclusion 42

3 Cable Deployment System for Unmanned Ground Vehicle(UGV) Mobile Microgrids 43

3.1 Abstract 43
3.2 Hardware in context 45

3.3 Hardware Description 50
3.3.1 Electronics 54
3.3.2 Control Code 56
3.4 Design files 57
3.5 Bill of materials 58
3.6 Build instructions 58
3.7 Operation instructions 63
3.8 Validation and characterization 66
3.9 Conclusion 67

4 AC/DC Power Connector for Unmanned Ground Vehicle(UGV) Mobile Microgrids 69

4.1 Hardware in context 71
4.2 Hardware Description 72
4.3 Design files 75
4.4 Bill of materials 77
4.5 Build instructions 77
4.6 Operating Instructions 82
4.7 Validation and Characterization 83
4.8 Conclusion 85

5 Visual Cable Tracking and Deployment for Autonomous Mobile Microgrid Power Transmission 86

5.1 Motivation 87
5.1.1 Introduction to Mobile Microgrids 88
5.1.2 Power Connection Coupling and Cabling 90
5.1.3 Statement of Contribution 92
5.1.4 Contents 92
5.2 Controllers for Docking, Connection, and Cabling 93
5.2.1 ROS move_base Implementation 94
5.2.2 Docking via Visual Feedback 95
5.2.3 Electrical Connection Coupling 98
5.2.3.1 Floating Connection 98
5.2.3.2 Fixed Connection 100
5.2.4 Cable Physical Model 101
5.2.5 Visual Cable Tracking 103
5.2.6 Cable Deployment Controller 108
5.3 Hardware Configuration and Validation Method 111
5.4 Results 115
5.5 Conclusion 125

6 Conclusion and Opportunity for Future Development 129

6.1 Conclustion 129
6.1.1 Chapter 2 130
6.1.2 Chapter 3 130

6.1.3 Chapter 4 131

6.1.4 Chapter 5 132

6.2 Extensions and Future Work 132

References **135**

A Visual Docking for Mobile Microgrid Power Transfer: Simulation and Validation **151**

A.1 Introduction 151

A.2 Microgrid UGV Configuration 154

A.3 Visual Controller for Docking 155

A.3.1 ROS move_base Implementation 156

A.3.2 Iterative Visual Feedback 156

A.4 Monte-Carlo Validation Method 157

A.4.1 Gazebo Simulation Environment 160

A.4.2 Outdoor Test Environment 160

A.4.3 3 Load Recharging 162

A.5 Results 163

A.5.1 Simulation 163

A.5.2 Outdoor Validation 166

A.5.3 3 Load Recharging 170

A.6 Conclusion 171

B Sample Code **173**

B.1 Cable Visual Tracking 173

B.2 Cable Control 180

B.3 Low-Level Micro-Controller 187

C Letters of Permission **197**

List of Figures

1.1 Mobile microgrids include power source, bus, and power transmission hardware that is integrated with mobile robots for operation in human-denied environments 2

1.2 Deployment of an Autonomous Mobile Microgrid depends on positioning of power sources and connection with infrastructure loads. . . . 3

1.3 Autonomous Mobile Microgrid UGV agents depend on waypoint navigation, visual docking, cable deployment, and power hardware controllers. 6

2.1 A Kilopower array would be autonomously deployed and interconnected by a number of UGV agents [1] 19

2.2 Clearpath Husky UGV provides all-terrain navigation with sufficient payload capacity for demonstration power grid hardware. 21

2.3 Control states for the UGV can include waypoint navigation, docking, connection coupling, and cable management. 22

2.4 Docking control for wireless power transfer depends on both Canonical elements of the ROS move_base software, and custom algorithms. . 27

2.5 Connection coupling depends on visual feedback of the receiving connector pose using a single camera. 28

2.6 Primare coil hardware is mounted on the front of the UGV, and secondary coil hardware can be mounted to any load. When the coils are brought within sufficient range, power transfer is initiated. 28

2.7 Deployment of the electrical cable leverages the UGV linear and angular velocity command, with known geometry characteristics, to trace the trajectory of the UGV as it traverses the operating environment. 31

2.8 Proof of concept electrical cable deployment system spools cable at a rate which tracks the path traversed by the UGV. 31

2.9 Validation tests were performed in an open environment outdoors, off road. 33

2.10 A Monte-Carlo type full system test is envisioned to demonstrate capability of the unified control architecture, with multiple UGVs participating 35

2.11 EKF filtered trajectories of the UGV transiting between random waypoints. 37

2.12 EKF filtered trajectories of UGV approaching and docking during early testing for near-field wireless power transfer. Successful approaches are blue. 38

2.13 EKF filtered trajectories of UGV duing Monte-Carlo testing for docking and near-field wireless power transfer. 39

3.1 An Autonomous Mobile Microgrid UGV deploys a power cable to connect a genset power source to an infrastructure load. 47

3.2 Complete assembly of the ACMM 51

3.3 Full assembly explosion and subsections. 52

3.4 COTS cable spool 52

3.5 Wiring Diagram 55

3.6 Function Block diagram of ACMM control code. 56

3.7 Frame and Motor assembly 59

3.8 Pully and belt drive system 61

3.9 Bearing Block Assembly 62

3.10 Tensioner Assembly 63

3.11 Electronics Enclosure 64

3.12 A demonstration video of the ACMM on a UGV includes, from left to right, an infrastructure load, the cable following the track of the UGV through the environment, and a genset. 67

4.1 Heterogeneous power source UGVs prepare to couple and form an AC/DC power bus. 71

4.2 The AC/DC connector system enables power transmission between a UGV and stationary infrastructure. 73

4.3 A flexible rubber element integrated in the connector complies to mitigate misalignment in the lateral and axial degrees of freedom during docking. 75

4.4 Locations of printed and COTS components, with BOM numbering. 78

4.5 Optional female mounting configurations include a bumper mount for the Clearpath Husky UGV. 80

4.6 Depending on mobility and application, the male connector can be mounted to an actuator, in this case a single degree of freedom in the axial direction. 81

4.7 UGV positions for connection insertions across an array of lateral and axial miss-alignment. 84

5.1 Mobile microgrids include power source, bus, and power transmission hardware that is integrated with mobile robots for operation in human-denied environments 89

5.2 Docking control leverages ROS core navigation functionality with a visual method which introduces increasingly accurate estimates of the position of the docking goal 96

5.3 ACDC electrical connector for UGV coupling. 99

5.4 Insertion of the floating connector is performed by a linear actuator, with visual feedback for extension set-point provided by the UGV, and PI feedback to achieve that set-point performed by a μcontroller. . . 100

5.5 Cable deployment is based on a simplified model of a catenary-shaped suspended cable in equilibrium with the cable-ground friction interaction. 101

5.6 Cable droop for a range of cable-ground static friction coefficients. . 104

5.7 Cable angle is fit to a mask, updated using HSV mean and standard deviation from ROI pixels. 105

5.8 New camera frames are reduced to a cable vertical angle estimation by estimating the HSV parameter range of the cable from a ROI and performing a Hough Line transformation on the masked pixels. . . . 106

5.9 Feed-forward/Feedback control of cable deployment depends on the vehicle velocity command, with visual estimation of the cable deployment angle closing the loop via PI gains. 109

5.10 Ruggedized UGV microgrid agents are equipped with a sensor suite appropriate for their operating environment, as well as power hardware. In this example, the UGV is equipped with a cable deployment system, electrical connectors, and monocular docking cameras, in addition to its standard navigational sensors. 111

5.11 The outdoor test environment was generally open and covered in grass. 113

5.12 The location of an obstacle in the cable path is permuted though a region laying in the UGV path. 114

5.13 Cable segmentation samples include variation in lightning and background content. 116

5.14 Feed-forward/Feedback control for 0 velocity steady-state, followed by a step input of 0.195 m/s in the forward direction. 117

5.15 Feed-forward/Feedback control for 0 velocity steady-state, followed by a step input of 0.195 m/s in the reverse direction. 119

5.16 Feed-forward/Feedback control for $-\pi/2rad$ turns at $-0.389rad/s$. 120

5.17 Feed-forward/Feedback control response with cable angle estimate for an outdoor cabling traversal and visual docking. 121

5.18 UGV trajectories while deploying cable between a genset and a load. 123

5.19 UGV trajectories while deploying cable with an obstacle in the UGV path. 125

5.20 UGV deploying cable with an obstacle in the UGV path. 126

A.1 Ruggedized UGV microgrid agents are equipped with a sensor suite appropriate for their operating environment, as well as power hardware. In this example, the UGV carries a generator and transfers power via wireless charging coil. The load coil location is indicated by an AR marker. 154

A.2 Pose goals were distributed about the face of the docking goal in a uniform random distribution throughout a half-washer shaped region 159

A.3 An outdoor environment with a fixed load was modeled in Gazebo, with a flat grass surface. 161
A.4 The outdoor test environment was a generally open and covered in grass. 162
A.5 Trials for $\mu = 0.36$ 165
A.6 Trials for $\mu = 0.5$ 165
A.7 Trials for $\mu = 0.77$ 166
A.8 Trials for $\mu = 1.54$ 166
A.9 Monte-Carlo evaluation of traversal time cost between waypoint poses in the Gazebo simulation environment. 167
A.10 Outdoor Monte-Carlo validation results consisted of 18 docking trials over a region that was expected to exceed the capability of the controller. 168
A.11 EKF filtered trajectories of the UGV transiting between random waypoints in an outdoor, grassy, environment. 170
A.12 A UGV with wireless power transmission approaches one of 3 loads to recharge it in simulation. 171
A.13 3 loads in a laboratory environment, with charging UGV making a wireless power transmission docking approach. 172

List of Tables

2.1 move_base parameters associated with the field trials reported in this work. 34

5.1 move_base parameters associated with the cable deployment vehicle reported in this work. 94

A.1 Success rate and arc for multiple values of μ on a simulated grassy surface. 163

Author Contribution Statement

This dissertation is partially comprised of published material which is republished in its entirety as chapters therein. Author contributions are as follows:

Autonomous Power Grid Formation for Surface Assets Using Multiple Unmanned Ground Vehicles **John Naglak:** Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration **Carl Greene** Methodology, Investigation, Writing - Review & Editing **Casey Majhor** Methodology, Investigation, Writing - Review & Editing **Nathan Spike** Methodology, Software **Jeremy Bos** Conceptualization, Writing - Review & Editing, Supervision **Wayne Weaver** Conceptualization, Writing - Review & Editing, Funding acquisition

Cable Deployment System for Unmanned Ground Vehicle(UGV) Mobile Microgrids **John Naglak:** Conceptualization, Methodology, Software, Validation, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration **Caleb Kase** Methodology, Writing - Original Draft, Visualization **Max McGinty** Methodology, Software, Writing - Review & Editing **Casey D. Majhor** Writing - Review & Editing **Carl S. Greene** Methodology **Jeremy P. Bos** Supervision **Wayne W. Weaver** Conceptualization, Funding acquisition

AC/DC Power Connector for Unmanned Ground Vehicle(UGV) Mobile Microgrids **John Naglak** Conceptualization, Methodology, Software, Validation, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration **Trevor Barrett** Methodology, Validation **Nathan Bondi** Methodology **Sam Krusinski** Methodology, Visualization **Max McGinty** Methodology, Software, Writing - Review & Editing, Visualization **Travis Moon** Methedology **Jeremy Bos** Supervision **Wayne Weaver** Conceptualization, Funding acquisition

Visual Cable Tracking and Deployment for Autonomous Mobile Microgrid Power Transmission **John Naglak** Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration **Jeremy Bos** Conceptualization, Supervision, Writing - Review & Editing **Wayne Weaver** Conceptualization, Writing - Review & Editing, Funding acquisition

Acknowledgments

This work is sponsored in part under U.S. Office of Naval Research under award N00014-16-1-2422.

We would like to thank Michigan Tech Recreation for the generous use of the Tech Trails and facility for field trials during our summer testing seasons.

Although a comprehensive recognition of everyone who has encouraged and supported my work is impossible, I would like to extend it to the following individuals:

My co-advisors Dr. Wayne Weaver and Dr. Jeremy Bos for providing a complementary advising team as we have achieved milestones together.

To my committee for flexibility throughout the events of the last year.

Lab-mates Carl Greene and Casey Majhor for partnering to provide the applications for my work.

Illustrious undergrads Caleb Kase and Max McGinty.

Nate Spike for Brotherhood.

The following have provided the context and encouragement which ultimately led me

pursue a PhD: Dr. Ryan N. Smith, Dr. William Nollet, David McAlvany, Tom Leps, Dr. Nicole Nichols, Dr. Andrew Keck, Mark Gross, Steve Mulich

My parents, Dave and Jennifer, for patience.

Silas and Daniel for Smiles.

My wife Laura, for everything else humanly possible!

And the Lord Jesus, for everything beyond my ability.

List of Abbreviations

AC/DC	Alternating Current/Direct Current
ACMM	Adjustable Cable Management Mechanism
API	Application Programming Interface
AR	Augmented Reality
AUV	Autonomous Underwater Vehicle
COTS	Commercial-Off-The-Shelf
DoD	Department of Defense
EKF	Extended Kalman Filter
FEMA	Federal Emergency Management Agency
FOB	Forward Operating Base
FOV	Field of View
GPS	Global Positioning System
GPU	Graphics Processing Unit
HSV	Hue Saturation and Value
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
OSS	Open Source Software
PCI	Peripheral Component Interconnect

PCL	Point Cloud Library
PI	Proportional Integral
PID	Proportional Integral Derivative
PV	Photo Voltaic
ROI	Region Of Interest
ROS	Robot Operating System
SDK	Software Development Kit
TIB	Timed Elastic Band
UAV	Un-manned Arial Vehicle
UGV	Unmanned Ground Vehicle
USB	Universal Serial Bus
USV	Un-manned Surface Vehicle (boat)
UTM	Universal Transverse Mercator

Abstract

Autonomous Mobile Microgrids provide electrical power to loads in environments where humans either can not, or would prefer not to, perform the task of positioning and connecting the power grid equipment. The contributions of this work compose an architecture for electrical power transmission by Unmanned Ground Vehicles (UGV). Purpose-specific UGV docking and cable deployment software algorithms, and hardware for electrical connection and cable management, has been deployed on Clearpath Husky robots. Software development leverages Robot Operating System (ROS) tools for navigation and rendezvous of the autonomous UGV robots, with task-specific visual feedback controllers for docking validated in Monte-Carlo outdoor trials with a 73% docking rate, and application to wireless power transmission demonstrated in an outdoor environment. An "Adjustable Cable Management Mechanism" (ACMM) was designed to meet low cost, compact-platform constraints for powered deployment and retraction by a UGV of electrical cable subject to disturbance, with feed rates up to 1 m/s. A probe-and-funnel AC/DC electrical connector system was developed for deployment on UGVs, which does not substantially increase the cost or complexity of the UGV, while providing a repeatable and secure method of coupling electrical contacts subject to a docking miss-alignment of up to +/-2 cm laterally and +/-15 degrees axially. Cabled power transmission is accomplished by a feed-forward/feedback control method, which utilizes visual estimation of the cable state

to deploy electrical cable without tension, in the obstacle-free track of the UGV as it transverses to connect power grid nodes. Cabling control response to step-input UGV chassis velocities in the forward, reverse, and zero-point-turn maneuvers are presented, as well as outdoor cable deployment. This power transmission capability is relevant to diverse domains including military Forward-Operating-Bases, disaster response, robotic persistent operation, underwater mining, or planetary exploration.

Chapter 1

Introduction to Autonomous Mobile Microgrids

1.1 Motivation and Applications

Autonomous Mobile Microgrids can provide electrical power to loads in environments where humans either can not, or should not, perform the task of positioning and connecting the power grid equipment. Mobile microgrids combine the functionality of independent power grids with multi-domain, multi-purpose, autonomous Unmanned Ground Vehicles(UGV). Applications with human risk or human denied environments are especially valuable, which range from military forward operating bases or disaster

Figure 1.1: Mobile microgrids include power source, bus, and power transmission hardware that is integrated with mobile robots for operation in human-denied environments

response, to infrastructure installation for future habitation of the Moon or Mars. Micro power grids incorporate energy generation with grid storage, power transmission, and grid optimization or power control. Energy generation can consist of mobile gas or diesel gen-sets, as well as photovoltiac arrays or other renewable sources. Grid-attached storage can be provided by mobile battery banks. Power transmission has traditionally depended on power cables, but can be implemented with wireless methods. Grid optimization and power control depends on power switching and conversion hardware. AC/DC conversion serves different types of infrastructure Load balancing and load shedding adapts the grid to changes in power demand or supply over time. Sensing and computational resources are required for optimization and feedback

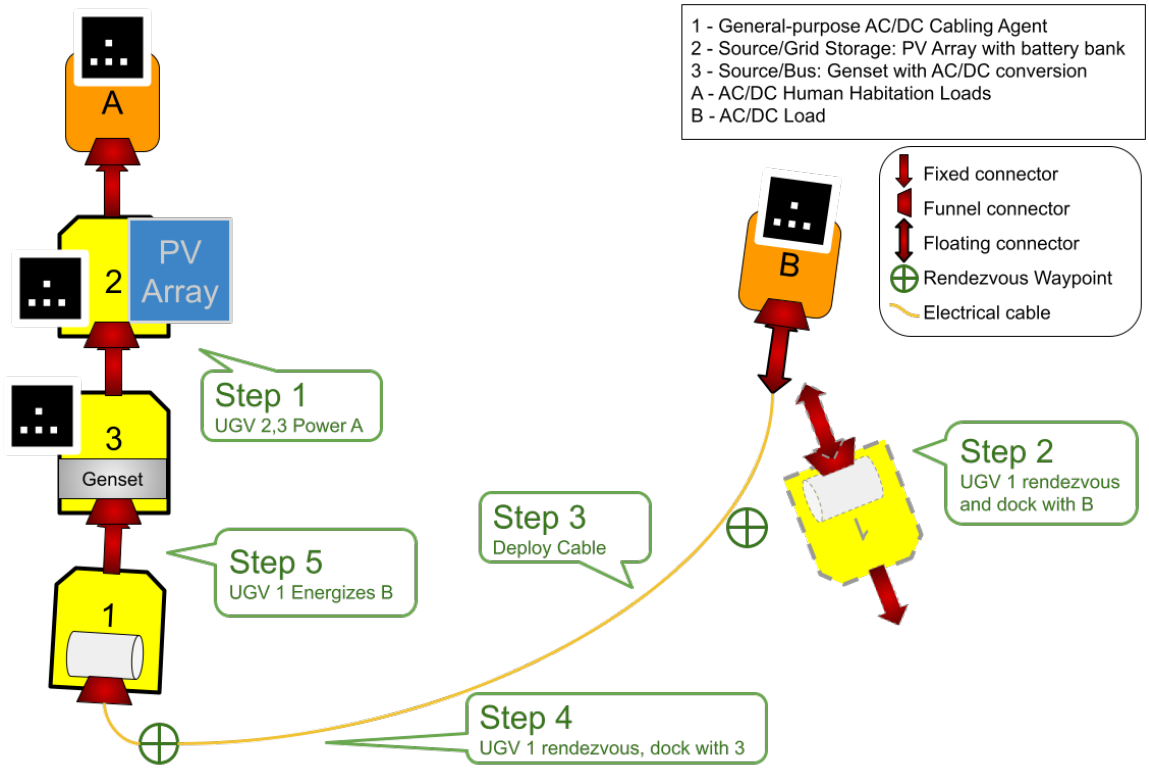


Figure 1.2: Deployment of an Autonomous Mobile Microgrid depends on positioning of power sources and connection with infrastructure loads.

power grid controllers. Additionally, mobile microgrids must optimize for both autonomous vehicle constraints and the power grid requirements. Power grid hardware is deployed on the mobile robots, which configure according to mission requirements and any fixed load or power source infrastructure, Figure 1.1. If conditions change, the mobile grid elements should be able to optimally reconfigure.

In this work, electrical cables are deployed from Unmanned Ground Vehicles to connect power sources to infrastructure loads. This task can be broken down into three processes: docking, connection, and cabling. Does existing work provide the robotic control necessary to locate and connect power resources? What hardware needs to

be developed to accomplish the tasks? Prior art specifically motivated by mobile microgrids, Section 1.2, shows minimal treatment of robotic control for mobile energy assets, and no autonomous deployment of these assets in outdoor environments. Cable state estimation in the prior art is almost exclusively handled in the simplified domains of air or water, or neglected with a taut-cable assumption. Consider a microgrid configuration scenario depicted in Figure 1.2. In it, two loads are fixed in the operating environment, A and B. These loads have different power requirements, both in power specification and in duty cycle. This arrangement is typical in the real world, where infrastructure may have both AC and DC power consumption, and some infrastructure exhibits a constant power demand, while other processes have high wattage, short duty-cycle demand. Consider, for example, a military Forward Operating Base (FOB), which includes human habitation infrastructure and bulk fluid transfer pumps. These loads are representative of many deployment scenarios, such as a disaster response environment, or on a moon or mars base. The habitation has AC loads such as HVAC system, and DC loads for lighting or communications. Pumping loads happen infrequently, but with relatively high power consumption. A migrogrid must be configured to support these demands, with power resources distributed to optimize efficient power delivery. In the scenario, for simplicity consider the mobile genset and PhotoVoltaic (PV) Array with grid storage to have already achieved positioning and connection at Load A, Step 1. If Load A has mixed AC and

DC power consumption characteristics, with constant power demand, the heterogeneous power sources deployed on mobile robots 2 and 3 can be optimized to meet these power requirements. UGV 1 approaches and connects one end of a power cable to Load B, Step 2, and activates a cable deployment controller, Step 3. It proceeds to autonomously traverse thorough the deployment region to rendezvous with, and connect to, the heterogeneous power sources, Step 4. Now, the mobile microgrid has been expanded to encompass this additional load, Step 5.

Traversal through the operating region, localization of the UGV relative to a connection target, docking of the UGV, and power connection and transmission, are all steps necessary to achieve the above scenario. These steps depend on prior development, and also form the contribution of this work. These contributions, and limitations thereof, should be briefly elucidated by considering their role in the scenario, with detailed description and background available in the chapter content. Figure 1.3 depicts the main control states the UGV would activate in the example scenario. Waypoint Navigation is a core capability of the UGV robot. In this application, GPS data is fused with IMU measurements to provide localization in reference to a world coordinate frame shared by all the microgrid assets. This localization estimate serves as a reference for measurement of obstacles identified by the vehicle LiDAR sensor, and for the optimal (global) and motion (local) chassis path planners. Each of these layers is available to this development thanks to the Open Source Software software (OSS) community organized around ROS, specifically the Navigation Stack. For purposes

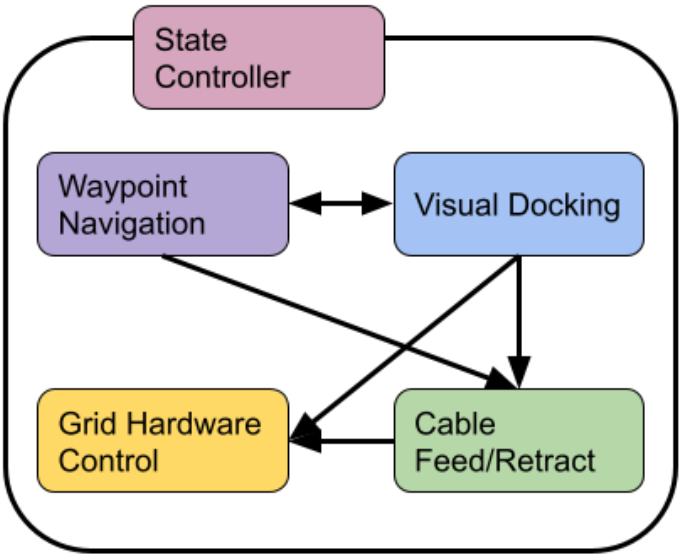


Figure 1.3: Autonomous Mobile Microgrid UGV agents depend on waypoint navigation, visual docking, cable deployment, and power hardware controllers.

of general transit of the mobile microgrid UGV resources, and rendezvous waypoint pose alignment, this navigation stack was configured and deployed via API's in the vehicle state controller. An important limitation of this controller is dependence on high fidelity GPS signal in the microgrid operating region. In some cases, noise in the GPS signal limits the certainty of the UGV position estimate relative to fixed infrastructure to the order of meters, requiring additional efforts to achieve precision docking for power grid connection.

Visual docking resolves the uncertainty in position estimate of the connection target in the UGV reference frame. Augmented Reality (AR) tags, a type of 2D fiducial marker, label the microgrid connection hardware. Monocular camera's are used to track the position of these AR tags, which can be used to update a docking alignment

waypoint for the Navigation stack, or to visual servo the UGV chassis during a docking maneuver. Use of fiducial markers is common in machine vision applications, and the utilization for mobile robotics provides value in dependable performance with established OSS library implementation. Other noteworthy implementations of fiducial markers include localization of sensors deployed by the Mars Curiosity rover [2], and feedback for docking of the two halves of the Axel rover [3]. Limitations in range and accuracy are generally correlated to the cost invested in the optics used for marker acquisition. Development of a robust visual docking controller is foundational to power connection or power transmission tasks in the autonomous mobile microgrid. Mating of conductive contacts is one outcome of docking. In this case, the risidual alignment error of the UGV chassis must be within the tolerance of the compliant mechanisms of the power connector. Alternately, there is increasing availability of inductive wireless power transmission hardware. Wattage capacity of these inductive coils is limited by physical size, but they have certain advantage for some mobile robotics applications. When housed in water resistant enclosures they are inherently resilient to outdoor environmental conditions, and can transmit power despite some miss-alignment between the transmitting and receiving fields, albeit at decreased efficiency. Of course, this residual miss-alignment can be measured using the visual marker tracking, and the UGV can re-attempt the docking approach if needed to ensure adequate power transmission efficiency. Docking time cost and docking alignment quality are also important characteristics for planning and optimization of mobile microgrid UGV

organization and power distribution control.

Cabled connection between components of the microgrid allows for integration of physically separate fixed infrastructure, or distribution of the UGV-deployed resources according to their requirements (E.g. solar gain, noise mitigation). One approach to planning for cable routing in a deployment environment is to depend on the UGV optimal path planner with obstacle avoidance, and lay cable in a manner that ensures it remains in the "track" that the UGV traverses. This places the contrary constraints on the cable feed controller to simultaneously ensure that the cable descends from a fairlead on the UGV to the ground as near to the rear of the chassis as possible, but does not touch or become ensnared in the chassis or wheels during turns or reverse travel. First, a cable feed system suitable for deployment on the UGV must be able to store, deploy, and retract cable as the vehicle maneuvers. Second, a control scheme for cable deployment must take into consideration the constraints of both the UGV and the cabling system. Motion planning for the UGV along this optimal track, provided by the ROS Navigation Stack, has configurable limits for linear and angular velocities and accelerations, and as configured seeks to optimize "elastic band" paths through the navigable space. Rather than developing a cabling-specific motion planner, the existing planner was utilized and constrained to acceptable limits for a cable feed rate controller. Because of a dearth of prior work on cable deployment by autonomous UGVs, and none which correspond to these constraints, a new visual feedforward/feedback controller transforms the UGV 2D motions into scalar cable

feed rates. This has produced paths that are suitable for the physical behavior of the electrical cable, with a pragmatic balance between UGV progression speed and cable deployment transients. Future work to develop a motion planner which optimized cable deployment (minimize jerk) as well as smooth UGV navigation, could produce measurable improvements in cable deployment control transients, as well is empirical improvement in cable deployment.

Power grid hardware state must correspond to the states of the rendezvous, docking, and connection process. The UGV state controller signals wired or wireless systems to control safety interlocks, and poll secondary systems for power grid performance. At this time, UGV microgrid agents act independently using a pre-planned mission schedule with shared time-base. Future work could perform real-time updates to the power distribution and optimization scheme, including scheduling updates for UGV progression.

This dissertation includes content that has been submitted to refereed conference and journal publications. The following list outlines these publications and the subset of the docking, connection, and cabling tasks addressed therein:

† Chapter 2: Docking depends on both a-priori knowledge of the rendezvous location between a UGV and its goal, and real-time measurement of the docking misalignment during final approach. Common Robot Operating System (ROS)

localization algorithms providing a navigation stack are extended by a visual Augmented Reality (AR) tracking feedback controller for docking.

† Appendix A: Development of simulation support for the mobile microgrid controllers has enabled further investigation into applications for the docking controller, including charge scheduling with real-world equivalent demonstrations.

† Chapter 3: UGV-specific design constraints should be considered for cable deployment hardware. A cable deployment drive system with integrated storage reel has been designed and deployed on an autonomous mobile microgrid robot.

† Chapter 4: Connection requires a purpose-designed power connector that meets the electrical specifications of the project, and mitigates the residual alignment error following the docking approach. AC/DC power connectors have been designed and installed for the UGV chassis and fixed infrastructure.

† Chapter 5: Cable should lay without tension in the deployment track of the UGV. Open-loop control of the cable feed rate as the UGV traverses is not robust to error in the vehicle's odometry estimate, nor can it compensate for disturbance in the cable-ground interface. Direct measurement of the cable tension is not practical. Instead, this work presents a visual method which measures the pitch angle of the cable as it exits the UGV. The angle set-point can be derived from a catenary cable physical model. This visual pitch measurement provides the feedback signal for feed rate as the UGV traverses a cable deployment or

cable recovery path. Combining information from the UGV traversal plan with visual angle estimation yields a practical feed-forward/feedback controller for deployment and retrieval of the cable.

1.2 Mobile Microgrid State-of-the-Art

Mobile microgrids are one solution to recovering from an event which disables a regional powergrid [4], with much work on utilization of existing distributed resources, such as home PV arrays [5] and consumer electric vehicles [6] as supply and energy storage nodes. Additional work provides modular generation and storage resources [7] , as well as optimization for their allocation [8], with truck-mounted mobile energy resources to restore islanded loads [9], and pre-positioning and re-allocation of mobile energy storage units [10]. Although some work explores the power system topology [11], there is little handling of real-world constraints or mobility costs for mobile power assets. One work explores routing cost in a damaged urban environment [8], but does not consider challenging implications of unstructured navigation or autonomy on the routing cost function. Unfortunately, damage to un-hardened distribution and control infrastructure, such as above-ground power lines and substations, is common [12]. To our knowledge, no work has provided an autonomous robotic capability for replacement of critical power grid infrastructure following a disaster. Robotic agents for power grid recovery should be prepared to fulfill the essential functions

of a microgrid: power generation, energy storage, transmission, and power control [13]. Mobile microgrids should be readily deployable, decentrally staged awaiting deployment, and capable of operating in the degraded infrastructure encountered following a disaster.

Very little work is available in the literature regarding the deployment of power grids on mobile robots. Some relevant work focuses on allocation and control of power sources [8, 14], as well as design of mobile power sources [7]. A modular mission payload system was designed, including an integral power source [15]. Regardless, autonomous mobile microgrid deployment has not been specifically addressed in the literature, outside of prior work by Michigan Technological University [16, 17, 18, 19, 20, 21].

Docking is a process that might be generally described as the alignment and mating of a vehicle with a set of capture surfaces on another vehicle or fixed infrastructure. Applications of docking vary from spacecraft in a micro-gravity environment or under-actuated AUV's [22, 23, 24], through UAV-to-UGV collaboration [25, 26, 27], to mating of constituents of a UGV [3]. The docking task generally depends on alignment of the more-mobile constituent with an approach path, using some type of feedback sensor, and maintaining an approach trajectory within the constraints of the capture geometry. A relative lack of prior work addressing docking controllers for UGV's compared to other domains and vehicle types may be simply related to the

UGV being the "less-mobile" constituent that is being docked with, rather than actively controlled. The UGV docking controller will be subject to the UGV kinematic constraints, as well as disturbance by the ground surface. A common motivation for docking shared by many of these sources is recharging of a more-mobile robot by fixed infrastructure or a less-mobile, higher-payload robot. For the purpose of mobile microgrids, docking supports a wireless power transfer task, as well as wired electrical connection coupling, with the mobile microgrid resources docking to each other and fixed infrastructure.

Manipulation of hardware or objects in the environment that an autonomous vehicle operates in is a difficult and open problem [28]. Much work is done to replicate the interaction of humans with their environment by robots that are equipped with mobile manipulators. In some cases, this can be done fully autonomously, but often teleoperation assists general-purpose manipulators in tasks using human-specific tools [29], [30], [31]. Despite this valuable work, some tasks may be better served by re-designing common hardware to accommodate the constraints of the autonomous mobile system. Work exists which investigates the advantages of a purpose-designed connector coupling system for transmission of power and data to a payload via a UGV mounted mobile manipulator [32], with demonstration in an outdoor environment. Despite the similarities with a mobile microgrid use case, an important distinction is the requirement of an expensive and complicated robotic arm, as well as divergent power transmission requirements.

One specific task which supports mobile microgrid formation is power cabling between grid assets. Autonomous control of cable trajectory as it is deployed from a UGV requires measurement of the cable trajectory as it exits the vehicle. A simplified system to validate a visual cable tracking and feedback control method deployed in an indoor and obstacle-free environment is presented in [33]. In a different domain, a particle-filter was applied to manually-labelled frames depicting an underwater cable and used to proved steering feedback to a AUV [34]. Position estimation for the cable has been handled a number of ways, including a graphical heuristic for stiff cables [35], and a taut cable via current-feedback control with visually estimated endpoints [36]. Regarding hardware development for cable handling by UGV robots, a design for load-bearing cables has been presented in [37] and [38]. One group in particular stands out for deployments of a tether from the same UGV platform used by the Autonomous Mobile Microgrid project [39]. Their application and assumptions differ from the mobile microgrid cabling task quite substantially in an underlying assumption that the tether is load-bearing and is kept taut at all times, and there is human supervision of the tether-obstacle interaction such that the obstacles actually form rigid guides for the tether trajectory. For the mobile microgrid application, the tether should not be considered load-bearing, and should lay free from interaction with whatever obstacles populate the environment.

1.3 Contribution: A method for UGV docking, connection, and cabling for electrical power transmission

The contributions of this work constitute an architecture for electrical power transmission by autonomous mobile microgrids. UGV docking and cable deployment software algorithms, and hardware for electrical connection and cable management, has been deployed on Clearpath Husky UGV robots, in an outdoor operating environment. Each of these contributions is not sufficiently addressed in the literature. More importantly, the cable deployment task is relevant to diverse domains beyond the FOB or disaster response application, including persistent operation, underwater mining, or planetary exploration. Specific contributions and their venue of dissemination are as follows:

† Navigation and rendezvous of the autonomous UGV robots leverages Robot Operating System (ROS) tools as well as a visual feedback controller for docking. Docking robustness is validated in Monte-Carlo outdoor trials which identify a 92% docking success sub-region for mission planning, presented at the refereed 2020 IEEE Aerospace Conference [40].

† An "Adjustable Cable Management Mechanism" (ACMM) was designed to meet low cost, compact platform, constraints for powered deployment and retraction by a UGV of electrical cable subject to disturbance, with feed rates up to 1 m/s, published in the HardwareX Journal [41].

† A probe-and-funnel AC/DC electrical connector system was developed for deployment on UGVs, which does not substantially increase the cost or complexity of the UGV, while providing a repeatable and secure method of coupling electrical contacts subject to a docking miss-alignment of up to +/-2 cm laterally and +/-15 degrees axially, submitted to the HardwareX Journal.

† Design of a feed-forward/feedback control method, which utilizes visual estimation of the cable state, deploys electrical cable without tension, in the obstacle-free track of the UGV, as it transverses to connect power grid nodes. Control response to step-input UGV chassis velocities in the forward, reverse, and zero-point-turn maneuvers are presented, as well as outdoor cable deployment, in preparation for submission to the AIAA Journal.

Chapter 2

Autonomous Power Grid Formation for Surface Assets Using Multiple UGVs[1]

2.1 Introduction

Mobile microgrids consist of deployable Unmanned Ground Vehicle (UGV) robotic agents, equipped with electrical power grid hardware. These agents need to be able to autonomously associate to configure this grid hardware and serve infrastructure

[1]The material in this chapter was previously published in the 2020 IEEE Aerospace Conference.

loads. Mission motivations include disaster recovery, military forward operating bases, and planetary exploration or habitation. In general, loads are considered fixed, and constrained by the mission motivation. The UGV microgrid agents are positioned based on the requirements for their power generation sources, and some of the UGVs are dedicated to providing electrical connection between the sources and loads. As mission requirements change, or infrastructure is reallocated, these connections should be re-configurable.

Consider the construction of a lunar base, with power, science mission, human habitation, and vehicle infrastructure. An array of Kilopower reactors, shown in Figure 2.1, could be positioned and connected to a power bus by a fleet of purpose-configured UGVs [1]. Electrical connections between the power bus and infrastructure would also be completed by UGVs equipped with cable and connection coupling hardware. An additional aspect is automated recharging of science missions which are located proximate to the base, but are temporary or inconvenient for point-to-point cabling connection.

Investigation of methods for navigation, docking, and electrical connection coupling and cabling using these mobile microgrid UGV agents specifically for planetary exploration has been informed by work in the field that address similar tasks. Recent work demonstrated moon-analog rover autonomy for navigation and manipulation [32]. A mobile rover manipulator has been developed for planetary missions [42].

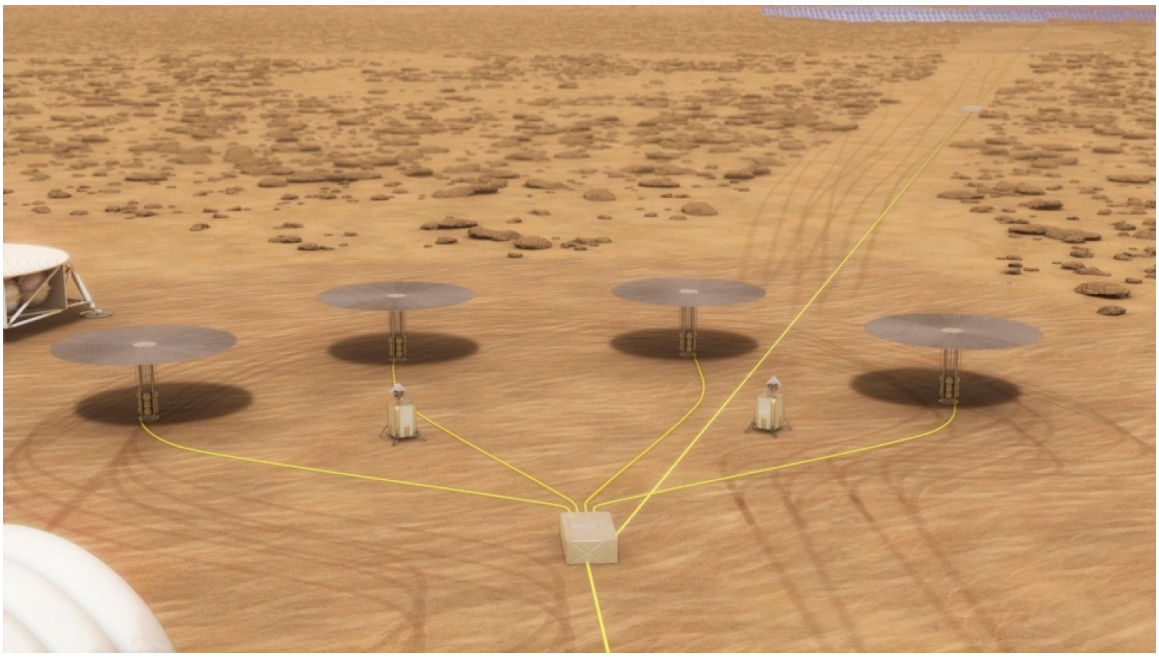


Figure 2.1: A Kilopower array would be autonomously deployed and interconnected by a number of UGV agents [1]

Docking and mobility analysis was performed for a tethered planetary rover [43]. A modular mission payload system was designed, including an integral power source [15]. Regardless, autonomous mobile microgrid deployment has not been specifically addressed in the literature, outside of prior work by Michigan Technological University [16, 17, 18, 19, 20, 21]. Work which addressed portions of mobile implementation of power grids includes real-time allocation of mobile generators [8], a design for containerized heterogeneous power sources [7], and testing of a communications scheme with operator oversight [14].

The contribution of this paper is the development of a comprehensive architecture for autonomous mobile microgrids, built on a foundation of available tools, with purpose-specific algorithm and hardware development for UGV docking and electrical

power transfer. The paper presents subsystem validation results in an unstructured environment, and presents a path forward for full system validation in a planetary-analog environment.

The organization of this paper is as follows: first is describes the general platform for all of the UGV agents, including the Robot Operating System (ROS) functionality. Next, the development of application capabilities is explored, including autonomous docking, power coupling and electrical cable deployment. Moving to subsystem and full system tests, a Monte-Carlo framework is applied to investigate robustness. The results of subsystem testing are presented. Finally, the conclusion presents future development work.

2.2 Robotic Platform

Clearpath Husky UGVs constitute the fleet of development vehicles for the work reported in this paper. These Husky UGVs shipped with a Mini-ITX computer, running an Intel i5 processor at 2.9 GHz and Ubuntu 14.04. Sensors include LORD MicroStrain 3DM-GX3-25 9-DOF IMU and NovAtel SMART6-L GPS receiver, as well as the SICK LMS-111 2D LiDAR. ROS provides the Husky UGV with a highly modular software environment to implement control architecture from low level hardware control, to modeling and simulation [44]. Numerous modifications have been made

Figure 2.2: Clearpath Husky UGV provides all-terrain navigation with sufficient payload capacity for demonstration power grid hardware.

to these vehicles and their descriptions constitute the large portion of this paper. These modifications include unique hardware, Figure 2.2, for the power grid: generation, conversion, switching, and transmission, as well as hardware which supports the autonomous robotic mission: perception, communication, and manipulators.

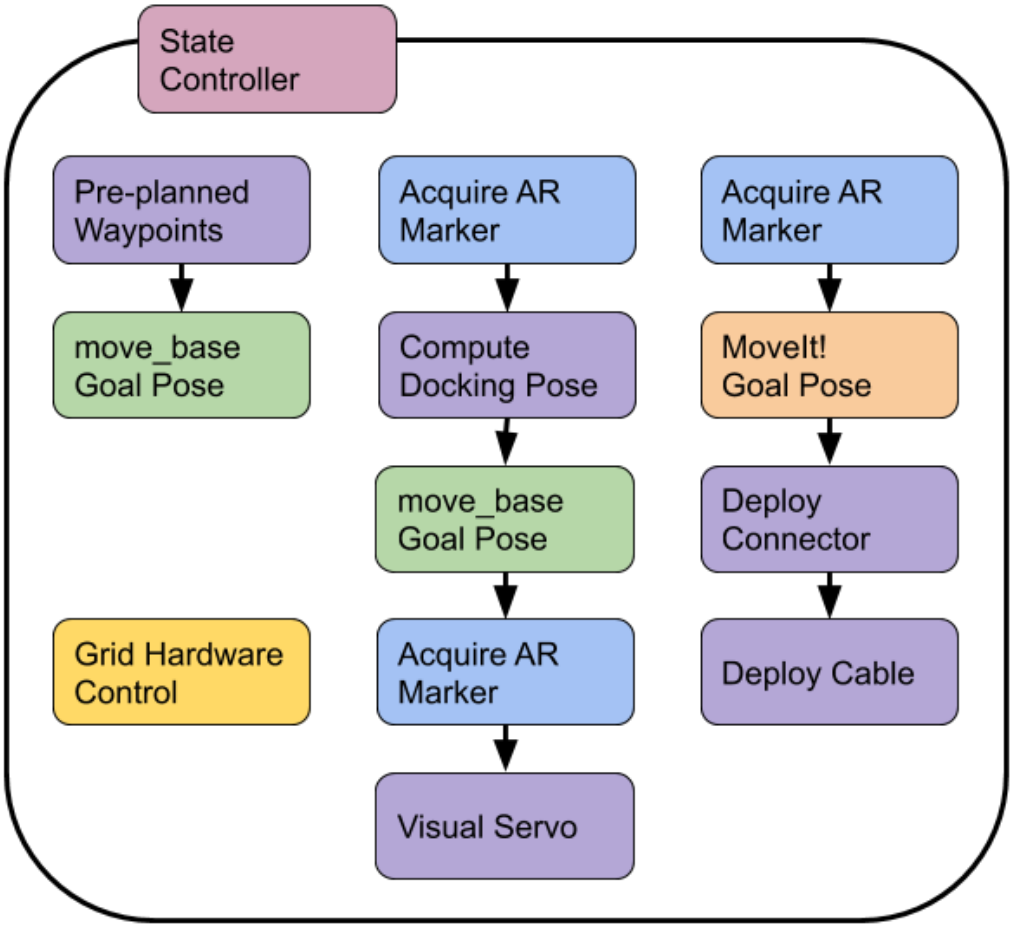


Figure 2.3: Control states for the UGV can include waypoint navigation, docking, connection coupling, and cable management.

2.2.1 Microgrid Control States

The framework for robotic control of mobile microgrids follows Figure 2.3, which encompasses all the tasks facing a cabling agent. Other agents perform a subset of these tasks. First, an Unmanned Ground Vehicle (UGV) agent navigates to a pre-planned rendezvous waypoint, using GPS data to inform it's localization. There is some error inherent in GPS data, and depending on the disturbance to the signal,

the magnitude of the error can be on the meters scale. A controller must locate connection targets, indicated by Augmented Reality (AR) markers, in the region of the rendezvous waypoint and proceed to dock adjacent to electrical connection hardware. Then, the connection can be initiated. Depending on mission requirements, the agent may proceed to another rendezvous waypoint and form a different connection. If cabling between connections is required, the cable must be deployed in a fashion the does not entangle it in the robot or obstacles.

2.2.2 Waypoint Controller

ROS includes a set of packages known as the navigation stack [45]. These packages provide a rendition of many of the common simultaneous localization and mapping (SLAM) algorithms that all autonomous robots depend on. ROS is very modular, and there are standard API's for developing new methods within the stack. The navigation stack maintains two costmaps which locate obstacle data, including a-priori and perceived obstacles [46]. Dijkstra's method [47] plans an optimal path through a map populated with known obstacles. For smooth motion near obstacles and to account for dynamic obstacles in the environment, a timed elastic-band planner is adopted [48]. Odometry for the robot is provided by an Extended Kalman Filter with wheel encoders, IMU, and GPS data as inputs [49]. While the waypoint controller depends

largely on open-source community contribution, the remaining controllers for the microgrid behaviors have been developed in-house, and are reported on through the duration of this paper.

2.2.3 Inter-Agent Communication

A WiFi network was established to provide communication between the mobile agents and stationary loads. With the physical expansion of the micro-grid as well as an increase in assets, the existing hardware of the network was determined to be outdated and over-tasked. To account for this increased network strain a tri-band (dual 5 GHz, single 2.4 GHz) wireless router was implemented to provide an increase in operational network range as well as eliminating network congestion due to the increased load. This router serves as the control node for the network and operator. Each mobile asset will receive an upgraded dual-band WiFi dongle with a high powered radio and 4 high gain antennas to extend their range. The stationary loads are equipped with Arduino micro-controllers outfitted with Adafruit WiFi shields containing external high-gain antennas. ROS does not have a suitable multi-robot communication scheme. Ongoing work to implement ZeroMQ, which provides a robust communication framework that can be integrated to individual robot masters, paves the way for real-time mission plan updates.

2.2.4 Heterogeneous Power Sources

In order to extend the battery operational utility of the Husky UGV, attain uniform traversal performance in missions, and produce useful energy for other agents or load use, a 120 V AC 1000 W gas generator is integrated with the UGV system. An AC to 24 V DC switching power supply is integrated with this to provide power for use within the UGV. This same type of system could be extended to other UGVs to provide a 120 V AC or 48 V DC electrical bus for power transfer to loads or to replenish energy stored in other battery bank agents.

2.3 Docking Controller

Microgrid UGV agents need to be able to achieve accurate positioning adjacent to infrastructure or other robotic agents before initiating power coupling. This docking controller, second row of Figure 2.3, needs a continuous exteroceptive source of localisation data. A viable solution is to implement a visual feedback method using a calibrated [50] monocular camera and Augmented Reality (AR) tracking of fiducial markers [51]. A fixed-eye camera is installed on each UGV. A zero-turn pan method to acquires the augmented reality marker in the camera frame when needed during the docking maneuvers.

After the rendezvous waypoint is achieved, the docking controller checks to see if the AR marker is in the camera field of view. If it cannot locate the marker, it engages a pan controller until the marker is acquired. Once acquired, it calculates a docking pose offset from the marker and passes this pose (x,y,θ) to the ROS Navigation stack as a waypoint goal. This waypoint goal is more accurate to the true pose of the docking target than any pre-planned rendezvous waypoint. Once the waypoint controller completes, the UGV is relatively near to the AR marker, which presents the best estimate of its position. It is advantageous to recalculate the docking pose offset and generate an updated waypoint goal, Figure 2.4. Empirically, $n = 2$ repetitions is sufficient.

Finally, the agent visual servos to the final docking pose. The pose error, e_x and e_y, between the robot chassis pose and the docking pose in a $2-$D cartesian frame is calculated using the AR marker frame in ROS tf. A proportional feedback loop is engaged, where $\dot{x}$ and $\dot{\theta}$, the linear and angular velocity commands, are calculated

$$\dot{x} = P_x \; e_x \tag{2.1}$$

and

$$\dot{\theta} = P_\theta \; e_y \tag{2.2}$$

where P_x and P_θ were tuned empirically. This behavior executes until e_x drops below a threshold distance. At this time, the docking method exits and signals the mission

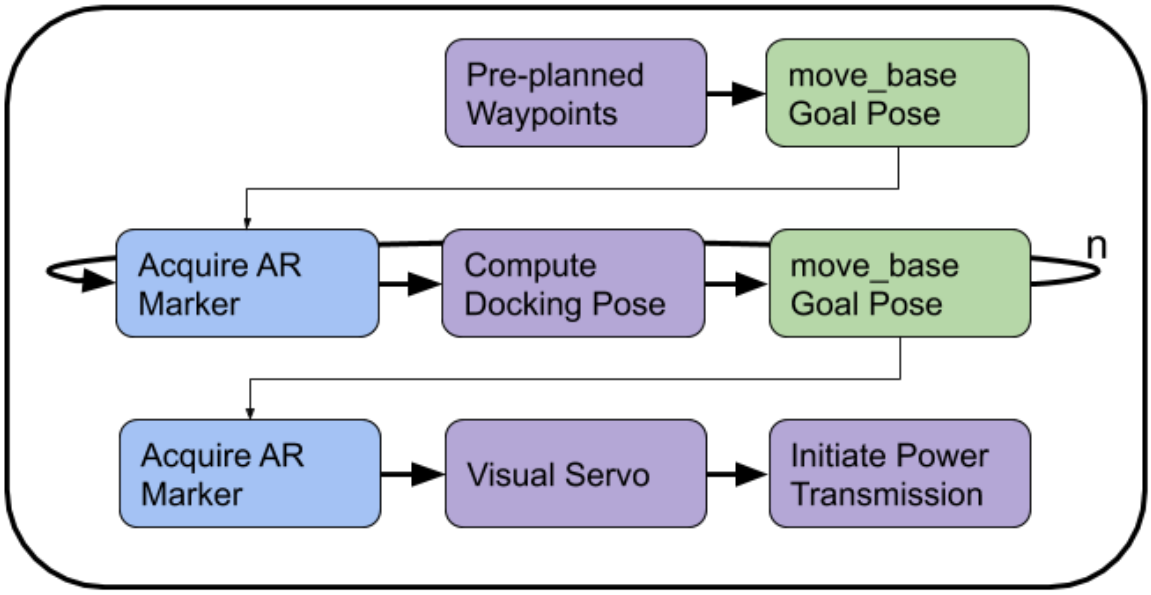


Figure 2.4: Docking control for wireless power transfer depends on both Canonical elements of the ROS move_base software, and custom algorithms.

controller to proceed to the next behavior.

2.4 Power Coupling

The critical task of the UGV microgrid agents is to transfer power from sources to loads autonomously. This section introduces two approaches, a connector with cable approach, and a near-field wireless approach. Cabled power transmission supports higher wattage, but is slower and more challanging to deploy. Near-field wireless power transmission is simple and fast to implement and re-configure, but the wattage is limited.

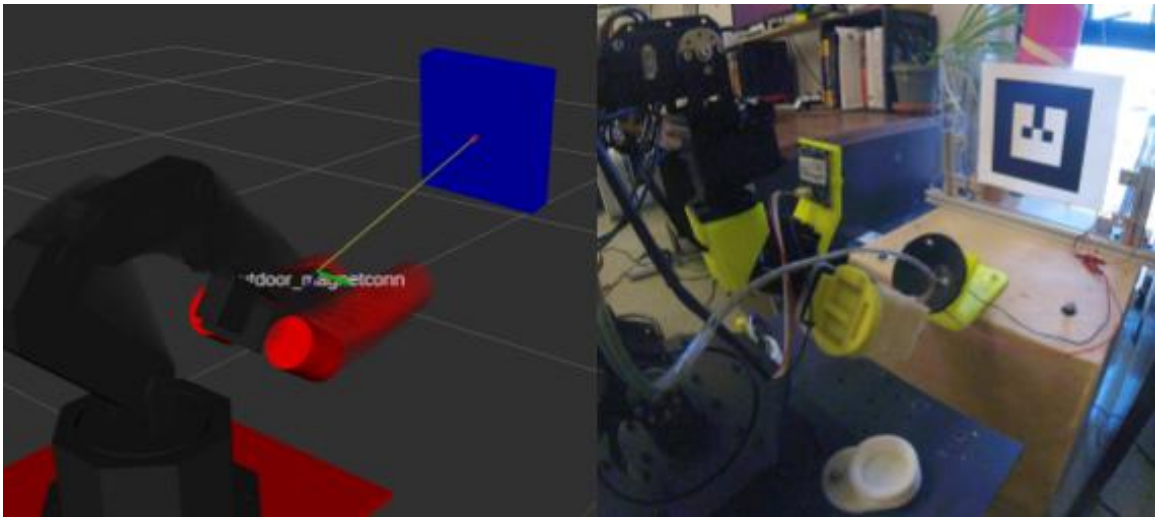


Figure 2.5: Connection coupling depends on visual feedback of the receiving connector pose using a single camera.

Figure 2.6: Primare coil hardware is mounted on the front of the UGV, and secondary coil hardware can be mounted to any load. When the coils are brought within sufficient range, power transfer is initiated.

2.4.1 Electrical Connection Coupling

An entry level 5−DOF robotic arm is installed on the cabling UGV agents, with a monocular camera providing a view of the workspace. ROS MoveIt! supports motion planning for manipulators in complex and cluttered environments with collision awareness [52]. A microgrid-specific end effector has been designed for the robot arm, which simplifies connection coupling. This end effector has been modeled in the ROS environment and MoveIt! leverages this model, as well as visual servoing from an Augmented Reality (AR) marker, to calculate a valid IK solution and perform the connection coupling, visualized in Figure 2.5.

Formerly, proof-of-concept connection hardware has been implemented on the cabling robots. As the project has progressed, a "clean slate" redesign of the electrical connectors and the connection actuation has become advantageous. This redesign will accommodate a power grid with requirements beyond the capacity of the former hardware. Commercial-off-the-shelf components suitable for autonomous electrical connection coupling are not available. Additionally, there is a gap in the price-point for multi-purpose robotic manipulators which results in affordable manipulators not being suitably robust to consider utilizing them to couple traditional electrical connections. Requirements for this system are a mission-specific 120VAC electrical connector designed to be coupled between two UGV's located in outdoor, unstructured

terrain. An associated actuation system should be tolerant to misalignment of the UGV chassis and provide a coupling action which results in a secure electrical contact with sufficient conductor surface area and conductor clamping force. The system must also provide a DC conductor pair, as well as a method of signaling a successful connection coupling event to the UGV controller.

2.4.2 Near-field wireless power transmission

Near-field wireless power transfer hardware was deployed on an agent which formerly did not have any connection functionality, Figure 2.6. This hardware was comprised of a pair of commercial-off-the-shelf 24 $Watt$ Litz wire coils with driver circuitry. It should be pointed out that the alignment tolerances which must be achieved for wireless power transfer are much more stringent than those for the robot arm connection coupling. For connection coupling, any final docking pose within the workspace of the robot arm is suitable. The wireless power transmission hardware, by comparison, requires alignment within a few centimeters absolute displacement of the centers of the coils, and angles not greater than 10 $degrees$. Refinment of the docking method resulted in the control flow shown in Figure 2.4. Figure 2.6 is a still from a demonstration video, and shows representative final alignment of the wireless primary and secondary coils, with the bright LED indicating power transfer.

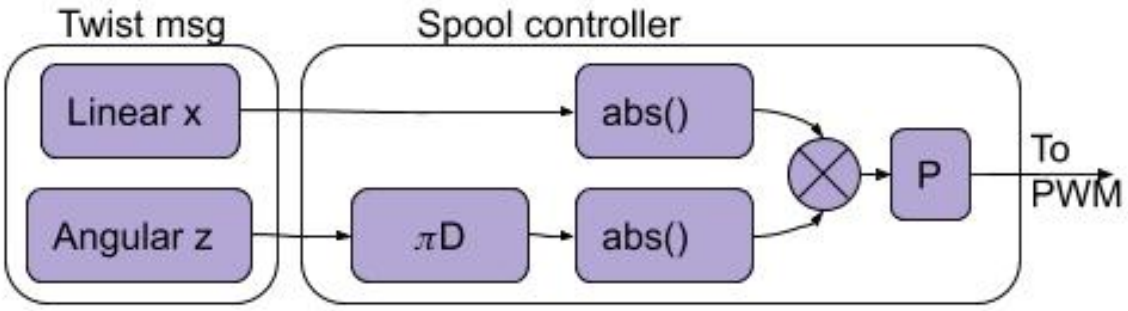


Figure 2.7: Deployment of the electrical cable leverages the UGV linear and angular velocity command, with known geometry characteristics, to trace the trajectory of the UGV as it traverses the operating environment.

Figure 2.8: Proof of concept electrical cable deployment system spools cable at a rate which tracks the path traversed by the UGV.

2.5 Cable Deployment

For power transmission to high consumption or permanent loads, electrical cabling must be performed autonomously. Cable routing and deployment in unstructured environments is an open problem. Most work in the area consists of a structured

problem with a-priori assumptions similar to this scenario for tracking the trajectory of a cable fixed between two UGVs in a lab environment [33]. In our development, the prototype exhibited a number of failure modes, including drag on the electrical connector from a taut cable, and entanglement with the robot chassis during maneuvers. A first prototype, which already mitigates the above failure modes, incorporates a powered feed dog to eject cable at a rate where the cable would "lay" along the trajectory of the robot.

The cabling controller, Figure 2.7, calculates the absolute velocity of the deployment fairlead, referenced to ground, using the linear and angular velocity commands produced by the robot's kinematic controller and the geometry of the robot. This is scaled by the appropriate coefficient, P, and output as a Pulse Width Modulation (PWM) signal to the drive servo on the feed dog. Figure 2.8 is a representative still from a demonstration video of the robot making a final docking approach for a connection, with cable deploying well out of the way of the wheels.

Unfortunately, the hardware at this time does not support reconfiguration of the microgrid, that is to say that once a cable has been deployed and connected there is no way to re-use that cable or node-to-node connection. A new cable storage, deployment, and recovery system is being designed, implemented on the UGV cabling agents, and tested. The system is designed to operate in rugged, unstructured terrain. The cable must be water resistant and durable, subject to snagging on obstacles

Figure 2.9: Validation tests were performed in an open environment outdoors, off road.

and being periodically driven on by the UGV. Retrieval of the cable may require a combination of maneuvers by the UGV and winching on the cable itself.

2.6 Validation Test Method

For this report on the progress of development, some of the hardware systems and controllers are ready for validation, while others are in the proof-of-concept stage. We present a test methodology that both assesses robustness of the controllers, and

Parameter	Value
Controller frequency	5 Hz
Recovery behaviors allowed	True
Global costmap resolution	0.1 m
Global costmap size	1600 m^2
Maximum range for an obstacle	5 m
Max volocity x	0.75 m/s
Max velocity x backwards	0.25 m/s
Max velocity theta	1.0 m/s
Acceleration limit x	0.5 rad/s
Acceleration limit theta	1.0 m/s
Minimum turning radius	0.0 m
Footprint model type	point
Footprint model radius	0.2 m
XY goal tolerance	0.05 m
Yaw goal tolerance	0.087 rad

Table 2.1
move_base parameters associated with the field trials reported in this work.

provides performance data for mission planning. Configuration parameters for open-source controllers are documented.

2.6.1 Subsystem Test Method

One practice for assessing robustness is to introduce a stochastic disturbance into a system over many trials and observe the control performance. This Monte-Carlo type method can be applied to the autonomous mobile microgrid controllers. Ideal mission configurations may not expose failure modes and edge cases, so a Monte-Carlo test regime should explore the limits of the capability of the system. Another benefit of the Monte-Carlo regime is that it can deliver stochastic performance measures which

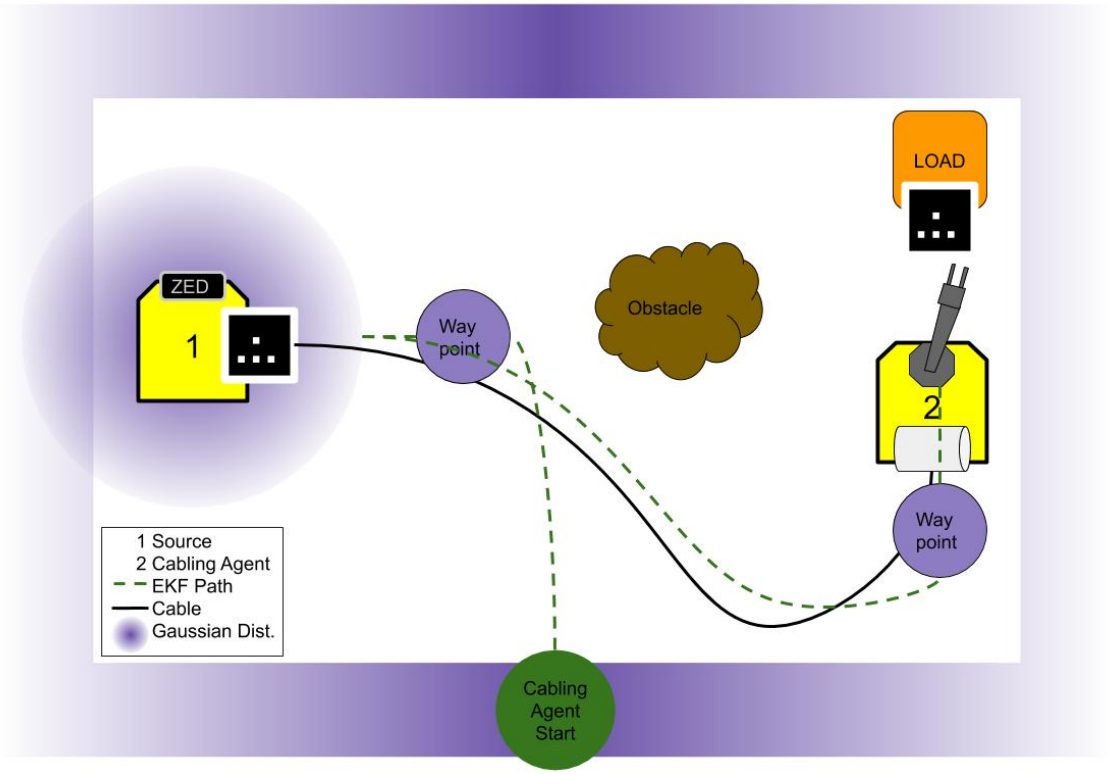


Figure 2.10: A Monte-Carlo type full system test is envisioned to demonstrate capability of the unified control architecture, with multiple UGVs participating

are important for mission planning.

The most fundamental application for this test method is traversal between a number of randomly distributed poses within an operating region. While evaluation of the open-source waypoint navigation controller is not novel, the evaluation grants insight on vehicle performance with a specific software parameter configuration. In this case, a $10mx10m$ subset of our operating region, Figure 2.9, was selected, and pose goals were distributed in a uniform random distribution throughout the region.

Docking of the UGV is a multi-step process, subject to different failure modes at

each step. Under ideal conditions, the rendezvous waypoint is offset from a docking goal by $1m$ along the target normal, with the docking camera facing the target AR marker. The validation test seeks to understand the effect of offset from this ideal pose along the target normal, orthogonal to this normal, and also in yaw orientation. The limits for the region of investigation are the camera resolution (maximum distance), and the assumptions of the kinematic planner regarding obstacle avoidance (minimum distance). A half-washer shaped region which satisfied these requirements was allocated about the target and populated with uniform random rendezvous waypoints. The minimum radius about the target face was 1 m, the maximum radius was 3 m. Angles were constrained to $1/2$ π either side of the normal vector pointing in to the marker. An automated test routine was developed which included a short reset traverse away from the final docking pose.

2.6.2 ROS move_base configuration

Optimum path planning, obstacle avoidance, and kinematic trajectory planning and control are provided by ROS move_base. Essential parameters for this application on the Husky UGV are provided in Table 2.1. It bears noting again that the UGVs are configured to use the teb_local_planner, rather than the default local_planner.

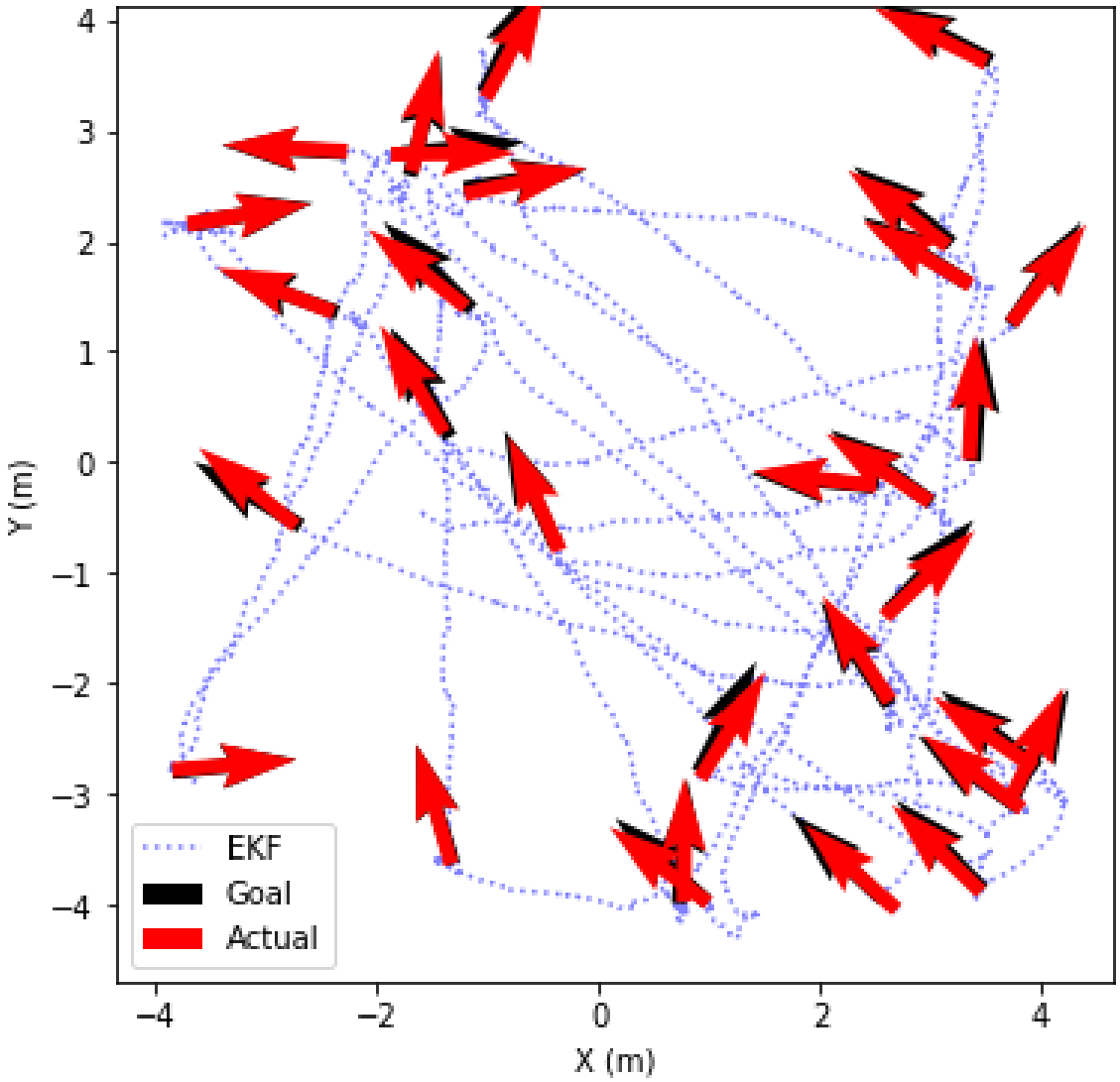


Figure 2.11: EKF filtered trajectories of the UGV transiting between random waypoints.

2.6.3 Proposed Full System Test Method

As development of the electrical connection and cabling subsystems proceeds, one goal is to validate these controllers before moving to demonstrating electrical grid control.

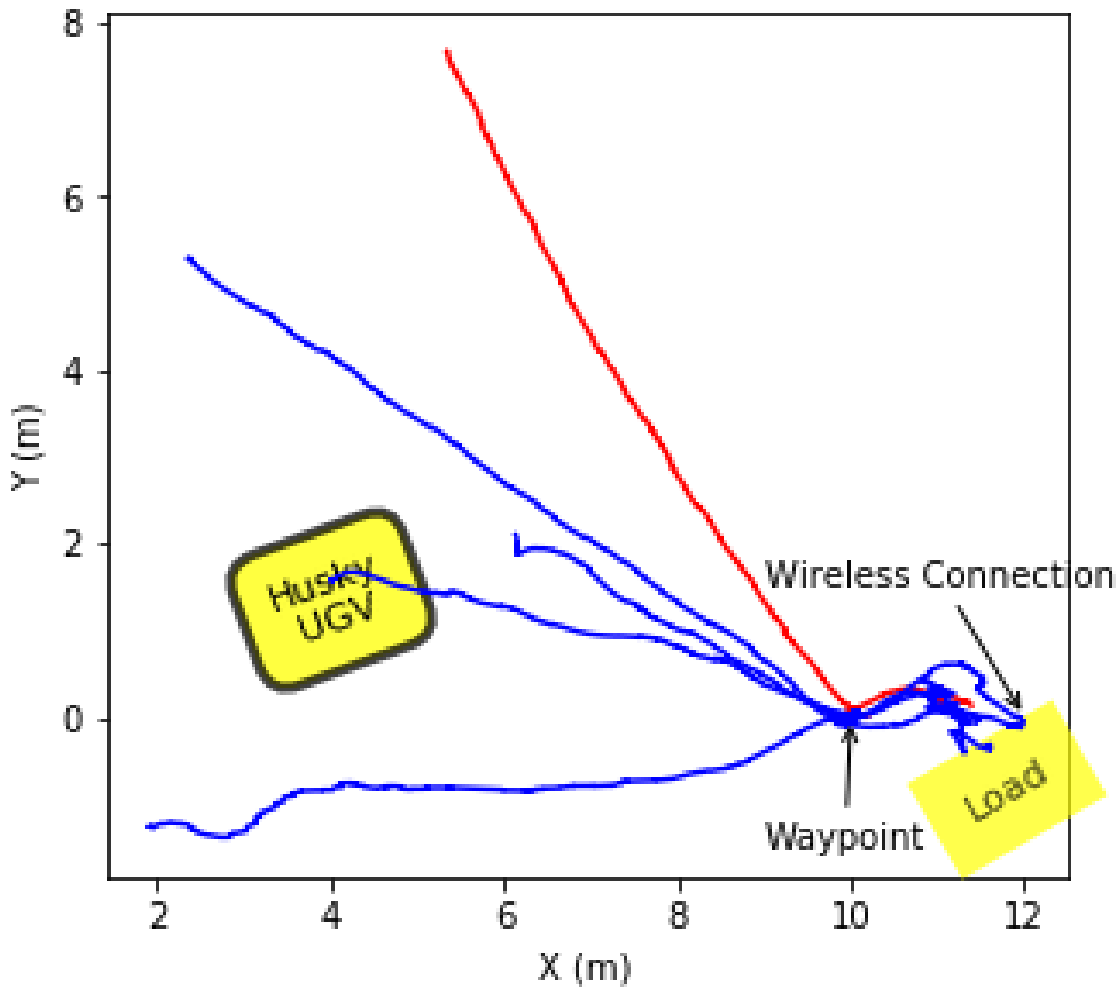


Figure 2.12: EKF filtered trajectories of UGV approaching and docking during early testing for near-field wireless power transfer. Successful approaches are blue.

We propose a Monte-Carlo scenario with a fixed infrastructure element which requires power, an a-priori map of fixed obstacles, a power source UGV, and a cabling UGV, Figure 2.10. The mission planner will randomly choose a goal pose for the power source UGV within the traversable region, then pragmatically generate rendezvous waypoints at adjacent to the power source and the infrastructure load. Both UGVs

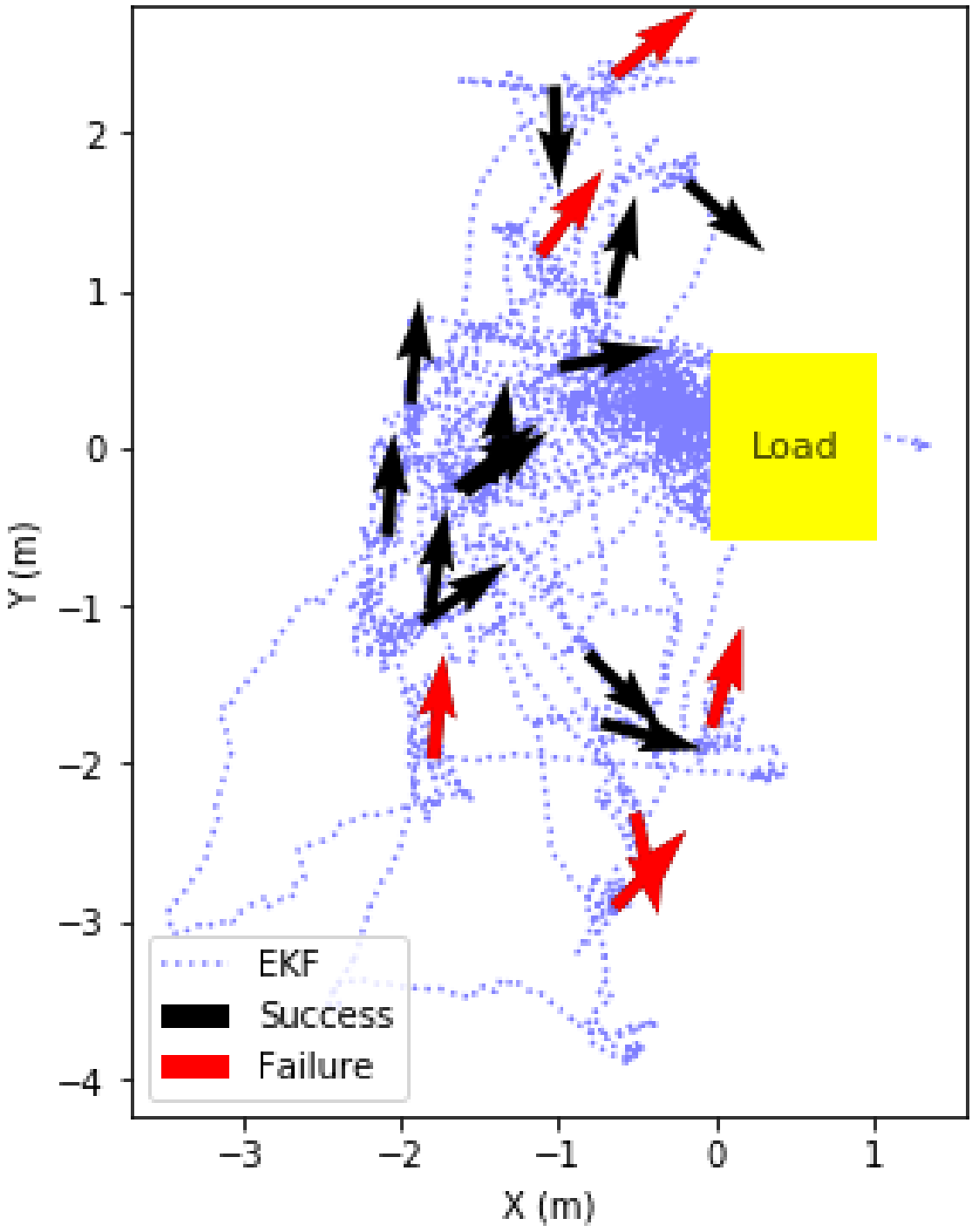


Figure 2.13: EKF filtered trajectories of UGV duing Monte-Carlo testing for docking and near-field wireless power transfer.

will originate from random locations around the parameter of the region, then proceed to their waypoints. The cabling controller will have to deploy and retrieve the electrical cable without entanglement in the obstacles.

2.7 Results

Waypoint navigation trajectories for 30 randomly generated waypoints are presented in Figure 2.11. The average traversal speed between waypoints, considering only their planned distance, was 0.257 m/s with a standard deviation of 0.115 m/s. The pose goal is represented by a black arrow. The pose attained is represented by a red arrow. There should be some small difference between the planned waypoint pose and the pose attained, as the planner is configured to consider the waypoint achieved when the linear and angular error of the actual chassis pose drops below their respective thresholds. This result is interesting in the context of a mission planner which needs to have aggregated actual performance of the robot in an anticipated operational environment. This type of test can be repeated in planetary-analog environments and for different types of terrain.

An example of a nominal rendezvous waypoint adjacent to a load, with an array of UGV origins, is shown in Figure 2.12. This type of outdoor demonstration resulted in 80% docking success rate, represented by this batch of 5 attempts presented here. The

position of the load in this demonstration was not considered ideal, with some angular and lateral displacement from the pose of the UGV at the rendezvous waypoint to the normal of the AR marker.

The final and most valuable subsystem test result is presented in Figure 2.13. Rendezvous waypoints are randomly seeded throughout a Monte-Carlo test region, and the UGV proceeds to dock for wireless power transfer. EKF trajectories of the UGV and the position of the load are indicated in the figure. Successful origin points are shown as black arrows, failures as red arrows. Of 18 docking attempts, 13 proved successful. The plot shows that docking failures are limited to rendezvous waypoints that are significantly displaced from the load, and especially those that have a very oblique viewing angle of the AR marker. One failure was attributed to impact of the UGV with the fixed infrastructure. The overall success rate of this difficult test was 73%. On average, it took 83 s from the time that the UGV achieved the rendezvous waypoint until it completed its docking approach, with a standard deviation of 43 s. It should be pointed out that the EKF trajectories represented impinge on the fixed load location. This should be expected, as there is error in the GPS data introduced in the filter. With careful mission planning, the rendezvous waypoint will be maintained in the area directly in front of the load power connection, and good docking success will be assured.

2.8 Conclusion

This paper presents a broad integrative effort to apply autonomous UGVs to the problem of mobile microgrid formation. This work has required custom hardware development, algorithms to support control of positioning of the UGV, and actuation of the power hardware. Although the work is ongoing, it has demonstrated a Monte-Carlo test of the waypoint navigation and docking controllers in an unstructured environment, with 73% docking success rate. This demonstration enables near-field wireless power transfer, and is prerequisite to electrical connection coupling for cabling between power sources and infrastructure loads. Future work on this project will focus on connection and cabling for power transfer. Connection hardware will be robust and purpose-specific. Cabling deployment hardware will include controlled retraction of the cable, and a visual cable tracking method which allows recovery of the cable without entanglement. As electrical connection and cabling hardware is completed, validation and demonstration of the full system will occur. The demonstration will include live electrical loads and centralized power grid optimization. The adaptive rescheduling of UGV tasks will respond to changes at the load infrastructure or UGV operability.

Chapter 3

Cable Deployment System for Unmanned Ground Vehicle(UGV) Mobile Microgrids[1]

3.1 Abstract

An ad-hoc autonomous mobile microgrid system requires electrical connections to be formed between physically separated resources. This work proposes the use of unmanned ground vehicles (UGVs) as the means to deploy the electrical cable that

[1]The material contained in this chapter was previously published in the HardwareX Journal in 2021.

creates these connections. This operation requires careful control of the cabling at variable speeds to avoid entanglement with the deploying UGV or obstacles in complex outdoor environments. Searching for a product that could supply the needed control and flexibility revealed a lack of compact and low-cost options. Existing options are very heavy ($>$ 100 lbs) and do not supply precision in their deployment. There is no commercial off-the-shelf option available for small-scale cable deployment operations with size and weight constraints. To fulfill the application requirements and to combat this deficiency, a custom design and build of an "Adjustable Cable Management Mechanism" (ACMM) was required. This ACMM provides a low cost, compact platform for powered and controlled deployment and retraction of different-sized cable under moderate loads, utilizing Commercial Off-The-Shelf components (COTS). Employing this design has enabled a variety of tasks that require distribution of electrical or data cables to be accomplished for small-scale projects. The goal of this paper is to give detailed design specifications of the ACMM and instructions on how to recreate it and calibrate it to be useful for tethering robots in various applications such as steep terrain, internet connection through tight spaces, or electrical connection between nodes for complex microgrids.

Specifications table:

Hardware name	Adjustable Cable Management Mechanism
Subject area	Engineering and Material Science
Hardware type	Mechanical engineering and materials science
Open source license	GNU GPL and CERN OHL
Cost of hardware	$USD434.23$
Source file repository	`https://doi.org/10.17605/OSF.IO/8WKJT`

3.2 Hardware in context

A microgrid is a system of interconnected, distributed energy resources, which creates a localized power infrastructure. Microgrid connections depend on the connection of two or more nodes with wired electrical connections. These nodes include energy sources and critical loads. An autonomous mobile microgrid employs many of the same resources as a traditional microgrid, but has the added benefits of mobility, self-organization, and re-configuration by autonomous agents. Unmanned Ground Vehicles (UGVs), equipped with energy generation, power control, cable coupling, and routing mechanisms, can establish this power infrastructure and transmission [21]. These autonomous mobile microgrids have many beneficial applications such as military forward operating bases, planetary operating bases, and disaster recovery [40, 53, 54], relieving humans from the time, effort, and dangers of manually establishing and maintaining these power systems. With the use of light detection and ranging (LiDAR), inertial measurement units (IMU), global positioning systems

(GPS), and camera vision, these mobile ground robots can search for a safe path to travel between the nodes, autonomously map their environment, and connect nodes in an electrical microgrid [20]. Using these onboard sensors, the UGV autonomously determines the shortest safe path and follows it. Once they reach an objective, the UGV deploys connectors to transmit power between the grid nodes. An example of two distributed UGVs and two loads in need of power are shown in Figure 3.1. In an indoor setting with few obstacles, deployed cable can be spooled out passively as the UGV travels. In less forgiving outdoor environments, natural or man-made obstructions such as branches, trees, and buildings prevent this from being a realistic approachOther work has been performed which deploys cable with a mechanism which maintains a taut cable between control points in the environment [38], but for this application cable tension causes impassable segments in the operating field for other mobile agents. These impassable segments limit other mobile agents from connecting to other loads or require more time and energy to navigate around them to make these connections. Too much slack deploys an unnecessary amount of cable and causes potential for entanglement with the UGV. This problem is solved by controlling the deployment rate of the cabling as a function of the speed the UGV so the cable is laid on the ground without tension, or slack, along the track of the UGV. Retracting the cabling is accomplished by retracing the path the UGV followed, recoiling as a function of UGV speed of travel. By requiring the cable deployment speed to be adjustable across the range of expected UGV speeds, the UGV can maintain

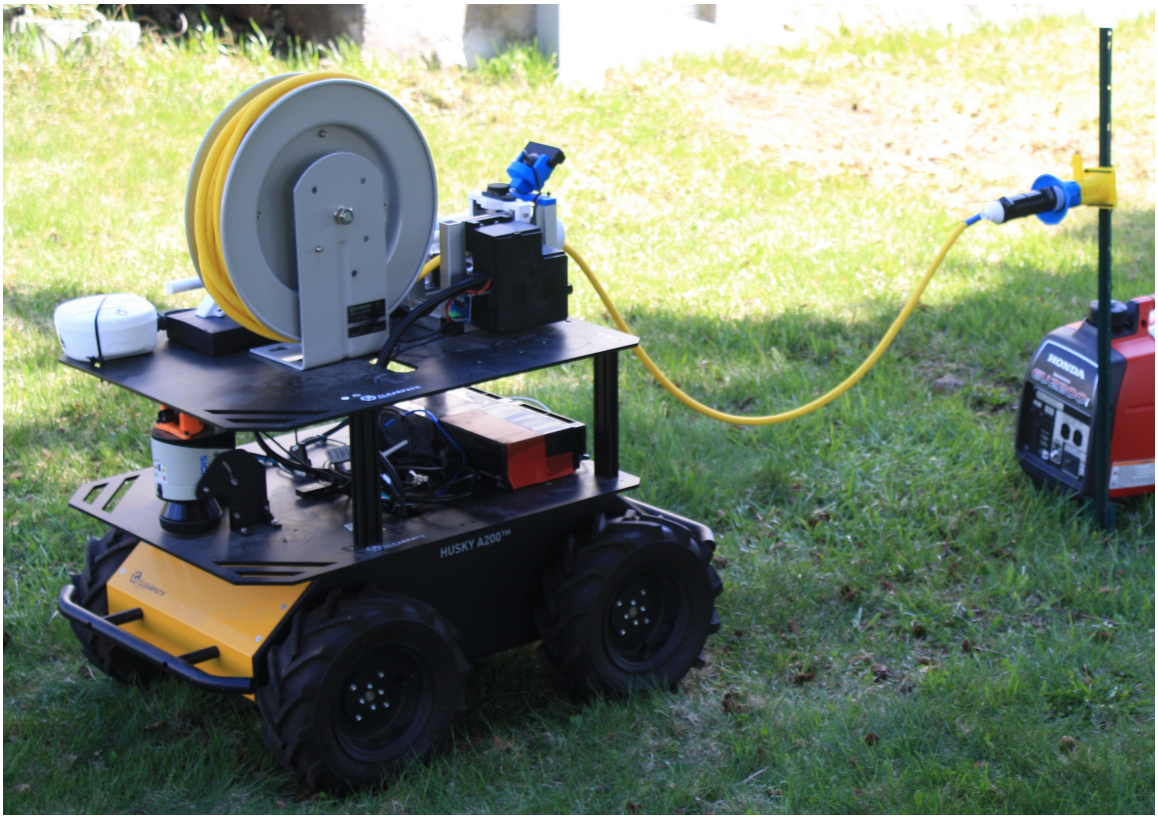


Figure 3.1: An Autonomous Mobile Microgrid UGV deploys a power cable to connect a genset power source to an infrastructure load.

its existing planner for smooth motions through the operational environment, with the cable drive system achieving the slack-cable constraint. Cable management hardware which performs these tasks, the "Adjustable Cable Management Mechanism" (ACMM), enables these cabling operations that prevent the cable from being pulled out of line, snagging on nearby obstructions, or caught in the UGV as it is deployed. The utility of the mobile microgrid system hinges on accurate and efficient control of cable deployment/retraction.

Microgrids require reconfiguration to meet dynamic load conditions when nodes will

need to be disconnected (shed) from the network, or when the grid needs to be reconfigured to accommodate different loads/sources. Once a load has been disconnected the retraction process can begin to recover deployed cable. Recoiling the cable autonomously allows for a single UGV to perform several connections in a different order or at a different location. Previous research has used tether systems as a means of guidance and used constant tension as a form of feedback in dimly lit environments where vision systems are not suitable, such as on the surface of Mars [55]. This however limits the terrain and complexity of the route. Our project focuses on retracing a cables path by following the steps used to create it and recoiling the cable along the way such that the cable is not allowed to tangle on obstacles. Other projects have utilized internal recoiling systems that are entirely self contained and use a series of gears to windup and release a tether from its storage reel. This solution, while effective for specific cable/ tethering needs, does not offer interchangeable tether options and depends on a complex and expensive set of gears to operate [55]. A similar design of a tether control system, used for lowering exploratory robots into Volcanoes, uses a feed through design with friction based attachment to the tether. This design was effective in controlling the release and retraction of the tether [56]. NASA's Jet Propulsion Lab (JPL) has investigated a design for jumping robots that utilize recoiling tether with precision tracking of its movement [57]. An encoder is connected to the feed wheel to track the remaining and deployed tether. The ACMM follows a similar method of tracking by integrating the rotations counted by its stepper motor,

allowing a precise measure of the deployed length of tether at all times, but contrasts from this work by imposing the speed control only on the tether rather than on the motion of the chassis. All the prior designs mentioned have a common similarity, they all appear to have been custom-designed to fit the user-specific application. It is likely that much time and resources could have been saved if complete, descriptive, open-source designs were available as a reference and inspiration. This is part of the goal in sharing the approach in this design.

Commercially available options for motorized spools exist. In majority, they are large industrial spools made for very high power applications and use brushed DC motors to quickly spool up and down cables. While these products are conducive to industrial operations, they are expensive in small quantity and too heavy for deployment on research-scale UGV's, although they could be utilized for full-scale autonomous vehicles. Our goal is to deploy 10-gauge electrical cabling from an unmanned ground vehicle, the Clearpath™ Husky. Limitations exist on the amount of cable available to connect the nodes. A cable capable of transferring 15 amps was selected for use with larger loads. We used a 75 foot cable spool to supply the UGV with this cabling. With such a limited amount of cable, it is important to know the exact amount remaining and to minimize usage. To determine the appropriate speed to move the cable, the turn radius of the vehicle and current directional speed are accounted for. The control on the ACMM inputs the velocity of the UGV and continually adjusts the cable extrusion motor speed.

3.3 Hardware Description

Attempts to find an existing hardware solution suitable for the scale of a mobile microgrid UGV revealed a lack of COTS options. Powered cable control systems were too large, too heavy, and/or prohibitively expensive. Powered cable control systems could weigh over $200lbs$ and cost $USD1,600+$. Creating a solution that provided precise control with a small footprint and mass was needed to fit on our project UGV's payload bay, leaving room for the necessary sensor array and cable spool. The UGV has a weight limit of 150 lbs under optimal conditions; the ACMM cable extruder is a lightweight 5 lbs to keep the UGV unencumbered. The cable spool is the largest contribution to weight, with cables suitable for specifications weighing over $50lbs$ for $70ft$ length. Our solution requires an $8inx5in$ footprint in the payload bay, runs off a $24V$ power supply, and can produce $15lbs$ of retraction force on a variety of wire sizes. It consists of mostly $3D$ printed components and low-cost materials, making it accessible to many users. This lightweight platform can be mounted on any surface and can be used with many types of retracting spool systems. The type of cable will determine the type of spool needed; this system is intended to work with a spring-powered spool as described in the Bill of Materials. The following diagrams detail the components arrangement for assembly. The sections are broken into the four parts for better viewing. The complete assembly is shown in Figure 3.2, with an exploded view of the assembly shown in Figure 3.3. Figure 3.4 shows the COTS

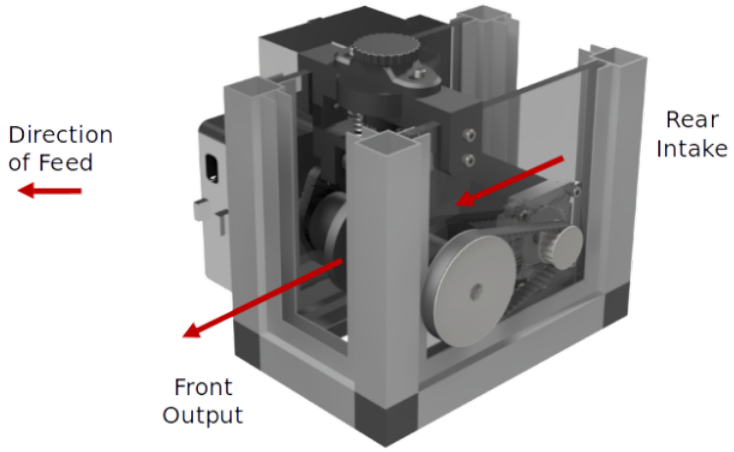


Figure 3.2: Complete assembly of the ACMM

cable spool which is integrated with the ACMM.

The cable spool in Figure 3.4 was purchased from Northern Tool and has a spring-powered recoil which reaches $15lbs$ at full extension of $75ft$. This spool helps maintain proper cable management, adding to the tension when recoiling and eliminating cord tangle. Other COTS variants of this product could be substituted dependent on local suppliers or availability.

Primary advantages and features of the ACMM are:

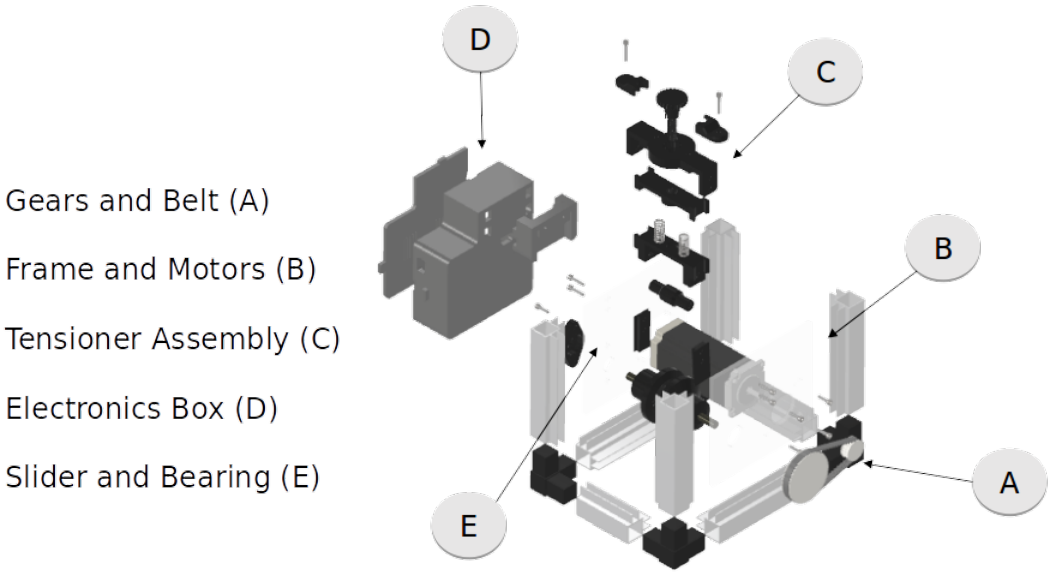


Figure 3.3: Full assembly explosion and subsections.

Figure 3.4: COTS cable spool

† Precise control to $+/-1\ in/70\ ft$

† Speed adjusts to the deploying vehicle up to a top speed of 3.3 ft/s

† Small footprint, $< 1\ ft^2$

† Applicable to cable gauges $8-15\ awg$

† Significant cost savings over comparable motorized cable reels

† Compatible with a range of cable reels and sizes

† Scalable and customizable to serve different applications

† Cost reduction of approximately $1000 compared to similar powered cable spools

The ACMM works by applying a large force to the cable, trapping it between the drive roller attached to the motor and to a free roller which is pressed down by a one-way tightening bolt adjusted from the top. Tension can be adjusted until the cable does not lose traction. The friction on the cable is dependent on the tension between the spur roller and the free roller which clamp the cable. The spur roller is rotated with the geared-up motor by way of a keyed $10mm$ drive rod. To adjust the tension, a bolt is turned to increase the clamping pressure. Because a stepper motor is used, a holding torque exists while activated, preventing the cable from being deployed inadvertently. Proper adjustment ensures no slip occurs between the cable and spur

roller, enabling precise accuracy in feed velocity. It is therefore very important to manually adjust the tension during commissioning so that slipping does not occur.

3.3.1 Electronics

The electronics system for the ACMM is composed of 4 parts:

† NEMA 24 stepper motor

† DMT542T stepper driver

† Arduino Uno

† Rotary Encoder

The Nema 24 stepper motor runs on $24V_{dc}$. The associated motor controller must be tuned to the appropriate PWM settings. The controlling software is written using the Arduino language. This motor can provide a $15lb$ holding force to the cable. Control software transforms the linear and angular velocity of the UGV to a linear velocity request for cable extrusion at the stepper motor. Because step rate is deterministic up to the rated slew speed of the stepper motor, the extrusion rate is known with very high accuracy and open loop control for the motor velocity command is sufficient. The system is connected to an Arduino Uno microcontroller. This microcontroller

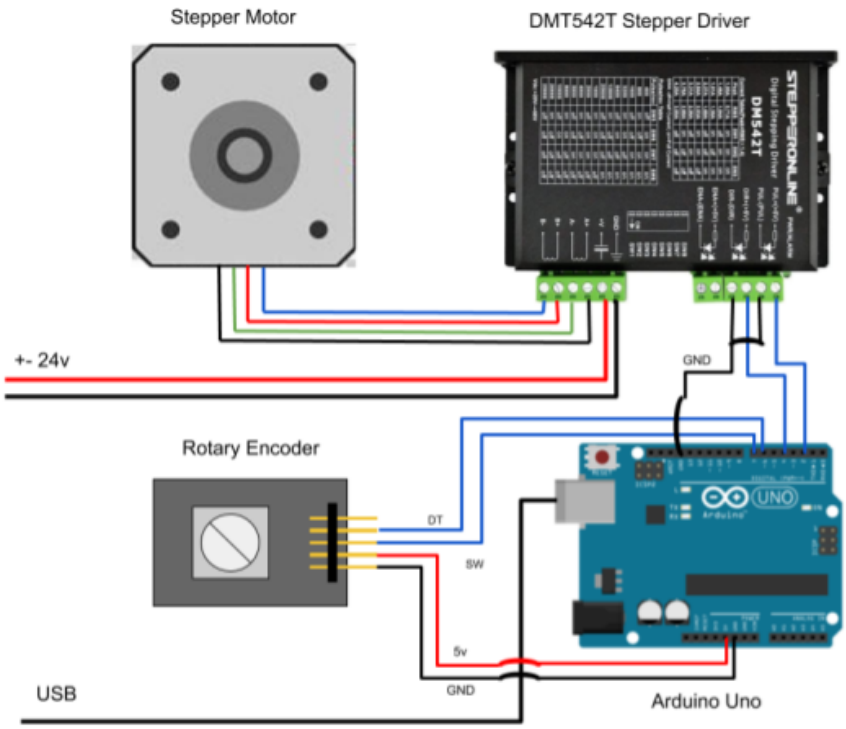


Figure 3.5: Wiring Diagram

was chosen for the well-established open-source software support for the hardware drivers used in this design. Anyone considering adapting this design to their needs should evaluate which microcontroller will provide sufficient computational power and peripheral capacity for all of their hardware subsystems, as well as the constant advancement of capability at a given price-point. The Arduino is wired into the motor controller with input from an encoder for manual speed adjustment, and a mode selection switch, Figure 3.5. The electronics are housed opposite the gears in the printed electronics enclosure.

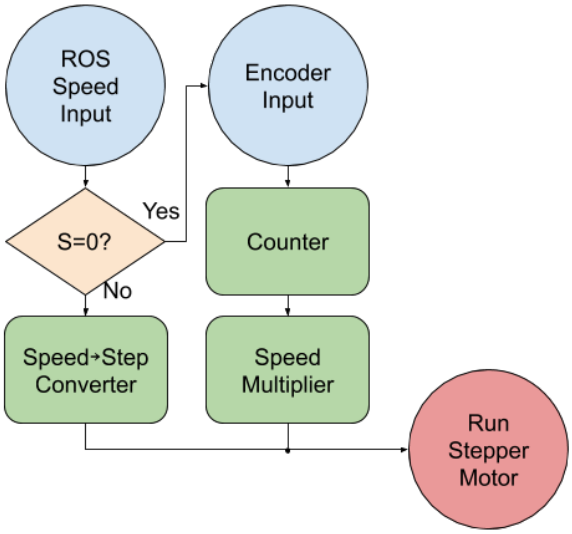


Figure 3.6: Function Block diagram of ACMM control code.

3.3.2 Control Code

Provided in the repository is control code for the ACMM, as an Arduino .ino file. The code accepts speed set-point inputs from both ROS and an encoder. The speed set-point is converted into the steps-per-second rate required by the stepper motor controller. If there is no ROS input, the ACMM can be controlled manually through an encoder. The operating principle behind the encoder code is the use of the difference in state generated by the rotary encoder to increment a counter higher or lower. This counter sets the speed of the stepper motor via a preset speed multiplier and the counter's magnitude. It is important to note that the acceleration of the stepper motor and its maximum speed fixed parameters in the code which cannot be adjusted in real-time. Figure 3.6 is a function block diagram for the control code.

3.4 Design files

To reduce the costs, weight, and complexity of construction, many of the components are 3D printed out of PLA filament using a Cetus MK3 3D Printer. The .STL files and drawings for reference are available as supplemental material with this article. See the 3D Printed Components BOM. The components that were not 3D printed have been selected commercially and purchased in majority through McMaster Carr.

Design filename	**File type**	**Open source license**	**Location of the file**
Bearing Block	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Box Lid	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Box Mount	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Electronic Box	STL	CERN OHL	10.17605/OSF.IO/8WKJT
plate 1	DXF	CERN OHL	10.17605/OSF.IO/8WKJT
plate 2	DXF	CERN OHL	10.17605/OSF.IO/8WKJT
Roller Housing	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Roller	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Screw Mount	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Screw Retainer	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Screw	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Slider Rail	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Spring Carrage	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Spur Roller	STL	CERN OHL	10.17605/OSF.IO/8WKJT
Speed_Control	INO	GNU GPL	10.17605/OSF.IO/8WKJT
extruderWiring	PDF	CERN OHL	10.17605/OSF.IO/8WKJT

The *Box Lid*, *Box Mount*, and *Electronics Box* comprise the enclosure for the stepper driver, Arduino, and associated wiring and components. Arduino software is provided in *Speed_Control*, with *extruderWiring* diagram for completing the electrical system. CNC cut panels, *plate 1* and *plate 2*, align the mechanical components. The friction

adjustment on the drive pulley consists of 3D printed components: *Screw Mount, Screw Retainer, Screw, Slider Rail, Spring Carrage* and *Spur Roller*. The drive pulley includes *Roller Housing* and *Roller*, with the stepper motor supported by the *Bearing Block*.

3.5 Bill of materials

#	Component	Qty	Unit Cost (USD)	Total Cost (USD)	Material Source	Material Type
1	Stepper Motor	1	39.99	39.99	StepperOnline	N/A
2	Motor Controller	1	39.89	39.89	StepperOnline	N/A
3	10 mm shaft bearing	1	6.89	6.89	McMaster	Steel
4	2.5" Diam. Pulley	1	21.00	21.00	McMaster	Al
5	1.5" Diam. Pulley	1	9.89	9.89	McMaster	Al
6	12" Drive Belt	1	5.87	5.87	McMaster	Rubber
7	10mm x 1' Rod	1	2.01	2.01	McMaster	Al
8	0.25" x 1' Rod	1	1.81	1.81	McMaster	Al
9	3" 8lb/in Spring	2	varies		Hardware	Steel
10	12" x 12" x 1/4" Sheet	1	15.89	15.89	McMaster	Polycarb.
11	4' Double Corner Slotted Rail	1	15.97	15.97	McMaster	Al
12	3way 90° Elbow	4	6.50	26.00	Mcmaster	Plastic
13	M4x18mm Q. 100 Socket Cap Screw	1	13.00	13.00	McMaster	Steel
14	80' Cable Reel	1	299.99	299.99	Northern Tool	Multiple

3.6 Build instructions

To construct the ACMM, begin by milling out the side panels on to polycarbonate sheets or similar 1/4" material. Print the parts listed in the 3D printed components

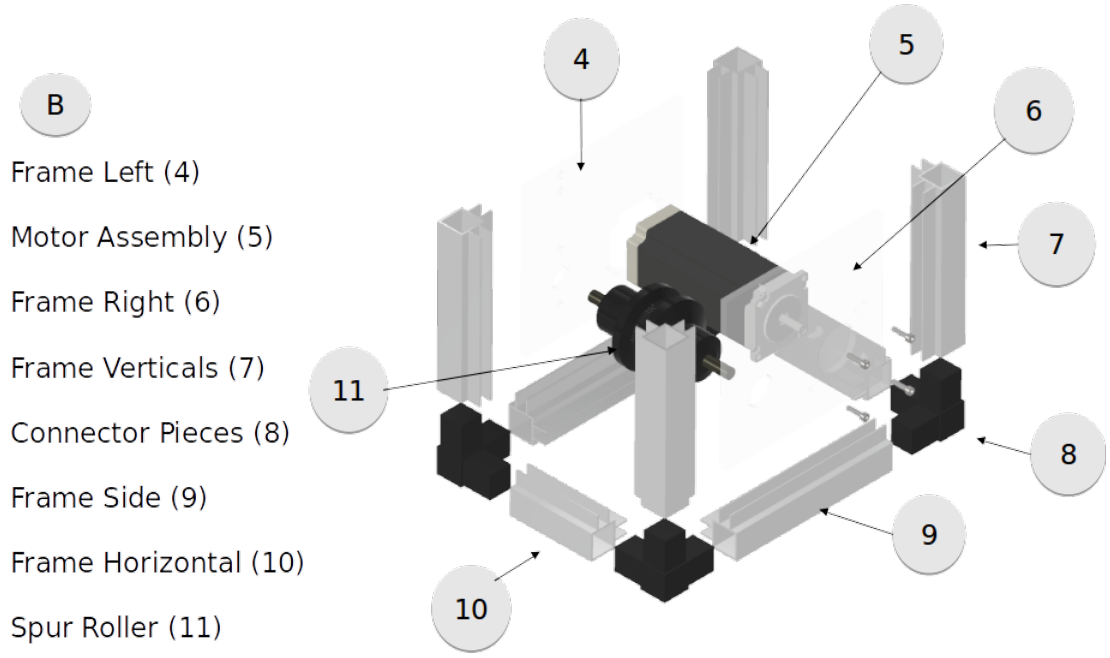


Figure 3.7: Frame and Motor assembly

BOM and use high fill settings.

The frame, shown in Figure 3.7, is composed of a slotted aluminum extrusion. To assemble these pieces cut pieces into three different lengths. Cut the extrusion pieces to the following lengths:

† 4” Frame Verticals - 4 parts

† 3” Frame Horizontals - 2 parts

† 4” Frame Sides - 2 parts

Cut 0.5” of the slotting at each of the side and horizontal extrusion pieces so they fit together at the corners. Hammer the extrusions into the corner pieces.

Add a key cut to the 10 mm rod (x3). The pulley system, shown in Figure 3.8, provides geared down power from the motor to the drive. The centers of both pulleys will need to be bored out to 10 mm before they will fit to the selected rod and motor. Connect the Larger pulley onto the 10 mm rod and fasten the screws (x4). Slide the keyed rod into the center roller (x5). Place the cut polycarbonate sheets into the slots of the frame (x6) Slide the motor into place, add loose bolts, then fasten the smaller pulley onto it (x7)

The bearing block, shown in Figure 3.9, holds the bearing that engages with the 10 mm rod running through the roller. This is fastened to the frame by two screws. Place rollers into (a), lubricate rails with lithium grease (x8). Install and fasten top support and sliders (x9)

The tensioner assembly, shown in Figure 3.10, is composed of 3D printed components that apply pressure on the cable held between the roller and the spur roller which is driven by the motor. The tensioner is adjusted by the screw, which tightens with a counterclockwise twist. The knurled surface of the roller will help maintain grip even in the event that the cable becomes damp. The slider rail should be lubricated with graphite powder or lithium grease. A short section of 0.25” rod is used between the roller support and the roller allowing it free spin. These components should be

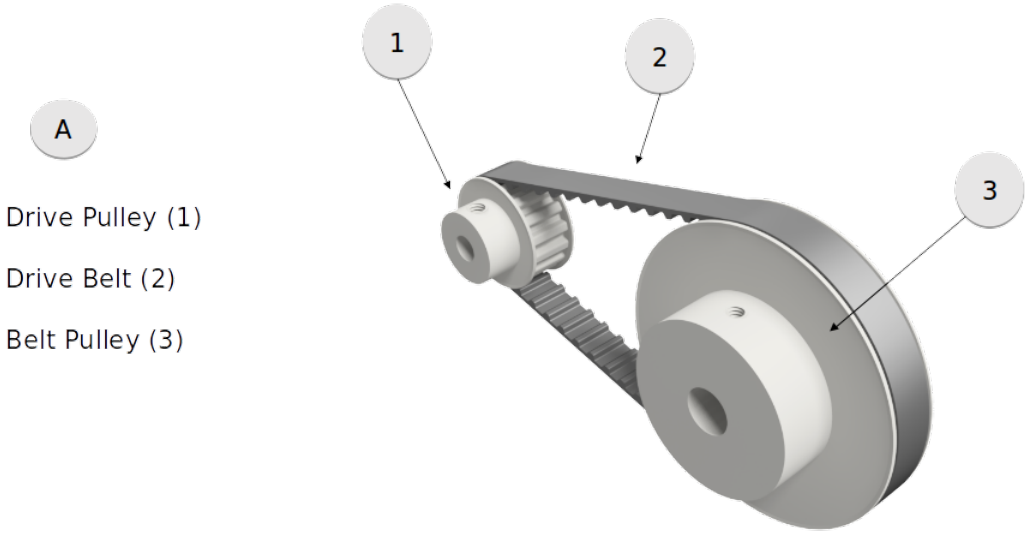


Figure 3.8: Pully and belt drive system

printed with high infill to ensure maximum strength.

The electronics box, shown in Figure 3.11, is designed to hold the needed components for the motor controller and arduino board. There is additional space for any added components and spacing for wire routing. The box anchor must be adhered to the Electronics box after the Box anchor has been placed. This is best done with a 5 minute epoxy.

Complete the wiring according to the diagram in Figure 3.5. To prepare the cable

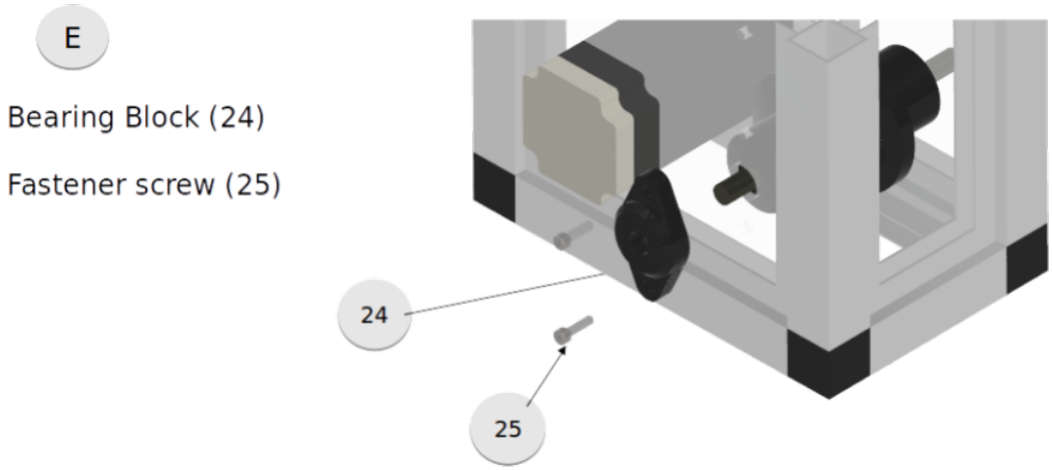


Figure 3.9: Bearing Block Assembly

spool for deployment, remove the locking mechanism located opposite the power input. This will prevent the spool from locking when the retraction process begins. Change out the power outlet end as needed for your project. Mounting of this component will be vehicle dependent, but the output path of the spool must align with the input of the ACMM.

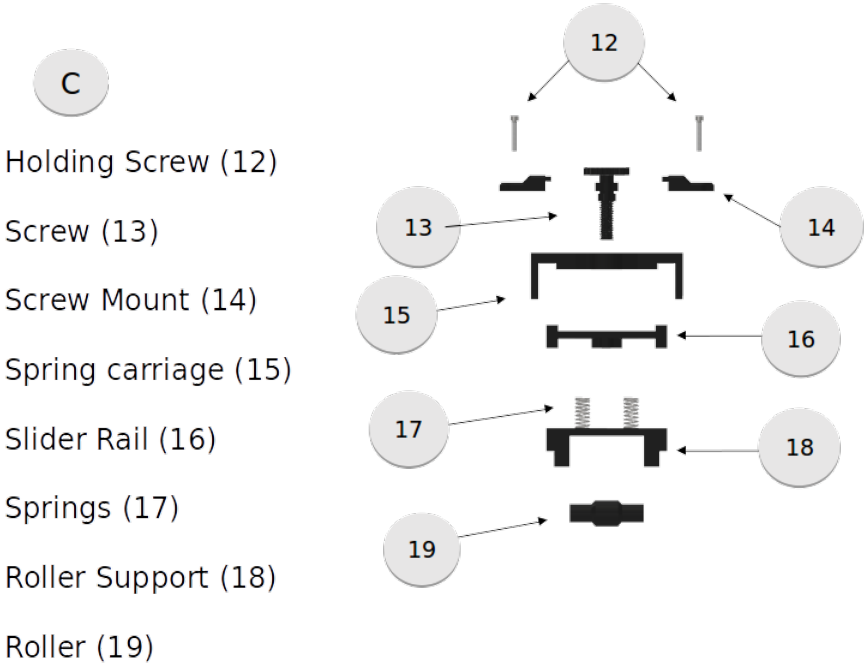


Figure 3.10: Tensioner Assembly

3.7 Operation instructions

While it is outside the scope of this article to define all the possible integration steps of this ACMM system with a host UGV, a brief handling of some contingencies is merited. This system is deployed in practice with a visual feedback controller for the cable trajectory as it exits the UGV. This controller runs in a Robot Operating System (ROS) high-level state machine, and can trigger a failure condition if it identifies that the cable deployment system is not maintaining its nominal state, i.e. the cable is

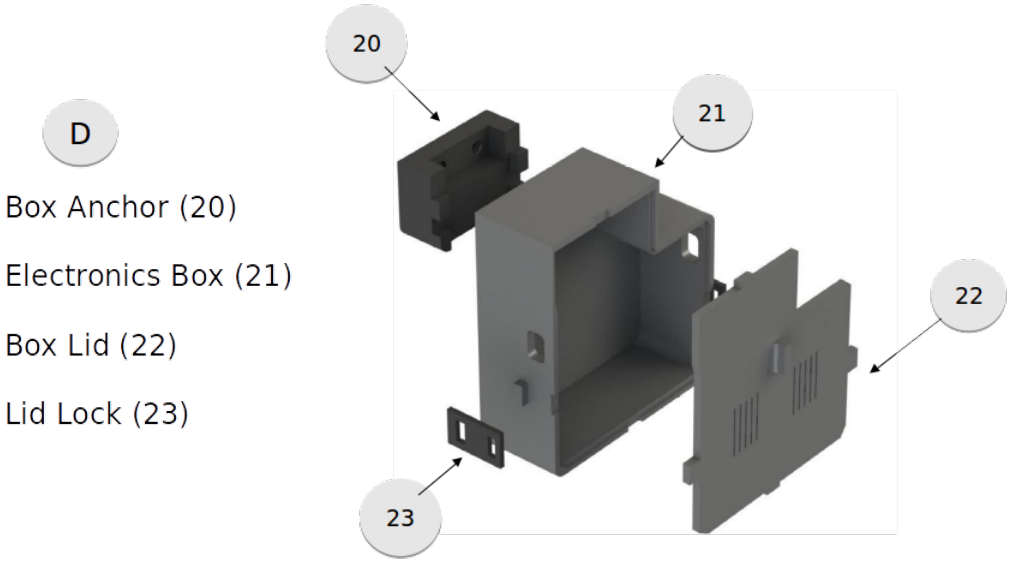


Figure 3.11: Electronics Enclosure

not paying out or retracting quickly enough to match the speed of the UGV chassis. This type of fault could occur from an unlikely entanglement or binding of the cable in the storage reel, or from a failure of the UGV navigation controller to sufficiently avoid obstacles, resulting in entanglement of the cable. In any of these conditions, the vehicle must be stopped, as well as the ACMM.

In regard to the fundamental operation of the ACMM subsystem, the following steps should be addressed:

† Securely mount the frame to the UGV.

† Securely mount base of reel behind the feed in location, leaving 3 inches of space for clearance.

† Feed the end of the desired cable to extrude from its reel through the back side of the extruder roll and tighten the tensioner to apply the desired amount of force. This tension required will depend on the cable size and reel used.

† Connect the extruder motor control to power and turn the encoder clockwise slowly to begin extruding the cable from the intake location (back) to the output side (front)

Safety Concerns:

† Keep loose hanging clothing and hair away from the extruder.

† Do not leave on when not in use to prevent motor overheating.

† Ensure the device is firmly attached before using.

† Keep hands away from the device while running.

3.8 Validation and characterization

Verification of the design is provided by a physical operation demonstration. Video of the demonstration is included with the supplemental material to this article, available at `https://doi.org/10.17605/OSF.IO/8WKJT`. This demonstration was performed in an outdoor environment. The UGV departs a power source location, and traverses through the region to rendezvous with an infrastructure load, figure 3.12. (The video is presented at 4x speed for brevity.) While the navigation, docking, and power connector coupling controllers are outside of the scope of this manuscript, some comments about the cable deployment are merited. The control system uses a feedforward-feedback controller, with the UGV velocity command as an input to the feedforward component of the control signal, and a visual cable angle estimate at the exit from the ACMM wrapped in a PID feedback loop. In the video, the feedback controller dominates when the vehicle is stationary, and then works with the feedforward control to provide disturbance rejection as the vehicle traverses, such as thick grass or small hills affecting the vehicle performance. Four types of maneuvers are captured in this demonstration video: forward progression while deploying cable, stopping the UGV and cable, retracting the cable while the UGV reverses back to its starting point, and a right turn while deploying cable on the ground in the track of the UGV.

Figure 3.12: A demonstration video of the ACMM on a UGV includes, from left to right, an infrastructure load, the cable following the track of the UGV through the environment, and a genset.

This design mixed standard off-the-shelf components with some 3D printed mechanisms, representing a trade-off between durability vs cost and ease of fabrication. The result is a design which is suitable for use in the robotic controls development iterations, with demonstrations and validation/robustness testing. For long-term deployment, one limitation of the design is that some of the 3D printed components in the tensioning assembly may need to be replaced by machined alloy components.

3.9 Conclusion

This ACMM fills an un-met need for a cabling deployment device for research-scale UGV robots. Valuable characteristics of the system include:

† Small size, weight, power consumption, and cost

† Deploys or retracts cable within the speed range of the UGV: up to $3.3 ft/s$

† Compatible control interface with the UGV software and hardware platforms.

† Meets electrical requirements for the microgrid AC/DC power system.

The system is installed and in use on the autonomous mobile microgrid robots.

Chapter 4

AC/DC Power Connector for Unmanned Ground Vehicle(UGV) Mobile Microgrids[1]

Abstract:

Autonomous Mobile Microgrids need a robust electrical connector for distribution of power resources deployed on Unmanned Ground Vehicles(UGV). This connector system should not substantially increase the cost or complexity of the UGV, while

[1]The material contained in this chapter has been submitted to the HardwareX Journal.

providing a repeatable and secure method of coupling electrical contacts. A probe-and-funnel connector design has been developed for deployment on Clearpath Husky UGVs, whose docking controller achieves a docking alignment precision of +/-2 cm laterally and +/-15 degrees axially. A flexible shaft material integrated into the probe mitigates this residual lateral and axial miss-alignment. Electrical contacts support transmission of AC 120 volts at 15 amps, and low voltage DC power up to 1800 watts. The design leverages 3D printing methods and commercial-off-the-shelf(COTS) components to optimize limited quantity production. Beyond the design files presented here for a UGV-to-infrastructure electrical connection, a UGV-to-UGV configuration is provided, as well as further insight for adapting this system for other applications. A set of connection trials is provided for validation of the system. This design fills the need for a purpose-specific and customizable electrical connector designed for power transmission by autonomous mobile robots, without resorting to expensive and complicated general-purpose manipulators and their control systems.

Specifications table:

Hardware name	AC/DC Power Connector for Mobile Robots
Subject area	Engineering and Material Science
Hardware type	Mechanical engineering and materials science
Open source license	GNU GPL and CERN OHL
Cost of hardware	$USD106.58$
Source file repository	Available in the article

Figure 4.1: Heterogeneous power source UGVs prepare to couple and form an AC/DC power bus.

4.1 Hardware in context

Autonomous Mobile Microgrids deploy Unmanned Ground Vehicles (UGV) to position and connect microgrid power hardware with islanded infrastructure. These mobile power resources include power sources, power conversion and distribution, and the interconnection hardware. Applications of this technology focus on human-denied environments, including military, disaster response, and planetary exploration [40]. This work focuses on the design of a connector for AC and DC power transmission by autonomous UGVs.

Commercial-off-the-shelf components suitable for autonomous electrical connection

coupling are not available. Additionally, there is a gap in the price-point for multi-purpose robotic manipulators which results in affordable manipulators not being suitably robust to consider utilizing them to couple traditional electrical connections. We need a mission-specific 120VAC electrical connector designed to be coupled between two UGV's located in outdoor, unstructured terrain, figure 4.1. An associated actuation system should be tolerant to misalignment of the UGV chassis and provide a coupling action that results in a secure electrical contact with sufficient conductor surface area and conductor clamping force. The system must also provide a low voltage DC conductor.

4.2 Hardware Description

Manipulation of hardware or objects in the environment that an autonomous vehicle operates in is a difficult and open problem. Much work is done to replicate the interaction of humans with their environment by robots that are equipped with mobile manipulators. In some cases, this can be done fully autonomously, but often teleoperation assists general-purpose manipulators in tasks using human-specific tools [29], [30], [31]. Despite this valuable work, some tasks may be better served by re-designing common hardware to accommodate the constraints of the autonomous mobile system. Work exists with investigates the advantages of a purpose-designed connector coupling system for transmission of power and data to a payload via a UGV mounted

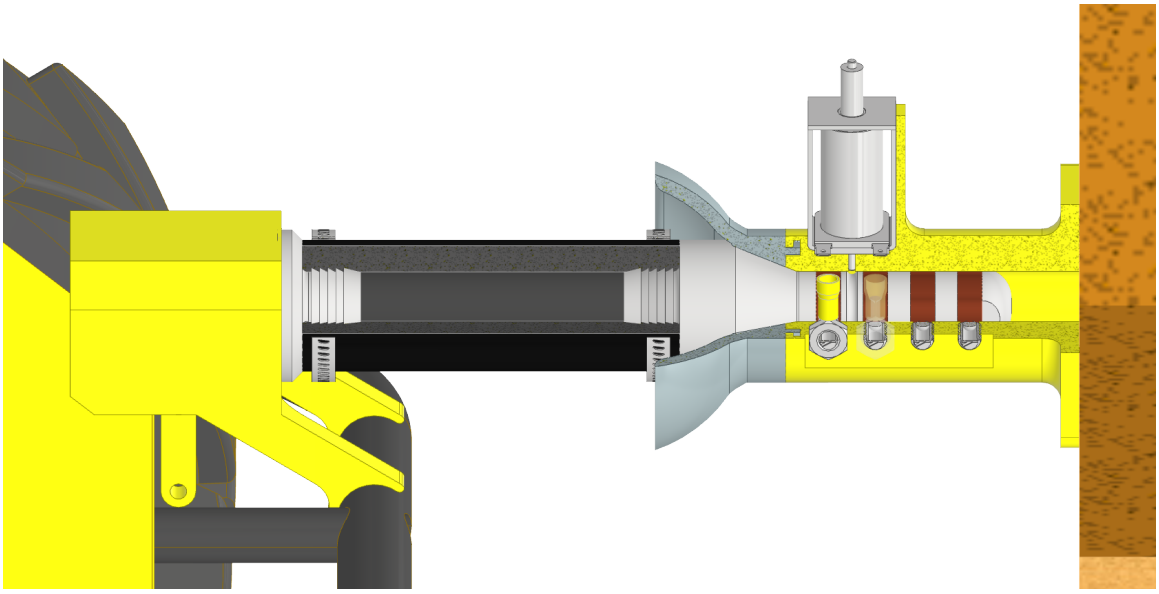


Figure 4.2: The AC/DC connector system enables power transmission between a UGV and stationary infrastructure.

mobile manipulator [32]. Another approach to power transmission for mobile robots is to employ a wireless power transmission method [53]. These interactions need to be repeatable and characterized, so that the robotic agents can fulfill optimized missions with temporal constraints [21, 54].

Electrical connectors for this project must be able to safely transmit 120VAC power at up to 15 amps. All elements of the connector design must accommodate these current and voltage requirements, including wiring, traces, and pads or prongs. In addition, the contact area and force between the mated connector surfaces must be sufficient to eliminate contact impedance at the connector. Strain relief from connector to electrical cable should be integral to the design. The UGV will dock adjacent to the target electrical connection, at this point, the connection system will

engage and align the two halves of the electrical connection and then securely latch the connection. Once the connection is latched, a signal should be sent to the robot controller to confirm the connection. The precision of the docking process is +/-2cm in any direction and +/-15degrees.

Inspired by the probe-and-drogue aerial refuelling method [58], a probe and funnel connector system was developed, figure 4.2. The probe consists of a linear array of conductor contact rings, mounted to a flexible material, figure 4.3. The probe engages a funnel which is sized to accommodate the expected lateral miss-alignment residual from the docking maneuver. After the UGV positions for a connection approach, either the vehicle chassis performs a controlled insertion, or a linear actuator can be used. The interior features of the funnel half of the connector pair include spring loaded electrical contacts and a latching mechanism. It should be pointed out that the linear array of contacts may be modified to support data transmission or a different number of electrical leads.

Primary advantages and features of this AC/DC connector are:

† Purpose-designed to accommodate the docking behavior of an autonomous UGV.

† Supports AC and DC electrical power transmission.

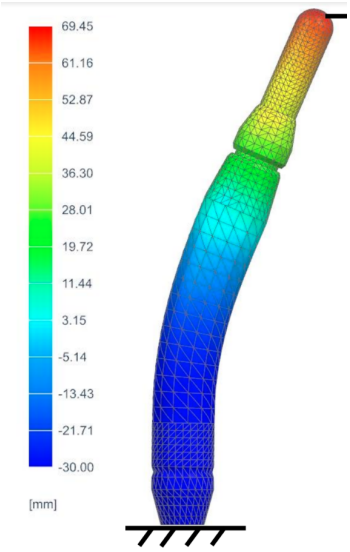


Figure 4.3: A flexible rubber element integrated in the connector complies to mitigate misalignment in the lateral and axial degrees of freedom during docking.

† Does not require an expensive multi-degree of freedom manipulator for connection coupling.

† Designed around inexpensive, readily available materials and components.

† Customizable for the end user power and/or data requirements.

4.3 Design files

To reduce the costs, weight, and complexity of construction, many of the components are 3D printed out of PLA filament using a Cetus MK3 3D Printer, with maximum infill. The .STL files and part drawings for reference are available as supplemental material with this article. See the 3D Printed Components BOM. The components

that were not 3D printed have been selected commercially and purchased in majority through McMaster Carr.

Design filename	**File type**	**Open source license**	**Location of the file**
funnel	STL	CERN OHL	available with the article
femaleJunction	STL	CERN OHL	available with the article
plug	STL	CERN OHL	available with the article
copper ring	PDF	CERN OHL	available with the article
cap	STL	CERN OHL	available with the article
sleeve	STL	CERN OHL	available with the article
ringRootSpacer	STL	CERN OHL	available with the article
sleeveDetent	STL	CERN OHL	available with the article
rubber	PDF	CERN OHL	available with the article
male adapter	STL	CERN OHL	available with the article
male box	STL	CERN OHL	available with the article
femaleJunctionBumper	STL	CERN OHL	available with the article

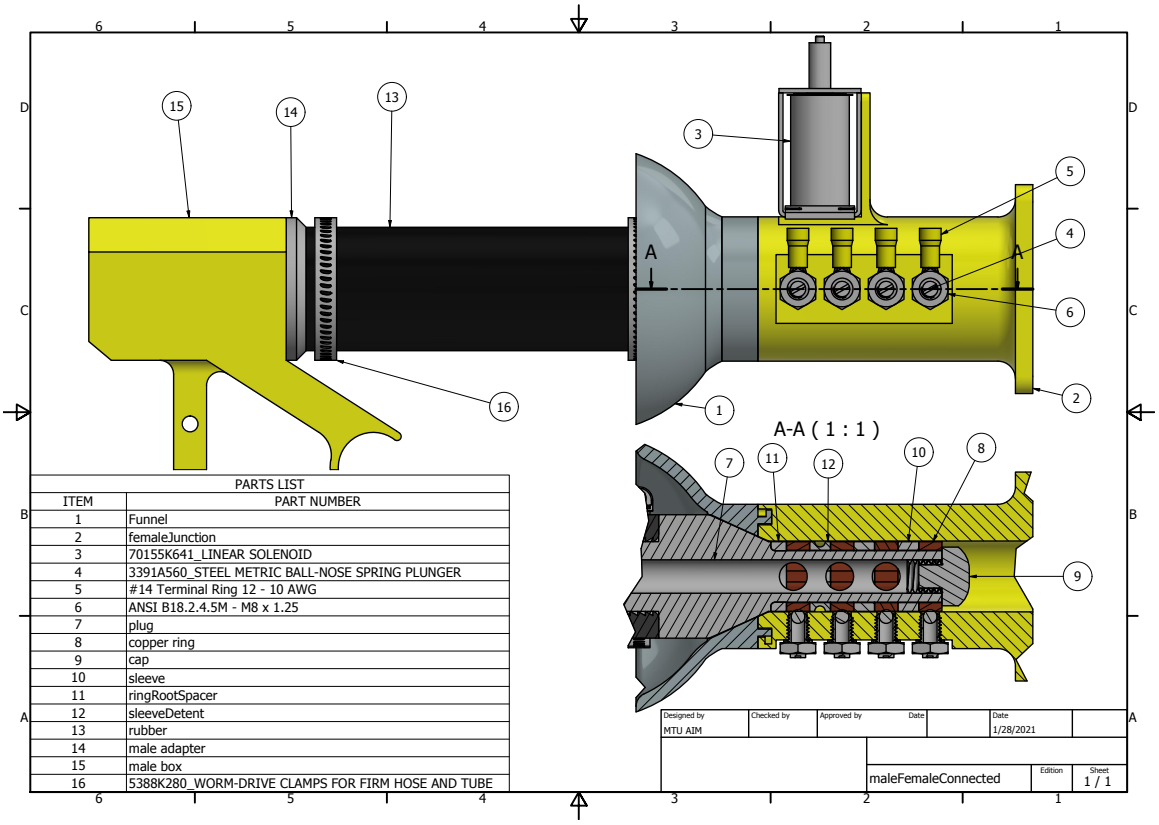
The *funnel*, *femaleJunction*, *plug*, *cap*, *sleeve*, *ringRootSpacer*, *sleeveDetent*, *male adapter*,*male box*, and *femaleJunctionBumper* files are STL type files, ready to be sent to a 3D printer slicing software and produced. The *copper ring* and *rubber* files are PDF type files containing the engineering drawings for those two components, which require manual fabrication with simple tools.

4.4 Bill of materials

#	Component	Qty	Unit Cost (USD)	Total Cost (USD)	Material Source	Material Type
1	funnel	1	2.02	2.02	amazon.com	PLA
2	femaleJunction	1	5.99	5.99	amazon.com	PLA
3	70155K641 linear solenoid	1	40.07	40.07	mcmaster.com	steel
4	3391A560 ball-nose spring plunger	4	4.80	19.20	mcmaster.com	steel
5	7113K223 ring terminal	4	0.39	1.56	mcmaster.com	brass
6	90690A055 M8 brass nut	4	0.66	2.62	mcmaster.com	brass
7	plug	1	2.12	2.12	amazon.com	PLA
8	8965K34 copper ring	4	1.43	5.72	mcmaster.com	copper
9	cap	1	0.13	0.13	amazon.com	PLA
10	sleeve	2	0.04	0.08	amazon.com	PLA
11	ringRootSpacer	1	0.02	0.02	amazon.com	PLA
12	sleeveDetent	1	0.04	0.04	amazon.com	PLA
13	87235K29 60A rubber	1	14.15	14.15	mcmaster.com	PU
14	male adapter	1	1.34	1.34	amazon.com	PLA
15	male box	1	2.29	2.29	amazon.com	PLA
16	5388K280 worm-drive clamps	2	0.89	1.78	mcmaster.com	steel
17	femaleJunctionBumper	1	7.45	7.45	amazon.com	PLA

4.5 Build instructions

This power connector design is modular and adaptable to different vehicles or applications. In this document, build instructions detail one configuration for connecting a UGV with stationary infrastructure. Additionally, the build instructions depict some



PARTS LIST	
ITEM	PART NUMBER
1	Funnel
2	femaleJunction
3	70155K641_LINEAR SOLENOID
4	3391A560_STEEL METRIC BALL-NOSE SPRING PLUNGER
5	#14 Terminal Ring 12 - 10 AWG
6	ANSI B18.2.4.5M - M8 x 1.25
7	plug
8	copper ring
9	cap
10	sleeve
11	ringRootSpacer
12	sleeveDetent
13	rubber
14	male adapter
15	male box
16	5388K280_WORM-DRIVE CLAMPS FOR FIRM HOSE AND TUBE

Figure 4.4: Locations of printed and COTS components, with BOM numbering.

alternate configurations that support other connection tasks. These resources should provide the reader with a starting point for their application. Safety considerations for the assembly of these components include a cutting operation using a band saw or small high speed rotary tool, the use of a small gas torch for soldering, and the use of a sharp blade for manual cutting operation. The hazards from these operations include, but are not exclusive to, lacerations, burns, damage to eyes, and inhalation hazards. The user must receive appropriate training regarding these hazards, and engage engineering controls and proper PPE to mitigate risk from these hazards.

Part count for this design is relatively low, and can be located with the help of figure 4.4. Starting with the male connector, cut copper conductive rings (BOM 8) with a band saw or small high speed rotary tool from the specified 1 inch OD stock. The power leads are soldered to the inside face of the copper contact rings. This will require liberal application of flux and a micro torch, as well as fixturing to hold the components during heating and cooling of the parts. The leads are laces through channels in the plastic sleeves (BOM 10,12) which separate the copper rings, and then through holes in the cylindrical plug (BOM 7). Do not neglect to include the alignment spacer (BOM 11). A threaded cap (BOM 9) holds the conductor stack in place. Cut the rubber tube to length with a sharp knife or large tubing cutters. Conductors are led through the flexible rubber tube (BOM 13) and the adapter (BOM 14) to the housing which mates to the UGV bumper (BOM 15). The rubber tube is pressed on to barbed features at each end, and secured with hose clamps (BOM 16).

Components of the female connector all interface with a core cylinder (BOM 2), which provides a rigid structure demanded by the tolerances in the conductor and latching elements. Insert and rotate the funnel (BOM 1) in the locking feature at the face of the cylinder. Chase the M8 threaded holes along the wall with a tap, then thread in the ball plungers (BOM 4). These electrical contact components should be adjusted to compress when the connection is made, without their housing protruding into the insertion path. Add the ring terminals (BOM 5) on the protruding threaded casing of the ball plungers, and lock the contact assembly in place with the nuts (BOM

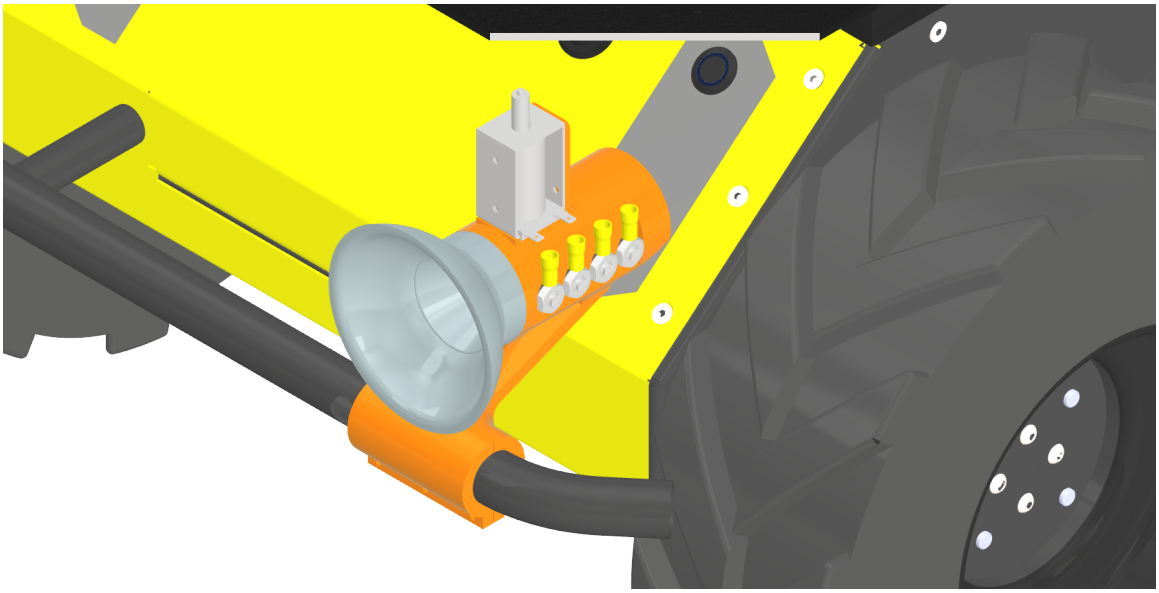


Figure 4.5: Optional female mounting configurations include a bumper mount for the Clearpath Husky UGV.

6). If the application demands a latching connection, a solenoid (BOM 3) can be mounted to the top of the housing. The pin of the solenoid engages a feature on one of the spacers (BOM 12) between the contacts of the male connector. Bolt the female connector to the infrastructure it serves using the 0.25 inch diameter holes provided.

An optional part, which is included in the design files with this paper, is a bumper mount, figure 4.5, for the female connector half (BOM 17). This mount is designed for the Clearpath Husky UGV, and is used by the authors to create a shared power bus between heterogenous power sources in a mobile microgrid. The mount includes a printable hinge feature, and is drilled for M2.5 fasteners.

Another configuration for this connector system equips the male connector with a single degree of freedom actuation system, figure 4.6. This improves the control

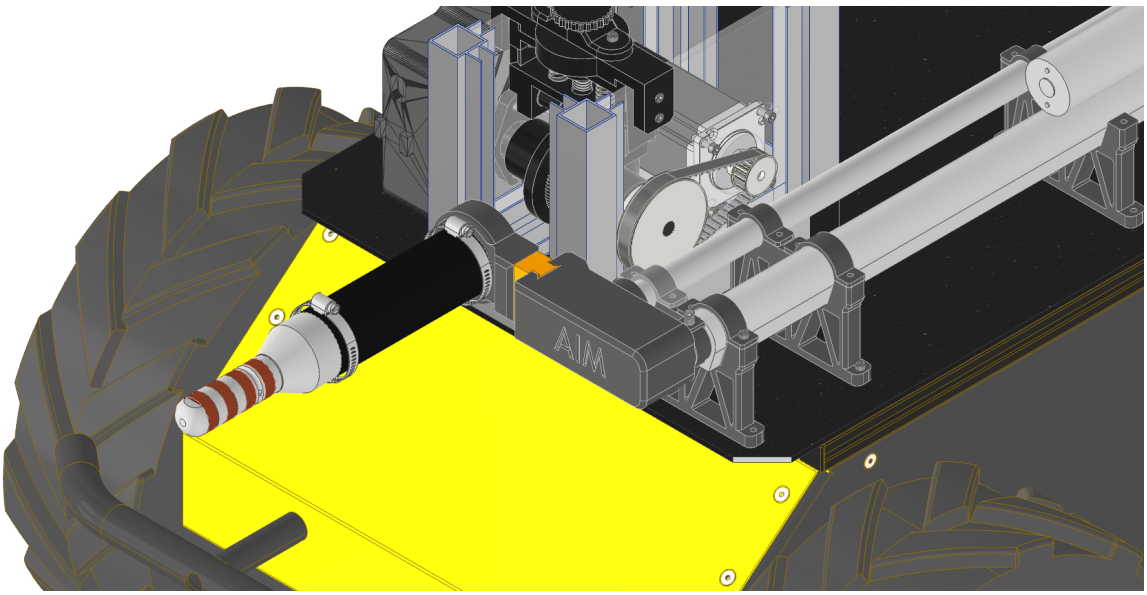


Figure 4.6: Depending on mobility and application, the male connector can be mounted to an actuator, in this case a single degree of freedom in the axial direction.

of insertion depth alignment compared to direct mount to the UGV chassis, while maintaining the benefits of the axial symmetry of the connector and the flexible component that mitigates lateral and axial misalignment. It also retracts the male connector inside the chassis when not in use, which simplifies the autonomous mobility planning for the vehicle.

One obvious modification to the design could be a reduced two-conductor implementation for a DC system. This would be valuable for agent recharging, supporting persistent operation such as surveillance or industrial robotics. Other potential applications could be a downward facing connector element on a Unmanned Aerial Vehicle (UAV) with the mating connector integrated into a landing pad for recharging.

4.6 Operating Instructions

Implementation of this connector system will be individualized to the vehicle and power application that adopts it. Regardless, there are some operation instructions that should be applied to all use cases. First, ensure that all the 3D printed components are free from flash or residual support material. The mating tolerances between connector halves are reasonable for the application of the 3D printing technique to fabrication of these components, but docking and departure will be hampered by undue friction or binding resulting from poor finish quality. Maintain a light silicon/PTFE lubrication of the termination of the funnel component ID (BOM 1,2).

Integration of the connector components with the AC/DC electrical systems is outside the scope of this paper, but some attention to the bus system is merited. Ensure that a Normally Open (N.O.) relay is wired in series between each connector contact and the associated bus. These N.O. relays should be controlled by the parent device (UGV or infrastructure), and should only be energized when the UGV signals that the docking approach is successfully completed and the vehicle is in an immobile state. In the autonomous mobile microgrid system, the UGV docking process is governed by a visual feedback controller, described in [40], that measures the insertion depth of the connector probe and signals successful connection coupling over a wireless network. A continuity check could also be incorporated to check the connection state. These

relays are the primary engineering control for electrical hazard mitigation. Conditions that should be mitigated are connection or disconnection while the contacts are energized, energizing of a partially inserted connector, and energizing of an un-mated connector with exposed contacts.

Finally, the integration of this system may constitute a substantial modification in the footprint of the parent vehicle. Update the vehicle's mobility control to account for the addition of these components and to avoid collision of the connector with obstacles.

† Engineering controls must be placed on the autonomous system to prevent connection/disconnection of the system while the connector is energized.

† Engineering controls must be in place to prevent energizing of the male connector contacts while the male connector is exposed.

† Mobility planning should take into account the dimensions of the connector to prevent collisions with people or objects in the operational environment.

4.7 Validation and Characterization

Beyond the use of this AC/DC power connector in mobile microgrid demonstrations, a test regimen has been devised. This test consists of automated insertion

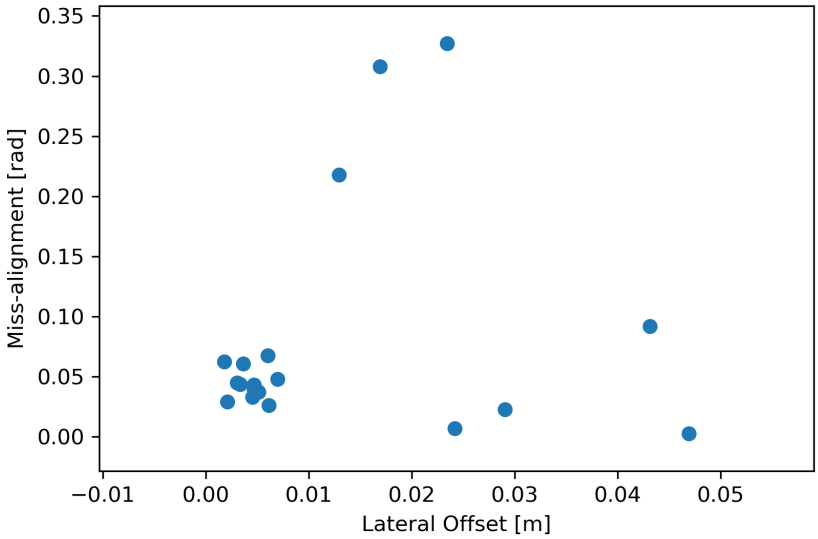


Figure 4.7: UGV positions for connection insertions across an array of lateral and axial miss-alignment.

attempts using a visual docking method, augmented by manual perturbation of the insertion miss-alignments. The tests were performed outdoors, but on a prepared surface in this case, with the vehicle reset by the operator between approaches. The lateral and axial miss-alignment during automated visual docking insertion covers a range of 0.0018 m to 0.0070 m and 0.026 rad to 0.063 rad, grouped in the lower left corner of Figure 4.7. This range is substantially better than that allowed by the connector design requirements, so additional manual perturbation during connector insertion was included to better explore the connection miss-alignment region. For a more generalized representation of the use of this hardware, review the linked video https://www.youtube.com/watch?v=zMo8FnATznk.

4.8 Conclusion

The AC/DC power connector described in this work provides a required component in a versatile robot-based solution for wired power transmission. This devices has the following attributes:

† Purpose-designed to accommodate the docking behavior of an autonomous UGV.

† Mitigates docking alignment error up to +/-2cm in any direction and +/-15degrees.

† Supports electrical power transmission requirements for an AC/DC mobile microgrid.

† Does not require an expensive multi-degree of freedom manipulator for connection coupling.

Beyond the application of wired power for autonomous mobile microgrids, consider the application of this flexible design for other high-wattage tasks, such as recharging of persistent-mission ground or aerial vehicles.

Chapter 5

Visual Cable Tracking and Deployment for Autonomous Mobile Microgrid Power Transmission[1]

5.1 Motivation

Mobile microgrids combine the functionality of independent power grids with multi-domain, multi-purpose, autonomous Unmanned Ground Vehicles(UGV). Applications with human risk or human denied environments are especially valuable, which

[1]The material contained in this chapter is in preparation for submission to a journal.

range from military forward operating bases or disaster response, to planetary infrastructure. Micro power grids incorporate AC/DC conversion, load balancing or load shedding, power transmission, energy generation with grid storage, and grid optimization or control. Additionally, mobile microgrids must optimize for both autonomous vehicle constraints and the power grid requirements. Power grid hardware is deployed on the mobile robots, which configure according to mission requirements and any fixed load or power source infrastructure, Figure 5.1. If conditions change, the mobile grid elements should be able to optimally reconfigure. This work demonstrates a method of power cable connection of microgrid sub-systems using autonomous UGVs. To support the presentation of this development, the Motivation section of this paper will outline the applications and characteristics of an autonomous mobile microgrid, and discuss some challenges inherent to power connection coupling and power cable deployment by mobile robots, including related or prior work.

5.1.1 Introduction to Mobile Microgrids

Mobile microgrids provide greatest utility when traditional power grids are unavailable. During natural disasters, damage to un-hardened distribution and control infrastructure, such as above-ground power lines and substations, is common [12]. Literature promotes microgrids as one solution to recovering from an event which disables

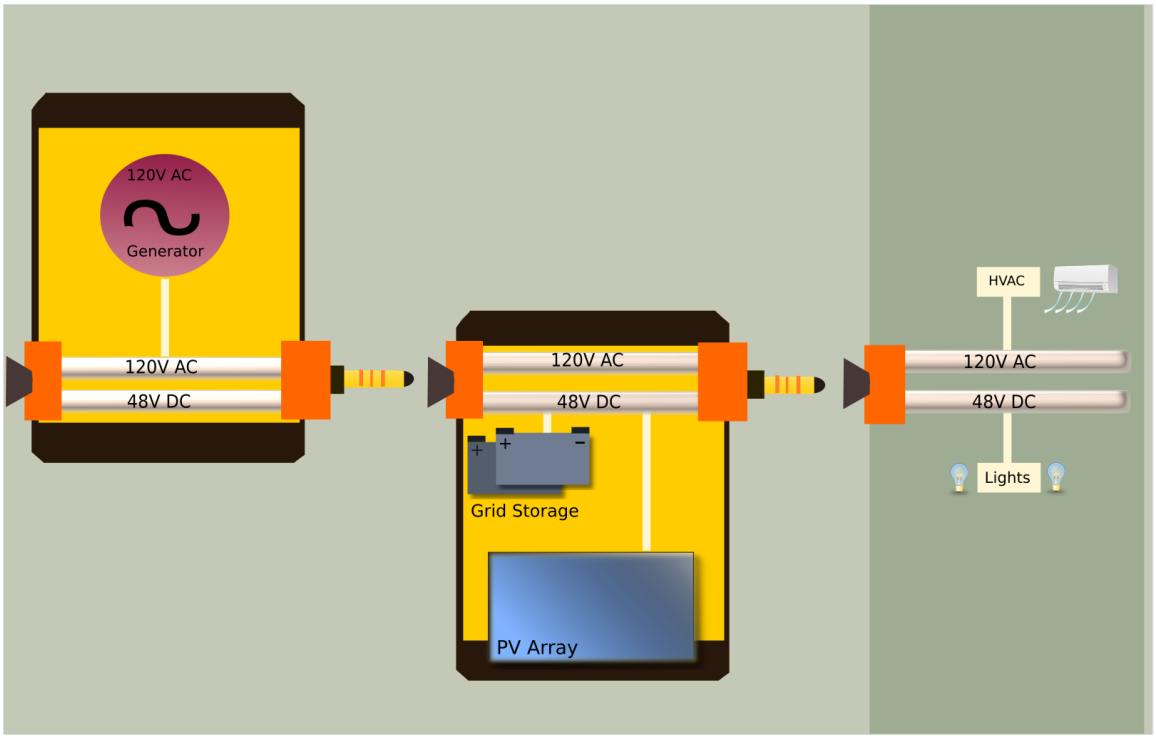


Figure 5.1: Mobile microgrids include power source, bus, and power transmission hardware that is integrated with mobile robots for operation in human-denied environments

a regional powergrid [4], with much work on utilization of existing distributed resources, such as home PV arrays [5] and consumer electric vehicles [6] as supply and energy storage nodes. To our knowledge, no work has provided an autonomous robotic capability for replacement of critical power grid infrastructure following a disaster. Robotic agents for power grid recovery should be prepared to form mobile microgrids. Agents need to fulfill the essential functions of a microgrid: power generation, energy storage, transmission, and power control [13]. Other related work [59] considers the control of different types of energy sources, in conjunction with energy storage, on marine vessels as a type of islanded microgrid.

Very little work is available in the literature regarding the deployment of power grids on mobile robots. The literature focuses on allocation and control of power sources [8] [14], as well as design of mobile power sources [7]. A modular mission payload system was designed, including an integral power source [15]. One work [9] allocated truck-mounted mobile-energy-resources to restore islanded loads after a disaster. A similar work [10] seeks to optimize the pre-positioning of mobile energy-sources, then re-allocate them as needed following a disaster. Although some work explores the power system topology [11], there is little handling of real-world constraints or costs for mobile power assets. Mobile microgrids should be readily deployable, decentrally staged awaiting deployment, and capable of operating in the degraded infrastructure encountered following a disaster. [16, 17, 18, 19, 20, 21, 40, 53, 54].

5.1.2 Power Connection Coupling and Cabling

Investigation of methods for navigation, docking, and electrical connection coupling and cabling using these mobile microgrid UGV agents has been informed by the following work in the field that address similar tasks. Recent work demonstrated moon-analog rover autonomy for navigation and manipulation [32]. A mobile rover manipulator has been developed for planetary missions [42]. Substantial work addresses robot docking, for different applications throughout many domains [22, 23, 24]. For the purpose of mobile microgrids, docking supports a wireless power transfer task,

as well as wired electrical connection coupling. These tasks depend on mm-scale docking precision, with a practical maximum of any misalignment angle $< 10^{\circ}$ [53]. Docking and mobility analysis was performed for a tethered planetary rover [43]. Visual tether points are compared to a catenary tether model using a non-linear least squares fitting procedure[33], with a 2-D control law presented for the follower robot to maintain tension in the tether. A particle filter is applied to 10000 manually annotated training frames depicting an under-sea cable to the steering controller of a survey Autonomous Underwater Vehicle (AUV) [34].

Tether management and control is often addressed in the aerial domain, but more seldom in a UGV application. One work [35] presents a concept for large-scale concrete printing robots, with a collection of logical heuristics for motion planning based on graphs of occupancy regions, which prevent entanglement of self-with-own tether or self-with-other UGV or tether. More applicable to the present application, one work [36] tethers two dockable halves of a rover, using a taut-tether requirement and and visual feedback from fiducials as inputs for motion planning. An additional work [37] documents the tether control mechanism and feedback sensors used in the above rover. Finally, very interesting work[38] deploys a taut tether from a UGV for mobility assistance in steep terrain, and [39] further demonstrates a-priori planning of the tether anchor points for comprehensive mapping of a rugged terrain region.

5.1.3 Statement of Contribution

There are a few distinctive requirements for electrical power cabling deployment, compared to the above related work. The electric cable, while robust, should not be considered a load-bearing tether. It should be deployed in a manner such that minimizes tension on the cable, and should not be pulled or dragged along the ground by the UGV. Entanglement with obstacles should be avoided, by laying the cable in the obstacle-free path traversed by the UGV. The electrical cable and electrical connection system should operate autonomously. This work contributes a UGV-based method for measuring and controlling the deployment of a cable which lies on the ground surface outdoors without tension, and free from entanglement with obstacles.

5.1.4 Contents

The contents of this paper are as follows: considering the Motivation 5.1, controllers to achieve the docking, connection, and cabling requirements are defined 5.2. The UGV and hardware configurations are provided, and a Monte-Carlo style validation method is proposed 5.3. Results depict the behavior of the controllers in use, and present validation trials 5.4. Beyond the successes demonstrated by this work, it concludes with ideas to extend and improve the method 5.5.

5.2 Controllers for Docking, Connection, and Cabling

Cable deployment requires hardware subsystems beyond other autonomous mobile microgrid UGV agents. An integrated cable storage spool and feed drive terminates at a linear actuator which deploys a floating electrical connector. Control for these systems is distributed between the ROS environment hosted on the UGV computer, and real-time control running on a microcontroller. set-points are generated in the ROS system from state machines and the navigation planner. A number of cameras provide feedback estimates to the ROS system. Cable deployment is controlled by a feed-forward/feedback method which calculates the length of cable to be extruded over the next time step. A ramp controller on the microcontroller assures that the stepper drive motor maintains speed and acceleration constraints while paying out the cable. A linear actuator is used to deploy the floating electrical connector which occupies the free end of the cable. Position control on this actuator is provided by a feedback controller running on the microcontroller, with extension lengths generated by the vehicle state machine and visual docking process. Examples of the code for these controllers can be found in Appendix B.

Table 5.1
move_base parameters associated with the cable deployment vehicle reported in this work.

Parameter	**Value**
Controller frequency	5 Hz
Recovery behaviors allowed	True
Global costmap resolution	0.1 m
Global costmap size	1600 m^2
Maximum range for an obstacle	5 m
Max volocity x	0.25 m/s
Max velocity x backwards	0.25 m/s
Max velocity theta	0.3 rad/s
Acceleration limit x	0.5 $m/s/s$
Acceleration limit theta	1.0 $rad/s/s$
Minimum turning radius	0.0 m
Footprint model type	point
Footprint model radius	0.2 m
XY goal tolerance	0.05 m
Yaw goal tolerance	0.087 rad

5.2.1 ROS move_base Implementation

ROS provides the Husky UGV with a highly modular software environment to implement control architecture from low-level hardware control to modeling and simulation [44]. ROS includes a set of packages known as the navigation stack [45]. These packages provide many of the common simultaneous localization and mapping (SLAM) algorithms which autonomous robots depend on. ROS is inherently modular, and there are standard Application Programming Interfaces (APIs) for developing new methods within the stack. The navigation stack maintains costmaps that locate obstacle data, including a-priori and perceived obstacles [46]. Dijkstra's method [47]

plans an optimal path through a map populated with known obstacles. A timed elastic-band planner [48] has been adopted for smooth chassis motion near obstacles and to account for dynamic obstacles in the environment. Odometry for the robot is provided by an Extended Kalman Filter with wheel encoders, IMU, and GPS data as inputs [49].

A summary of the important configuration parameters for the move_base elements of the navigation stack as deployed for this testing is provided in table 5.1.

5.2.2 Docking via Visual Feedback

Visual docking is performed using a combination of open-source algorithms available in ROS, and set of task-specific calculations based on visual feedback from an Augmented Reality(AR) marker adjacent to the docking goal at a load. ROS platform elements can be replicated based on the configurations from parameter files. The visual steps are addressed in greater detail.

Microgrid UGV agents need to be able to achieve accurate positioning adjacent to infrastructure or other robotic agents before initiating power coupling. This docking controller needs a continuous exteroceptive source of localization data. A viable solution is to implement a visual feedback method using a calibrated [50] monocular camera and Augmented Reality (AR) tracking of fiducial markers [51]. A fixed-eye

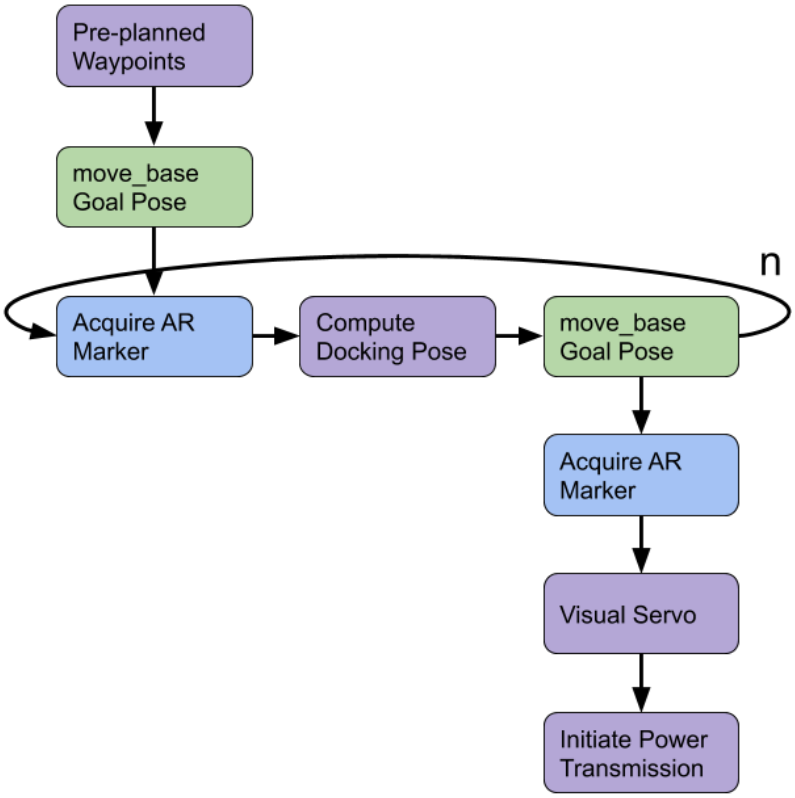


Figure 5.2: Docking control leverages ROS core navigation functionality with a visual method which introduces increasingly accurate estimates of the position of the docking goal

camera is installed on each UGV. A zero-turn pan method to acquire the augmented reality marker in the camera frame when needed during the docking maneuvers. If the pre-planned rendezvous waypoints between UGV and docking target are very poor, or there is some un-accounted for occlusion in the environment, the zero-turn pan method would fail to acquire the marker; thus it is especially important to verify the suitability of the rendezvous waypoint in the pre-planning stage of the mission.

After the rendezvous waypoint is achieved, the docking controller checks to see if the AR marker is in the camera field of view. If it cannot locate the marker, it engages

a pan controller until the marker is acquired. Once acquired, it calculates a docking pose offset from the marker and passes this pose (x,y,θ) to the ROS Navigation stack as a waypoint goal. This waypoint goal is more accurate to the true pose of the docking target than any pre-planned rendezvous waypoint. Once the waypoint controller completes, the UGV is relatively near to the AR marker, which presents the best estimate of its position. It is advantageous to recalculate the docking pose offset and generate an updated waypoint goal, Figure 5.2. Empirically, $n = 2$ repetitions is sufficient.

Finally, the agent visual servos to the final docking pose. The pose error, e_x and e_y, between the robot chassis pose and the docking pose in a $2-$D cartesian frame is calculated using the AR marker frame in ROS tf. A proportional feedback loop is engaged, where $\dot{x}$ and $\dot{\theta}$, the linear and angular velocity commands, are calculated

$$\dot{x} = P_x \; e_x \tag{5.1}$$

and

$$\dot{\theta} = P_\theta \; e_y \tag{5.2}$$

where P_x and P_θ were tuned empirically. This behavior executes until e_x drops below a threshold distance. At this time, the docking method exits and signals the mission controller to proceed to the next behavior.

5.2.3 Electrical Connection Coupling

Electrical connection coupling for autonomous robots is an open problem [28]. One approach is to develop a control scheme for a general-purpose multi-degree-of-freedom robot arm, at great cost for the manipulator hardware itself, and with great difficulty in the grasping and force-feedback manipulation of the connector itself. A more practical approach is to design an electrical connector system to the mobility constraints of the UGV vehicle in its anticipated operating environment. This work has been undertaken [60], resulting in a probe-and-funnel design which mitigates the lateral and axial error of the UGV docking approach, Figure 5.3, and supports AC and DC power transmission at wattage suitable for the needs of this project.

5.2.3.1 Floating Connection

The floating connector has two insertion actions required for its deployment. It must be retained on the UGV until needed, inserted into the target connection and locked, then released from the UGV. To recover the connector, the UGV-end retention mechanism is inserted onto the cabled end of the floating connector, before it is unlocked from the electrical connection and retrieved. To facilitate these steps, the floating connector is deployed by a linear actuator, rather than direct-mounted to the

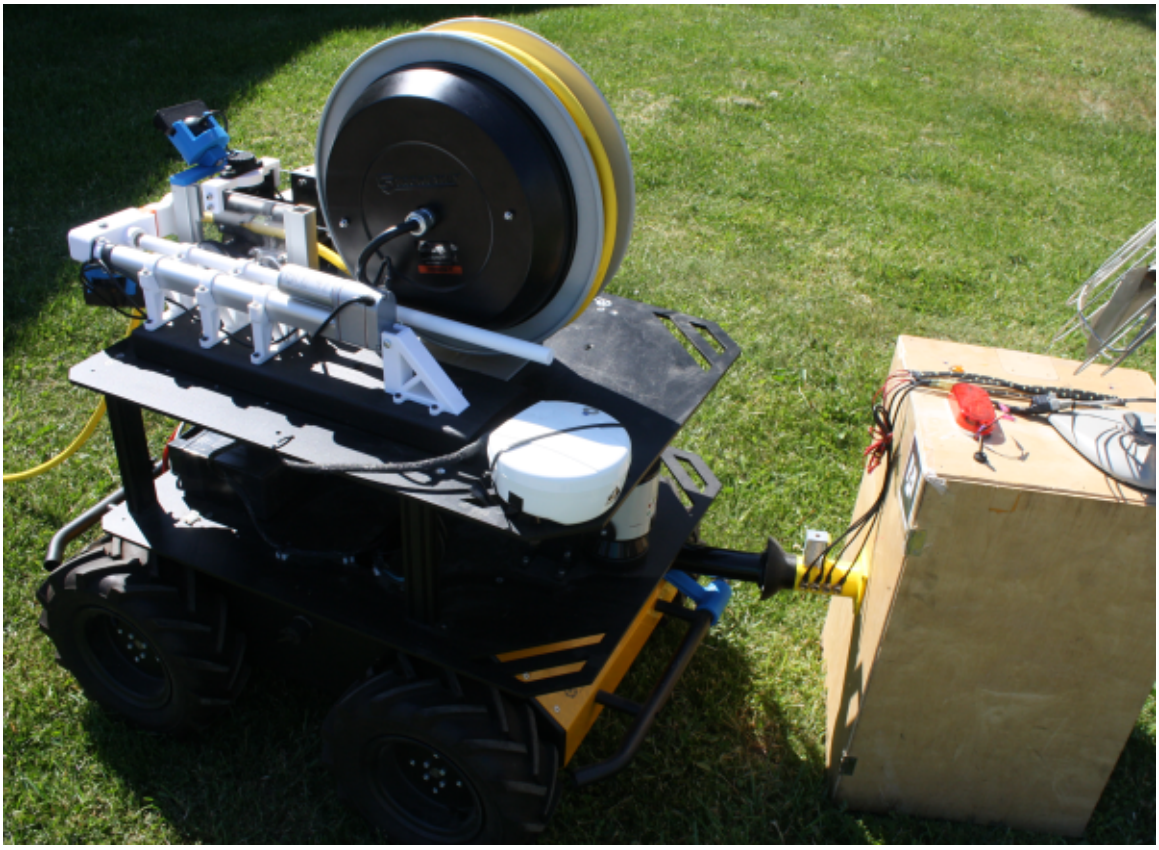


Figure 5.3: ACDC electrical connector for UGV coupling.

UGV chassis. This linear actuator is driven by a PWM signal to a dual-H-bridge, with a potentiometer providing position feedback. Position control for the floating connector depends on the state machine running on the UGV control computer, including visual measurement of the target insertion depth using the AR marker stack used in docking, as well as a low-level PI controller running on an ATMEL 328p μcontroller for set-point tracking. Figure 5.4 depicts the relationship between the high-level visual position controller, and the low-level set-point tracking controller.

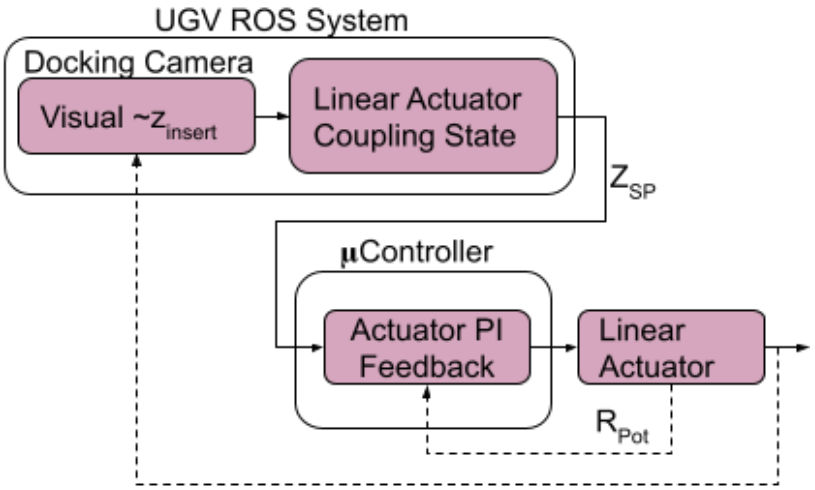


Figure 5.4: Insertion of the floating connector is performed by a linear actuator, with visual feedback for extension set-point provided by the UGV, and PI feedback to achieve that set-point performed by a μcontroller.

5.2.3.2 Fixed Connection

One of the connectors on an Autonomous Mobile Microgrid cabling UGV is mounted directly to the the UGV chassis. This fixed connector is coupled after the UGV has traversed from the origin point of the cabling connection, to the terminal point of the cable. Connection coupling of the fixed connector can be achieved using a modified visual docking controller, discussed in section 5.2.2. After the docking controller reports a successful connection approach, relays are activated to enable power transmission through the connector element.

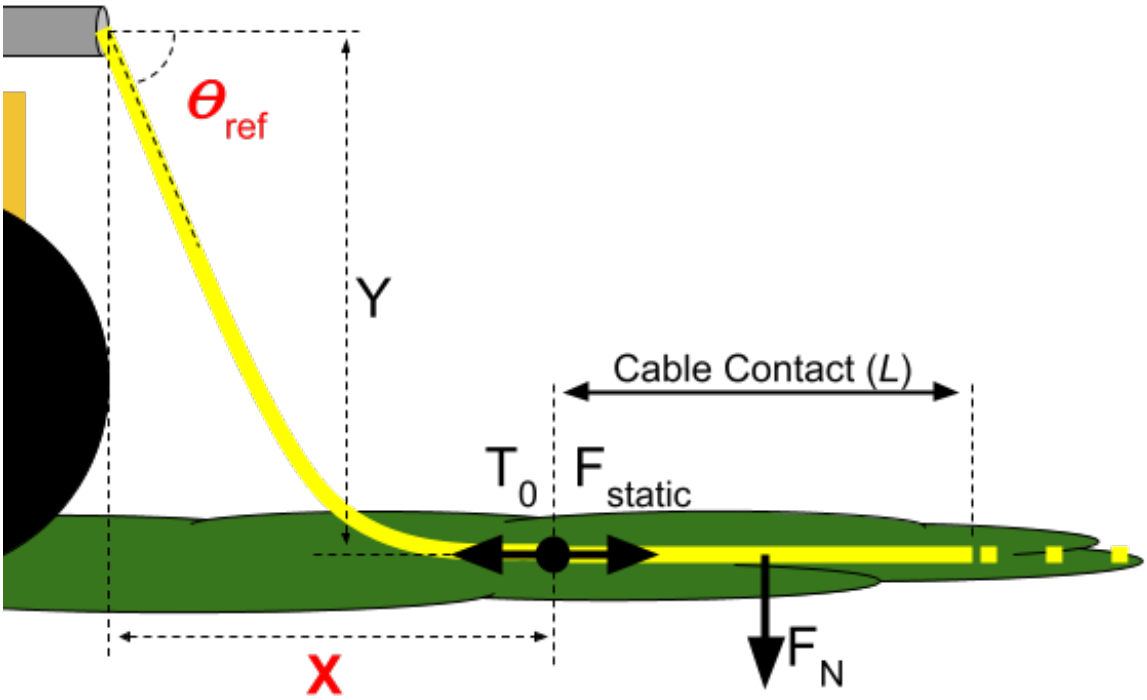


Figure 5.5: Cable deployment is based on a simplified model of a catenary-shaped suspended cable in equilibrium with the cable-ground friction interaction.

5.2.4 Cable Physical Model

Before undertaking a control system design for deployment of the electrical cable, some consideration of the assumptions made for the physical model of the cable were necessary. Figure 5.5 depicts the behavior of a cable departing from a slow-moving UGV, in some small region where the cable is considered actively manipulated by

the vehicle, rather than laying slack on the ground. From a fairlead above the rear wheels, the cable exits the vehicle and hangs in a catenary shape, with the lowest point of the curve in contact with the ground. The remainder of the cable lays on the ground without dragging. For the purposes of this design, behavior of the cable due to its weight were considered to dominate over effects of cable stiffness.

Consider some electrical cable with a weight-per-unit-length $\mu_c = gdm_{cable}$, where g is acceleration due to gravity, and dm_{cable} is the unit mass of the cable. At the lowest point of a single-sided catenary formed by the cable freely hanging from fairlead to ground surface, the tension in a static cable, T_0, must equal a frictional force, F_{static}, provided by the ground-cable interaction. Consider equation 5.3, where this static friction force depends on some length, L, of cable which is considered actively providing this ground contact, as well as μ_c, and a representative static coefficient of friction μ_{static}.

$$F_{static} = \mu_{static}\mu_c L \tag{5.3}$$

By choosing limiting values for L and μ_{static}, a maximum tension force, T_0, before dragging the cable across the ground is estimated. It can be shown that for a given fairlead height above the ground, Y, the relationship in equation 5.4 can be numerically solved for the distance, X, of the point of ground contact as a function of this

force T_0. For purposes of control, this X is used to calculate the minimum angle from the horizontal, θ_{ref} , that the cable may be permitted to extend from the fairlead, by equation 5.5.

$$Y = \frac{T_0}{\mu_c}(cosh(\frac{\mu_c x}{T_0}) - 1) \tag{5.4}$$

$$\theta_{ref} = tan^{-1}(sinh(\frac{\mu_c x}{T_0})) \tag{5.5}$$

This relationship can be solved for a variety of cable-ground static friction coefficients, μ_{static}, resulting in a range of possible cable droop angles, shown in Figure 5.6. For 0.3 to $5.3N$ holding forces, θ_{ref} is anticipated to fall between 88 through 64 degrees, respectively. This gives ample envelope to choose a set-point for a cable deployment controller which satisfies the slack-cable requirement.

5.2.5 Visual Cable Tracking

Measurement of the cable deployment rate has been shown to be difficult by mechanical means. Furthermore, open-loop deployment of the cable based on the UGV's planned traversal distance will inevitably accrue error over time. A camera located

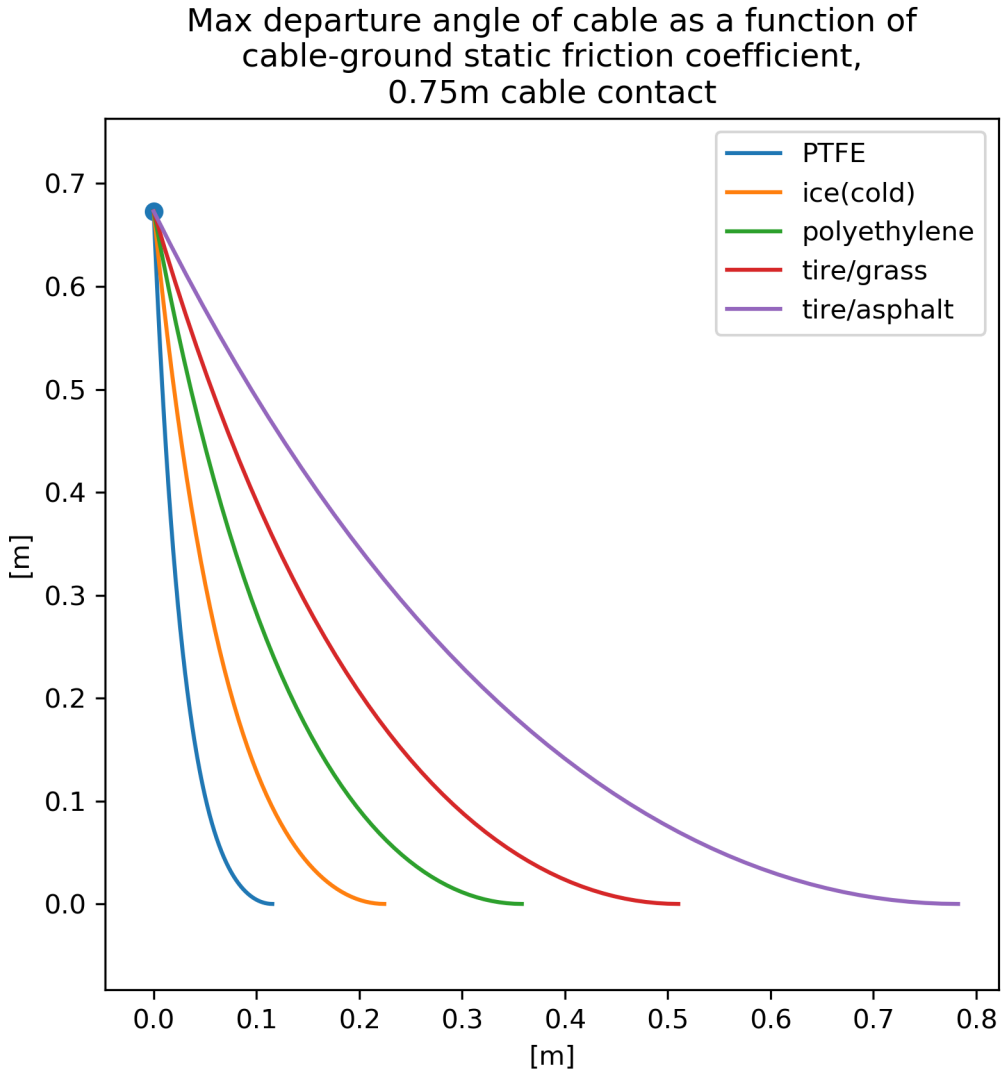


Figure 5.6: Cable droop for a range of cable-ground static friction coefficients.

adjacent to the cable fairlead allows for visual estimation of θ_{ref} and provides the feedback signal to the deployment controller. This visual estimation consists of a multi-step convolution of image filters using OpenCV. Of note is an adaptive parameter estimation of the pixels representing the cable in the Hue, Saturation, and Value (HSV) image space, using a Region-of-Interest (ROI) in the camera frame, Figure 5.7.

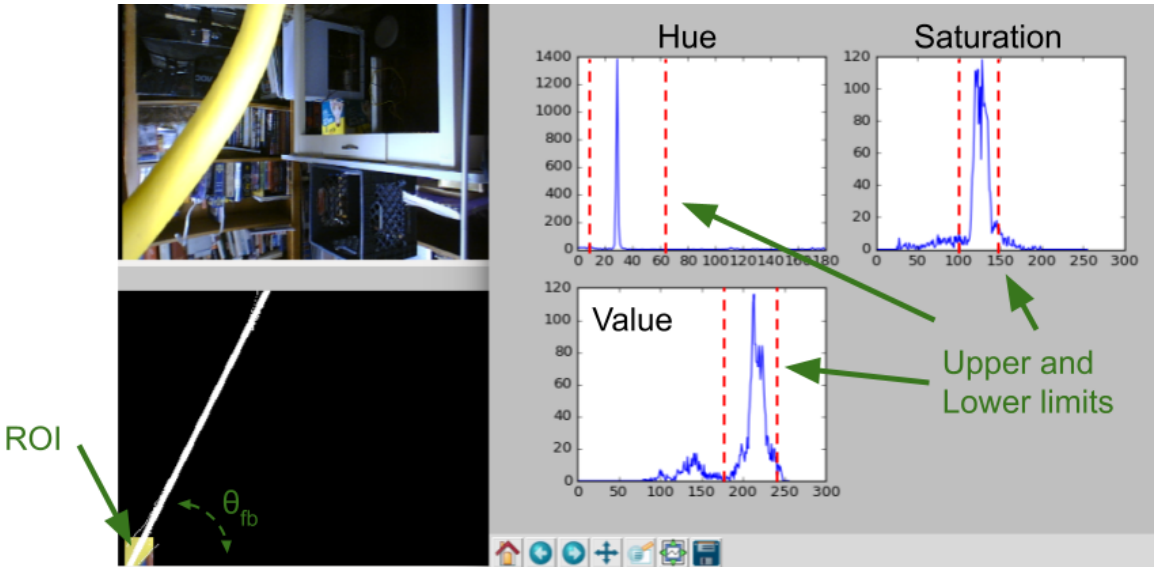


Figure 5.7: Cable angle is fit to a mask, updated using HSV mean and standard deviation from ROI pixels.

As lighting conditions change, the HSV mean and standard deviation of the cable is estimated, and limits are calculated to generate an image mask. Further filters reduce this mask to a linear fit of the feedback angle, θ_{fb}, Figure 5.8.

New camera frames are passed from the ROS image topic callback at $5hz$ into an OpenCV processing stack. First, the images are converted to a Hue Saturation and Value (HSV) colorspace. Next, the mean and standard deviation of the cable in the HSV space for this image frame is calculated from the pixels in a small Region-of-Interest (ROI). This ROI is defined by a bounding box around the pixels where the cable exits the fairlead, thus the cable is always present in the ROI. The mean and standard deviation values are used to calculate an upper and lower threshold for each of the HSV channels as a mask for a binary image representing Cable/Not-cable. Because the HSV channel thresholds are updated for every image frame, the

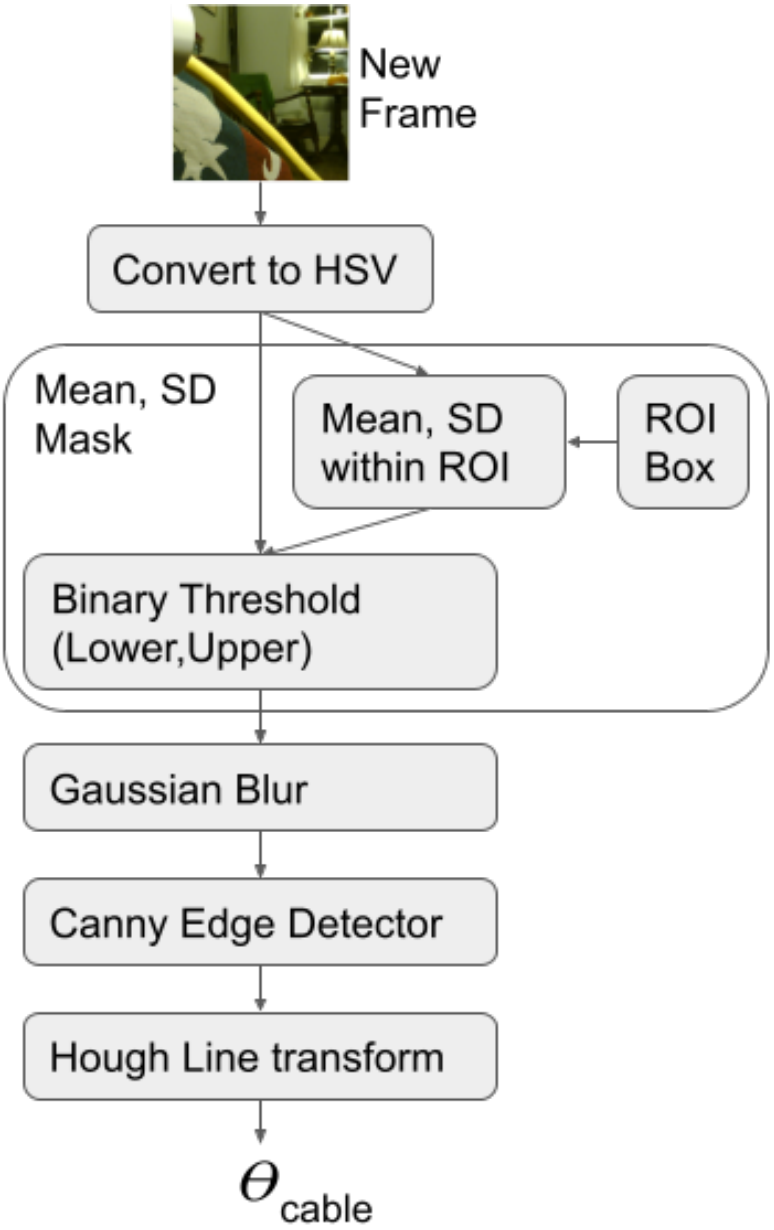


Figure 5.8: New camera frames are reduced to a cable vertical angle estimation by estimating the HSV parameter range of the cable from a ROI and performing a Hough Line transformation on the masked pixels.

cable segmentation is robust to the changes in lighting conditions which constantly occur in an outdoor environment. The binary image undergoes a blur and then is run through a Canny Edge Detector to filter out spurious false regions that may have been

identified as Cable in the HSV mask step. While the cable regions have linearity, the false positives from the background of the image that may satisfy the HSV thresholds generally do not have much linearity and are eliminated by the Cabby Edge filtering. Finally, the binary image is composed of a distribution of non-zero pixels along the path of the cable. A Hough Line transform [61] is performed on the binary image, and the slope of the line with the greatest accumulator votes is retained as the cable angle in the image. This angle is an approximation of θ_{cable}, required for the visual feedback controller.

In similar fashion, an azimuth estimate is provided, leveraging the rear docking camera to track the cable in the horizontal plane. The vision processing stack used for the angle estimate is applied for the azimuth estimate, with the important distinction that the HSV mean and standard deviation values calculated for the cable ROI in the side camera are utilized by the rear camera. The side camera HSV values are modified to slightly enlarge the permissible range of Saturation, and significantly expand the upper limit on Value. Azimuth estimates are provided at a slower rate of 1 hz, partially to reduce computational burden, and also because the azimuth values are only used to toggle handling of the chassis angular velocity by the feed-forward controller, when available.

5.2.6 Cable Deployment Controller

Deployment of the power cable connecting two grid elements must avoid entanglement with the UGV as it traverses the operating environment, and avoid entanglement with obstacles. One constraint that satisfies these requirements is that the cable should lay without tension in the track of the UGV, touching the ground some distance from the rear of the vehicle. This behavior can be approximated by a physical model including a catenary curve for suspended cable, and a length of cable contact which provides sufficient friction to anchor the cable, defined in section 5.2.4. Considering the slow rate of traversal acceptable for this application, a static relationship can be developed which provides an angle, θ_{ref}, which the cable will depart the fairlead, and distance, x, where the cable will first touch the ground behind the UGV, Figure 5.5. If the cable is deployed such that x is similar in magnitude to the path planning diameter of the UGV, the cable will remain in the obstacle-free path that the UGV has traversed through the environment.

To ensure that the cable is deployed according to the above constraints, regardless of disturbances either to the UGV as it traverses or to the cable deployment system, a feed-forward/feedback controller has been designed, Figure 5.9. Velocity commands to the UGV chassis are transformed into a scalar deployment speed, V_{FF}, by the feed-forward controller. The UGV velocity updates are comprised of a linear component,

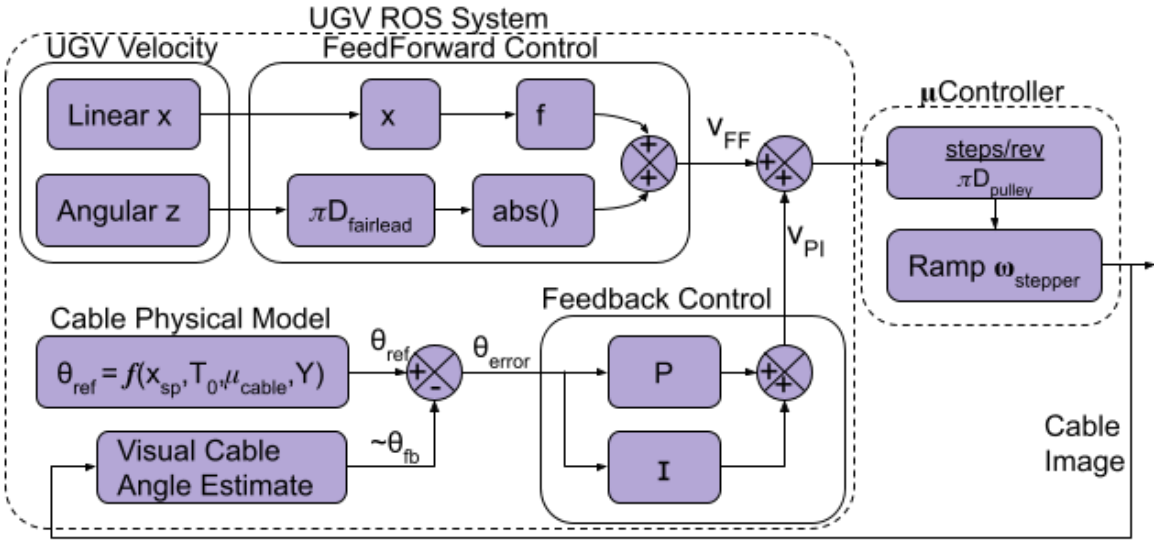


Figure 5.9: Feed-forward/Feedback control of cable deployment depends on the vehicle velocity command, with visual estimation of the cable deployment angle closing the loop via PI gains.

x, and an angular component, z. A user-tuned parameter, f, applied to the linear component helps mitigate bias in the feed-forward control due to any un-estimated disturbance within the system. The linear cable speed can be approximated as the sum of the linear component, with the absolute value of the angular component multiplied with the circumference of the arc the cable fairlead traverses over the ground as the UGV turns, $\pi D_{fairlead}$. V_{FF} is summed with a feedback speed, V_{PI}. This feedback loop calculates θ_{error} of the cable droop angle with respect to horizontal, between θ_{ref} and the visual cable angle estimation of θ_{fb}, section 5.2.5. θ_{error} is passed through a PI controller, with empirically tuned coefficients $P = 0.75$ and $I = 0.1$. Additional features of the PI controller are independent positive and negative integrator anti-windup limits, values $[-20, 10]$, and an asymmetrical deadband, set at $[0, 0.8]$. The integrator is reset whenever the error is within the deadband range.

Because of the asymmetry of the plant in regard to the effect of gravity (angles approaching horizontal have a large length of suspended cable, angles approaching vertical have a small length of suspended cable), the PI controller must deal with substantially different dynamics at the limits of the controllable range of angles. Rather than implement a gain schedule, especially for positive vs. negative error, an integrator-heavy tuning with asymmetrical anti-windup limits was able to achieve suitable control with simplified tuning. The feedback controller rejects saturated values of $0, 90°$, falling back on the feed-forward controller in those situations. This feed-forward/feedback controller is running on the UGV chassis at $15hz$, with the ROS maintaining the node timing and publishing the output of the controller.

Velocity updates from the feed-forward/feedback controller on the UGV ROS system are handed off to a microcontroller handling low-level control. First, the linear velocity update is transformed to an angular speed, taking into account the steps-per-revolution configuration of the stepper motor and stepper controller hardware configuration, and the diameter, D_{pulley}, of the feed pulley in the cable drive system. This angular speed, $\omega_{stepper}$, is passed to a linear ramp stepper motor control method, [62, 63].

Figure 5.10: Ruggedized UGV microgrid agents are equipped with a sensor suite appropriate for their operating environment, as well as power hardware. In this example, the UGV is equipped with a cable deployment system, electrical connectors, and monocular docking cameras, in addition to its standard navigational sensors.

5.3 Hardware Configuration and Validation Method

Clearpath Husky UGVs constitute the fleet of development vehicles for the work reported in this paper, one of which is shown in Figure 5.10. These Husky UGVs shipped with a Mini-ITX computer, running an Intel i5 processor at $2.9GHz$ and Ubuntu 14.04. Sensors include LORD MicroStrain 3DM-GX3-25 9-DOF IMU and NovAtel SMART6-L GPS receiver, as well as the SICK LMS-111 2D LiDAR. A

number of monocular PS3 Eye cameras are mounted on the robot, in forward and rear facing orientations, as well as a side view of the cable fairlead. Electrical connections are labeled with Alvar AR tags.

Microgrid Husky's are equipped with a range of power grid hardware. The husky used in the work has power cabling and connection coupling hardware. Other agents have grid attached storage, electrical bus, or a gas $1100W$ generator. Another feature available feature on microgrid robots is wireless power transfer [53]. For high wattage power transmission or large distances, a physical power connector and cabling system provides superior efficiency. A purpose-designed AC/DC connector supports autonomous connection coupling by the UGV microgrid robots [60]. This connector terminates a cable which is deployed by the powered cable spool, with a hardware design that specifically supports the constraints of the UGV deployment platform [41]. A generic $12v$ linear actuator with $1ft$ extension range and a potentiometer feedback assists the coupling of the floating connector attached to the cable, driven by a BTS7960 type motor driver. The other power connector is mounted to the front bumper of the UGV.

For demonstration of the controllers defined in this work, the remainder of the microgrid hardware includes a stationary load and a pre-positioned AC genset. Demonstration was performed in a generally open, grassy test region. The region is portrayed in Figure 5.11. The vehicles EKF filter was utilized to locate the load at the $0m, 0m$

Figure 5.11: The outdoor test environment was generally open and covered in grass.

origin point associated with a UTM datum point in the test region. The load then remained in that location for the duration of the Monte-Carlo trials. In simular fashion, the location of the pre-positioned genset was calibrated in the mission planner using the vehicles EKF filter.

One practice for assessing robustness is to introduce a stochastic disturbance into a system over many trials and observe the system response [64]. This Monte-Carlo

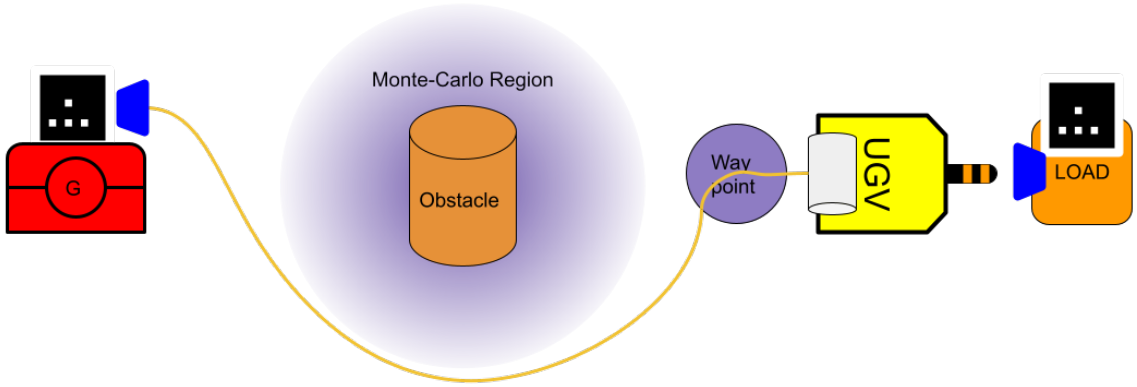


Figure 5.12: The location of an obstacle in the cable path is permuted though a region laying in the UGV path.

type method can be leveraged to validate the autonomous mobile microgrid docking controller. Ideal mission configurations may not expose failure modes and edge cases, so a Monte-Carlo test regime should be designed to explore the limits of the capability of the controller. Another benefit of the Monte-Carlo regime is that it can deliver stochastic performance measures that are important for mission planning [54]. A test routine was developed that introduced the Monte-Carlo uncertainty by permuting the location of an obstacle in the cabling environment, Figure 5.12. The cabling UGV departs from the genset connection, detects and avoids the obstacle while deploying cable, then approaches the rendezvous waypoint for the load and docks with the electrical connector. The obstacle location is permuted within a subset region of the test environment and the cable deployment is repeated.

5.4 Results

Results of this work are represented in three ways. To begin, a sampling of cable angle estimate frames are presented to show its behavior in various conditions. Next, the behavior of the feed-forward/feedback controller for cable deployment is depicted in response to a step change in vehicle velocity. The behavior of the cabling controller can be evaluated across three sets of conditions, with some important differences between them. Finally, trajectories for cabling demonstration trials are reported.

Segmentation of the cable in camera image frames must be robust to changes in lighting and background scenery. Figure 5.13 supports the capability of the cabling segmentation stack by providing examples of variation in the visual environment. From upper-left proceeding clockwise, the conditions depicted are: partial shade, camera aimed directly at sun, full sun with structured objects in background, camera in full sun with shaded cable, full sun, and shaded with unstructured background. These frames were captured with manual orientation of the chassis to ensure that conditions beyond the scope of the demonstration environment were included in the sample. Some limitations of the method can be observed. In the absence of sufficient illumination, the method will fail to have a pronounced peak in ROI HSV values, compared to the remainder of the pixels in the frame. Obviously, if the background of the image contains content which is similar in Hue to the yellow cable, especially

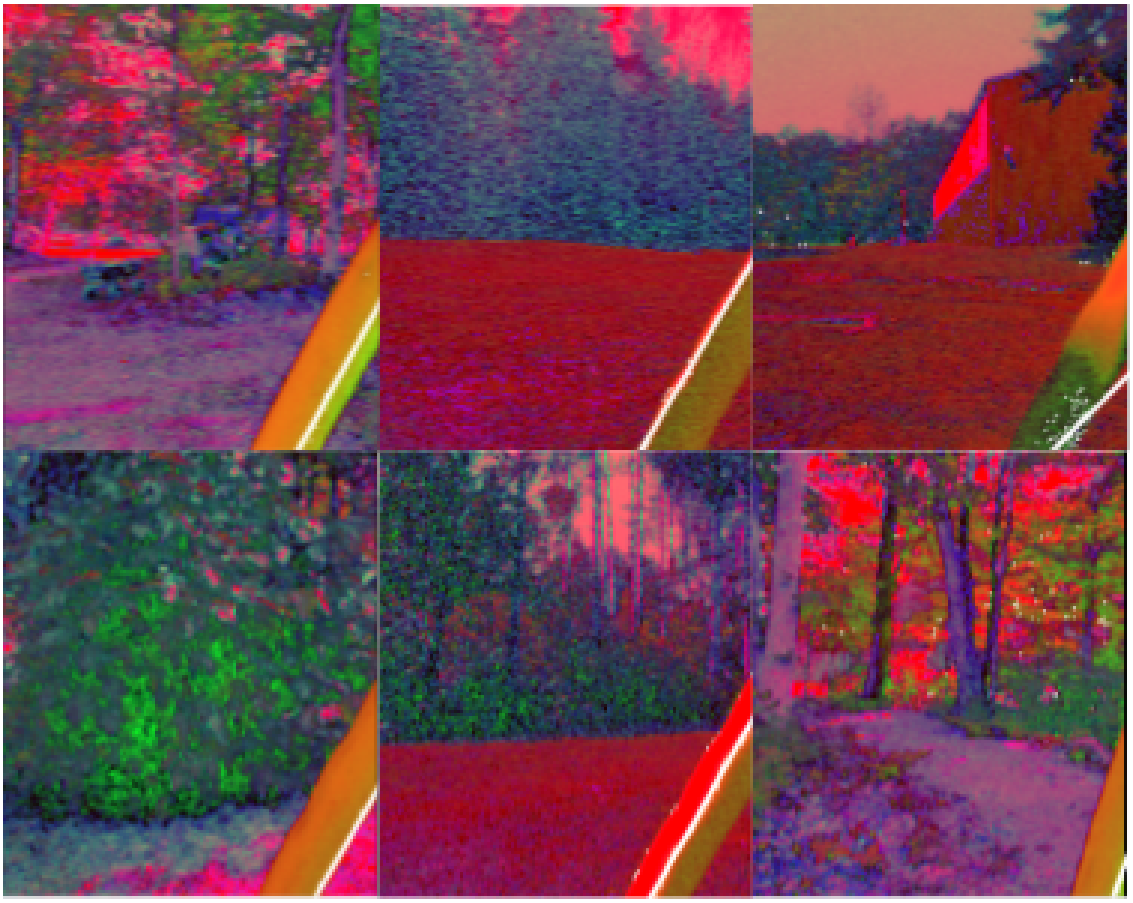


Figure 5.13: Cable segmentation samples include variation in lightning and background content.

if the content is linear or wide-spread, the method will fail.

Observe the behavior of the cable controller as it deploys cable while the vehicle is generally travelling in a forward direction in Figure 5.14. The cable feed-forward control signal is a red dashed line, and a good proxy for the UGV chassis velocity control. Feed-forward and feedback signals are summed in a solid red line, showing the total cable control action in response to both the UGV chassis commands and the visual cable angle estimate. Remember that the cable set-point angle is measured

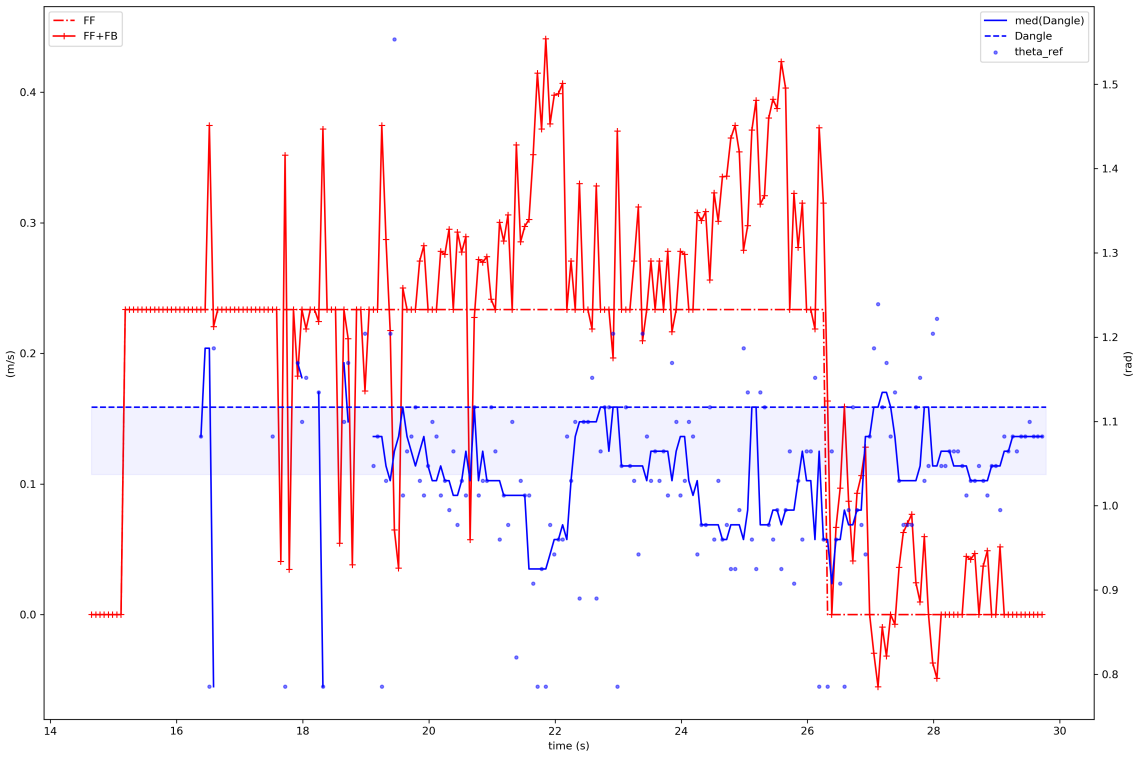


Figure 5.14: Feed-forward/Feedback control for 0 velocity steady-state, followed by a step input of 0.195 m/s in the forward direction.

down from horizontal at the cable fairlead, shown as blue dots. For clarity of interpretation, the angle estimate is also plotted as a median filter of the raw data, with a 7 sample window. For this data, the set-point was 64°, indicated by a dashed line. Deadband for the angle measurement is °, biased toward the horizontal (away from the UGV chassis), shown as a blue region. Of course, the choice of set-point angle is, in practice, a trade-off between cable laying on ground as close to the vehicle as possible, verses the cable becoming entangled with the chassis.

The bias in the deadband allows for noise mitigation in the visual angle signal, while

minimizing the risk of entanglement with the UGV chassis. Noise in the angle measurement is subject to the ambient conditions. This data is relatively noisy, as the data was acquired in low-light conditions, which reduce the range between the cable pixel characteristics and the background regions. This affects the control response by increasing the noise in the feedback component of the control signal, and by kicking the angle measurement out of the deadband region. Depending on tuning parameters, this type of noise could create an unstable controller for gains that would otherwise be suitable. If the visual feedback is highly compromised, it may stop reporting angle updates, or report clipped values. The cable controller is designed to fall back on the feed-forward signal alone in these conditions, which unfortunately constitute the early section of this data. Up to around $18sec$ the cable control signal is dominated by the feed-forward component due to missing or erratic angle updates. Nevertheless, the feedback controller is able to begin to mitigate slow deployment of the cable relative to actual chassis progress around $20sec$ into the sample. In fact, the feed-forward controller had maintained a suitable deployment speed, and from this point through the end of the forward motion, the feedback controller maintains the cable within approximately 10° of the set-point. Once the vehicle stops moving, there is slight overshoot as the integrator in the feedback controller acts to drive the cable angle towards the set-point.

Second, a reverse chassis direction adds additional challenge for the cabling controller. Specifically, the distance between the cable and the rear of the UGV chassis is very

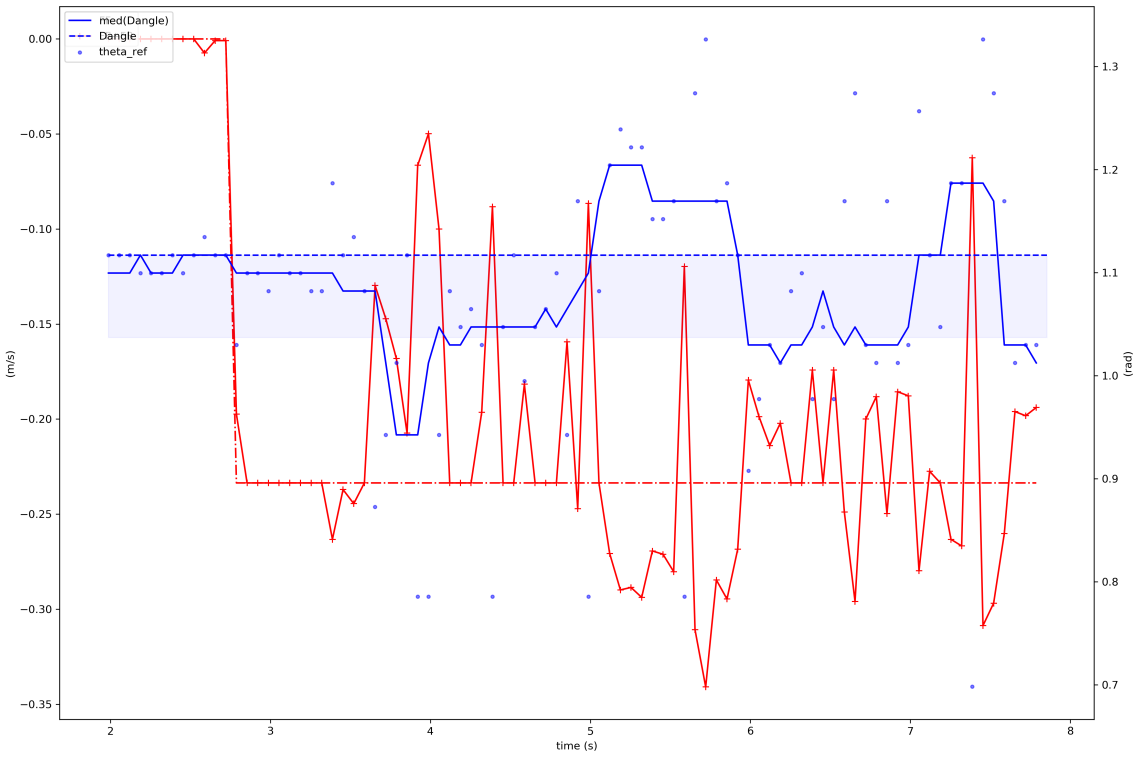


Figure 5.15: Feed-forward/Feedback control for 0 velocity steady-state, followed by a step input of 0.195 m/s in the reverse direction.

small. If the cable angle deviates from the set-point toward the vertical direction, there will be a tendency for the chassis to run the cable over. If the angle feedback signal increases beyond the set-point angle, the feedback controller must be quick to increase the cable retraction speed, such as around $5sec$ in Figure 5.15. The feed-forward controller has a user-tuned parameter to ensure that the retract rate favors over-retraction, which is manifest in the data as the angle feedback hugging the horizontal edge of the deadband.

Cable deployment through turns provides a third perspective on the challenges for the cabling controller. The feed-forward controller either feeds or retracts cable based

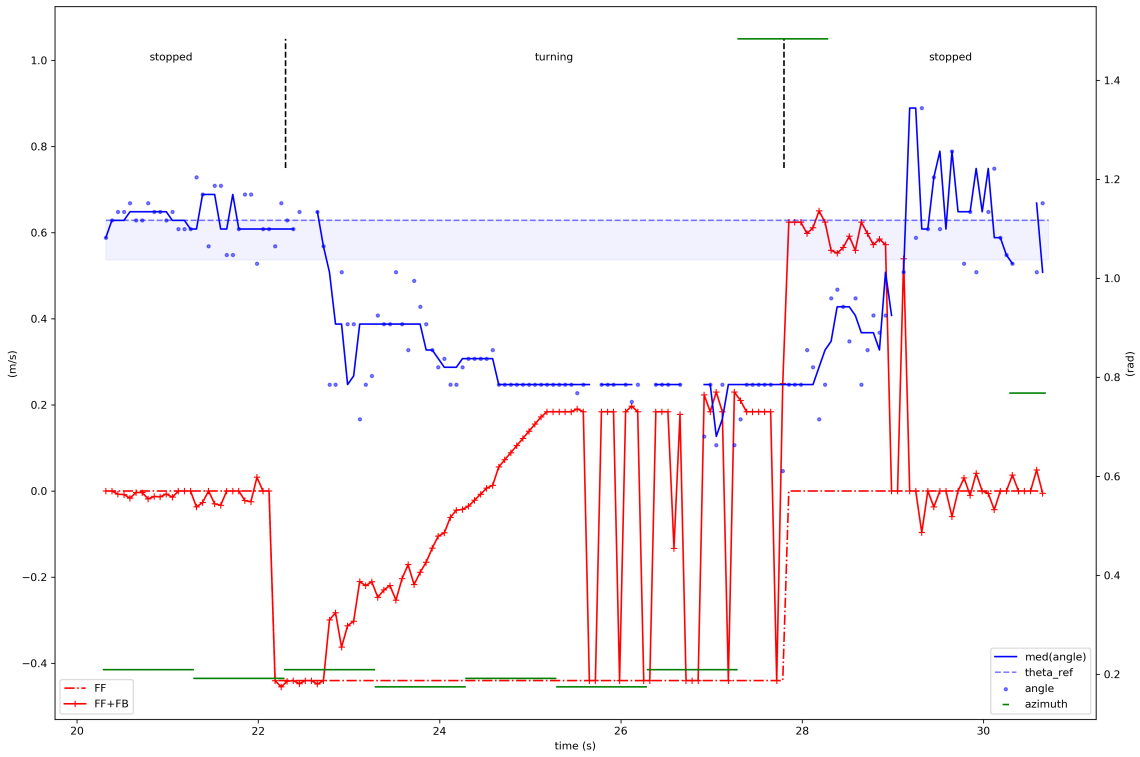


Figure 5.16: Feed-forward/Feedback control for $-\pi/2rad$ turns at $-0.389rad/s$.

on the azimuth of the cable (horizontal plane) relative to the rear of the UGV chassis. This azimuth feedback is updated at a much slower rate than the cable angle (vertical plane). In Figure 5.16, the vehicle is undergoing a turn through 90° in the clockwise direction, with an angular velocity command to the chassis of $-0.389rad/s$. Because of wheel slip throughout this zero-point turn, the total time to complete the turn is approximately $5.5sec$. The initial azimuth of the cable is slightly positive, taken with the rear-facing mid-line of the vehicle as zero and a z-up, right handed coordinate system. Accordingly, the sign of the angular component of the feed-forward controller is negative, and the cable begins to retract as the chassis turns after $22sec$. As the

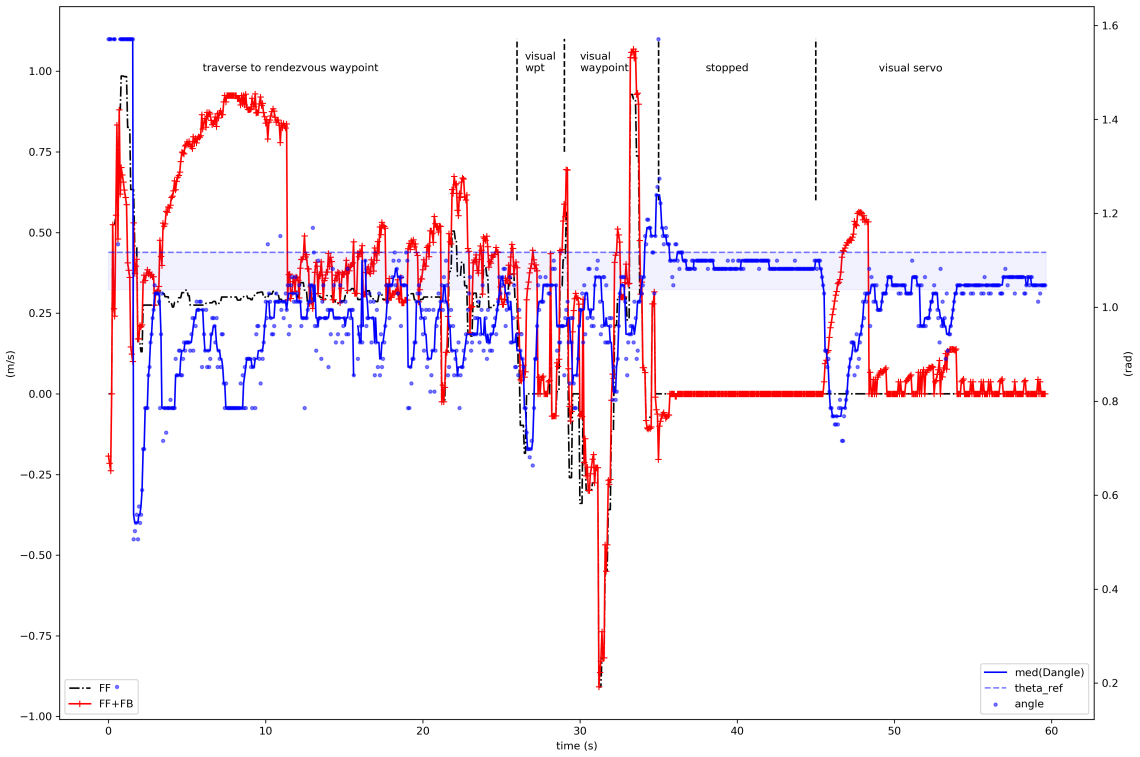


Figure 5.17: Feed-forward/Feedback control response with cable angle estimate for an outdoor cabling traversal and visual docking.

feedback controller reacts to the cable angle, the integrator winds up, eventually driving the cable deployment rate positive. This somewhat counter-intuitive result helps avoid entrapment of the cable in the chassis by initially tightening the cable (raising it away from the chassis), then easing the cable as needed. After chassis motion ceases, the feedback controller continues to deploy cable in the new orientation until achieving the angle set-point around 29*sec*. This zero-point turn is a fairly extreme maneuver compared to the smooth motions the chassis controller seeks to achieve in practice. Following the turn, the azimuth estimate has increased, as expected.

In practice, the navigation of the UGV compounds linear and rotational motion.

Figure 5.17 shows a representative control response in an outdoor traversal from cable start point, through the rendezvous waypoint and visual docking steps. Some regions of this plot which should be discussed include the starting transient, cable angle set-point recovery during relatively constant motion, waypoint achievement, steady-state, and a section of pure feedback control. Visual angle feedback data in this example shows very few incidences where the visual stack is unable to calculate and update the angle estimate, thus the feedback controller is active throughout the run whenever the feedback angle is outside the deadband region. The one exception to this is a start transient through the first second of motion, where a severe discontinuity is observed in the angle estimate. As expected, as the vehicle departs the genset, the cable is essentially horizontal. The feedback controller, with the help of the integrator, demands additional cable deployment speed up through the $16sec$ mark, where the visual angle estimate first falls within the deadband region. From this point through the achievement of the rendezvous waypoint, the feedback controller is able to make smaller adjustments to the feed-forward control signal. Rendezvous waypoint orientation adjustment happens around $25sec$, indicated by increased feed-forward control action to accommodate the chassis angular velocity changes.

From about $26sec$ to $35sec$ the vehicle achieves a series of visual docking waypoints. Although the total distanced traversed is small, discussed below, the control action by the UGV chassis controller in this case turn out to be much larger than in other portions of the run. In response, the feed-forward controller shows large excursions,

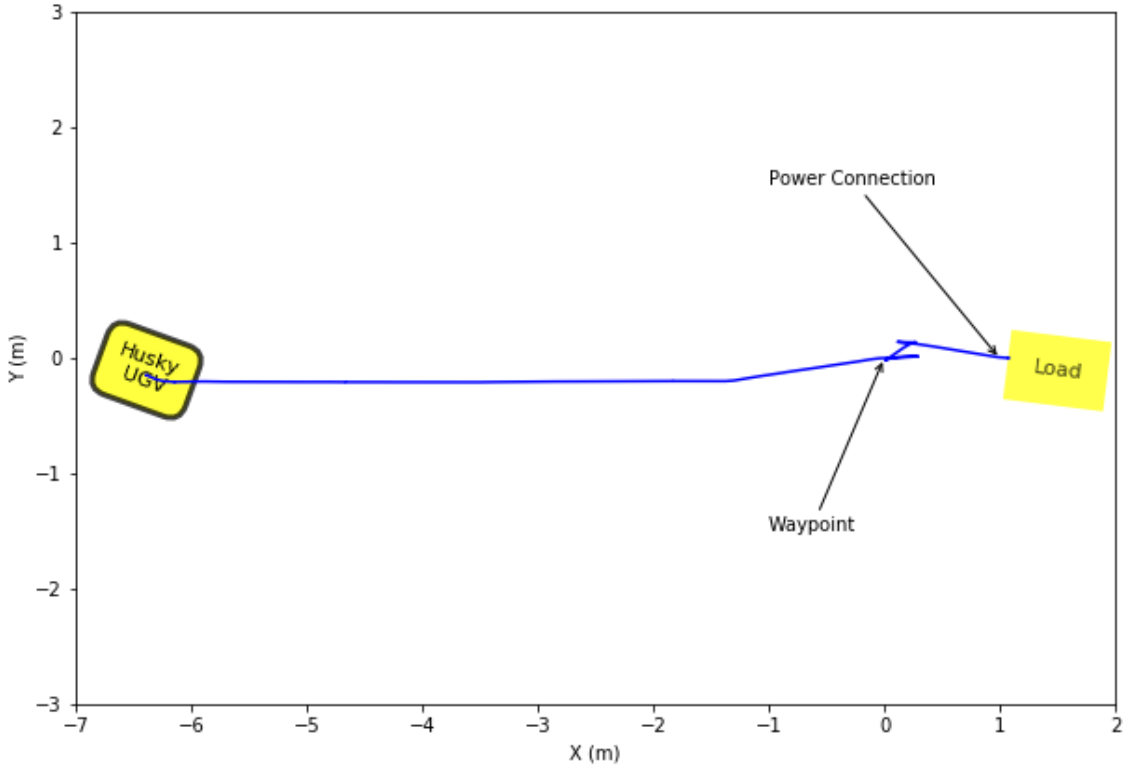


Figure 5.18: UGV trajectories while deploying cable between a genset and a load.

which are effective in keeping the angle of the cable within a reasonable bounds about the set-point. While the feedback controller is active, the data does not show large excursions due to the integrator action, as expected. After achieving the visual docking waypoint alignment, there is a steady state region where the chassis is not in motion. The cabling controller quickly drives the cable near the angle set-point within the deadband region. Finally, during the direct visual servo approach for insertion of the electrical connector, the chassis receives low-level velocity commands which do not activate the feed-forward cabling controller. Instead, the feedback controller

alone is able to respond to the relatively smooth forward motion of the chassis, from $45sec$ to the end of the run.

The vehicle EKF for the traverse and docking approach above is shown in Figure 5.18. For the purposes of elucidating the control response in the outdoor environment, the traverse region is obstacle-free. The UGV departs the genset location and follows a direct path to the rendezvous waypoint. Achievement of the rendezvous waypoint requires only a slight turn to fall within both the linear and angular success criteria. Because the rendezvous waypoint was well planned to coincide with the actual location of the load, specifically the visual docking waypoint, small linear and angular maneuvers are used by the chassis motion planner to re-position for the visual docking approach. While these maneuvers are small from the standpoint of the chassis planner, they result in large discontinuous velocity requirements at the cable as the chassis switches from forward to reverse motions, with relatively high angular velocities. As Figure 5.17 shows in the $26 - 35sec$ time window, the cabling controller is able to respond to these demands and drive the cable angle back within the deadband range of approximately $64°$.

Obstacle avoidance while deploying cable is presented in Figure 5.19. The UGV EKF trajectories over six autonomous approaches to a load rendezvous waypoint at $(3, 0)\ m$ are plotted, as well as locations of the obstacle for each trajectory. The floating connection was attached to a genset source at $(-3, 0)\ m$ and remained connected

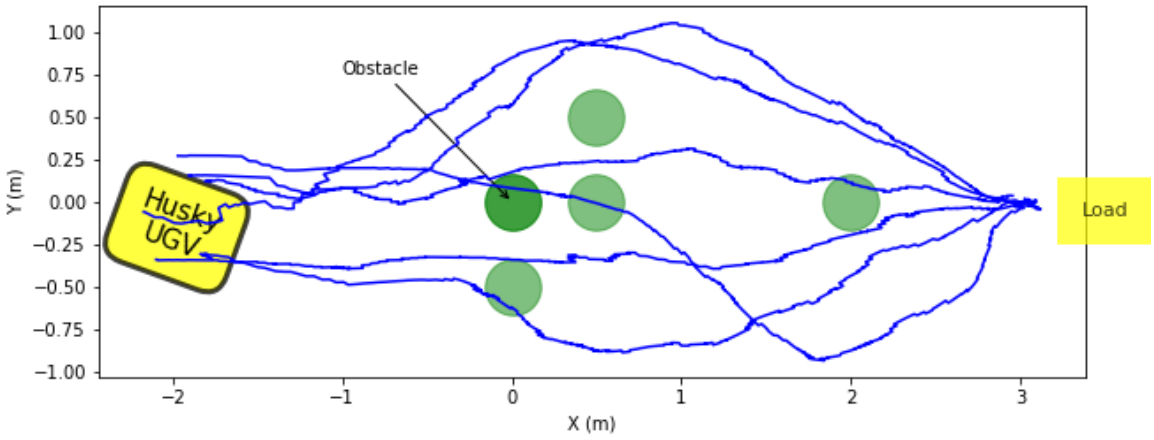


Figure 5.19: UGV trajectories while deploying cable with an obstacle in the UGV path.

throughout the trials. During one approach, the cable briefly touched the obstacle, otherwise the cable remained clear of the obstacle while the UGV navigated the environment. This scenario was reset between trials by manually driving the UGV in reverse, around the obstacle, to a start position near $(-2, 0)$ m, while the cable controller retracted cable. A representative image from these trials is included in Figure 5.20

5.5 Conclusion

This work has demonstrated a controller with visual feedback for autonomous electrical cable deployment from a UGV. Autonomous electrical cabling allows the formation of microgrids with distributed mobile energy resources in hazardous or human-denied

Figure 5.20: UGV deploying cable with an obstacle in the UGV path.

areas. This includes diverse applications encompassing disaster response, military forward operating bases, or planetary exploration. Controllers described in this work can feasibly be scaled up beyond the demonstration UGV and power grid hardware to existing mobile power assets, in larger operating theaters ($1kW$ to $100kW$ genset, $100m^2$ to $1000m^2$ operating area).

Valuable avenues to extend this work can be elucidated, especially in regard to the visual cable tracking and feedback estimate. First, an investment in labeling a representative cabling dataset, comprised of a variety of cables in different environmental or terrain conditions, would promote the training of a robust machine learning cable segmentation method. Beyond improvement of the visual feedback controller described here, this segmentation would allow tracking and estimation of the cable track along the ground at some distance from the UGV, with obvious benefits for

cable state estimation. This state estimation could be used to allow a single cabling UGV to deploy and recover multiple cables, or for other autonomous agents to avoid disturbing the cable. Further work could include a cable recovery path planner which fuses the prior estimates of both the UGV track and the cable pose, with updated estimates of the cable pose as it is followed back to its origin connection.

Chapter 6

Conclusion and Opportunity for Future Development

6.1 Conclustion

The contribution of this work is the development of a comprehensive architecture for power transmission by autonomous mobile microgrids. Purpose-specific UGV docking and cable deployment software algorithms, and hardware for electrical connection and cable management, has been developed for UGV electrical power transmission. To summarize the contributions described in this dissertation, a review of the chapter outcomes is provided here:

6.1.1 Chapter 2

Navigation and rendezvous of the autonomous UGV robots leverages Robot Operating System (ROS) tools as well as a visual feedback controller for docking. This dissertation presents a broad integrative effort to apply autonomous UGVs to the problem of mobile microgrid formation. This work has required custom hardware development, algorithms to support control of positioning of the UGV, and actuation of the power hardware. Robustness of the navigation and docking controllers is investigated with a Monte-Carlo test in an unstructured environment, with 92% docking success region identified for use in mission planning. This demonstration enables near-field wireless power transfer, and has since been extended to electrical connection coupling for cabling between power sources and infrastructure loads.

6.1.2 Chapter 3

A custom design "Adjustable Cable Management Mechanism" (ACMM) was required to meet low cost, compact platform constraints for powered deployment and retraction of electrical cable subject to disturbance. This ACMM fills an un-met need for a cabling deployment device for research-scale UGV robots. Valuable characteristics of the system include:

† Small size, weight, power consumption, and cost

† Deploys or retracts cable within the speed range of the UGV: up to $3.3ft/s$

† Compatible control interface with the UGV software and hardware platforms.

† Meets electrical requirements for the microgrid AC/DC power system.

6.1.3 Chapter 4

A probe-and-funnel AC/DC electrical connector system was developed for deployment on UGVs, which does not substantially increase the cost or complexity of the UGV, while providing a repeatable and secure method of coupling electrical contacts. The AC/DC power connector provides a required component in a versatile robot-based solution for wired power transmission. This devices has the following attributes:

† Purpose-designed to accommodate the docking behavior of an autonomous UGV.

† Mitigates docking alignment error up to +/-2cm in any direction and +/-15degrees.

† Supports electrical power transmission requirements for an AC/DC mobile microgrid.

† Does not require an expensive multi-degree of freedom manipulator for connection coupling.

6.1.4 Chapter 5

This work has demonstrated a controller with visual feedback for autonomous electrical cable deployment from a UGV. The feed-forward/feedback control method utilizes visual estimation of the cable state to deploy the electrical cable without tension, in the track of the UGV, as it transverses to connect power grid nodes. The work presents subsystem validation results, including control response to step-input UGV chassis velocities in the forward, reverse, and zero-point-turn maneuvers, and cable deployment in an unstructured environment outdoors.

6.2 Extensions and Future Work

Autonomous electrical cabling allows the formation of microgrids with distributed mobile energy resources in hazardous or human-denied areas. This includes diverse applications encompassing disaster response, military forward operating bases, or planetary exploration. In general, this work will allow critical power infrastructure deployment without exposing human operators to undue risk or harm. Portions of this

work are deployable on other vehicles or for new tasks with little additional work. This includes the visual docking via AR marker localization feedback. Some elements of this work invite additional investigation. Controllers described in this work can feasibly be scaled up beyond the demonstration UGV and power grid hardware to existing mobile power assets, in larger operating theaters ($1kW$ to 100 kW genset, 100 m^2 to 1000 m^2 operating area). The cable deployment task is relevant to diverse domains beyond the FOB or disaster response application, including underwater mining and planetary exploration. Hardware developed for the mobile microgrid task could be re-purposed for this research. Beyond the application of wired power for autonomous mobile microgrids, consider the application of the probe-and-funnel connector design for other high-wattage tasks, such as recharging of persistent-mission ground or aerial vehicles.

Valuable avenues to extend this work can be elucidated, especially in regard to the visual cable tracking and feedback estimate. First, improvement of the visual cable tracking could be pursued. An active marking scheme would simplify the visual processing stack. Alternatively, an investment in labeling a representative cabling dataset, comprised of a variety of cables in different environmental or terrain conditions, would promote the training of a robust machine learning cable segmentation method. Beyond improvement of the visual feedback controller described here, this segmentation would allow tracking and estimation of the cable track along the ground at some distance from the UGV, with obvious benefits for cable state estimation. This

state estimation could be used to allow a single cabling UGV to deploy and recover multiple cables, or for other autonomous agents to avoid disturbing the cable.

References

[1] "kilopower." [Online]. Available: https://www.nasa.gov/sites/default/files/thumbnails/image/stmd_kilopower_1.jpg

[2] A. Trebi-Ollennu, W. Kim, K. Ali, O. Khan, C. Sorice, P. Bailey, J. Umland, R. Bonitz, C. Ciarleglio, J. Knight, N. Haddad, K. Klein, S. Nowak, D. Klein, N. Onufer, K. Glazebrook, B. Kobeissi, E. Baez, F. Sarkissian, M. Badalian, H. Abarca, R. G. Deen, J. Yen, S. Myint, J. Maki, A. Pourangi, J. Grinblat, B. Bone, N. Warner, J. Singer, J. Ervin, and J. Lin, "InSight Mars Lander Robotics Instrument Deployment System," *Space Science Reviews*, vol. 214, no. 5, p. 93, Jul. 2018. [Online]. Available: https://doi.org/10.1007/s11214-018-0520-7

[3] J. B. Matthews and I. A. Nesnas, "On the design of the Axel and DuAxel rovers for extreme terrain exploration," in *2012 IEEE Aerospace Conference*, Mar. 2012, pp. 1–10, iSSN: 1095-323X.

[4] C. Abbey, D. Cornforth, N. Hatziargyriou, K. Hirose, A. Kwasinski, E. Kyriakides, G. Platt, L. Reyes, and S. Suryanarayanan, "Powering Through the Storm: Microgrids Operation for More Efficient Disaster Recovery," *IEEE Power and Energy Magazine*, vol. 12, no. 3, pp. 67–76, May 2014.

[5] A. Kwasinski, "Technological assessment of distributed generation systems operation during extreme events," in *2012 3rd IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Jun. 2012, pp. 534–541.

[6] S. Maharjan, Y. Zhang, S. Gjessing, O. Ulleberg, and F. Eliassen, "Providing Microgrid Resilience during Emergencies Using Distributed Energy Resources," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[7] S. Janko, S. Atkinson, and N. Johnson, "Design and Fabrication of a Containerized Micro-Grid for Disaster Relief and Off-Grid Applications," *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 2A, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1115/DETC2016-60296

[8] S. Lei, J. Wang, C. Chen, and Y. Hou, "Mobile Emergency Generator Pre-Positioning and Real-Time Allocation for Resilient Response to Natural Disasters," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[9] A. Kavousi-Fard, M. Wang, and W. Su, "Stochastic Resilient Post-Hurricane Power System Recovery Based on Mobile Emergency Resources and Reconfigurable Networked Microgrids," *IEEE Access*, vol. 6, pp. 72 311–72 326, 2018.

[10] J. Kim and Y. Dvorkin, "Enhancing Distribution System Resilience With Mobile Energy Storage and Microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 4996–5006, Sep. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8476200/

[11] L. Che and M. Shahidehpour, "Adaptive Formation of Microgrids With Mobile Emergency Resources for Critical Service Restoration in Extreme Conditions," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 742–753, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8440111/

[12] C. Chen, J. Wang, and D. Ton, "Modernizing Distribution System Restoration to Achieve Grid Resiliency Against Extreme Weather Events: An Integrated Solution," *Proceedings of the IEEE*, vol. 105, no. 7, pp. 1267–1288, Jul. 2017.

[13] G. Joos, J. Reilly, W. Bower, and R. Neal, "The Need for Standardization: The Benefits to the Core Functions of the Microgrid Control System," *IEEE Power and Energy Magazine*, vol. 15, no. 4, pp. 32–40, Jul. 2017.

[14] J. Gregory, J. Fink, E. Stump, J. Twigg, J. Rogers, D. Baran, N. Fung, and S. Young, "Application of Multi-Robot Systems to Disaster-Relief Scenarios with Limited Communication," in *Field and Service Robotics*, ser. Springer

Tracts in Advanced Robotics. Springer, Cham, 2016, pp. 639–653. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-27702-8_42

[15] W. Brinkmann, F. Cordes, T. M. Roehr, L. Christensen, T. Stark, R. U. Sonsalla, R. Szczuka, N. A. Mulsow, F. Bernhard, and D. Kuehn, "Modular payload-items for payload-assembly and system enhancement for future planetary missions," in *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–12.

[16] Weaver, Wayne W., Nina Mahmoudian, and G. Parker, "Autonomous mobile power blocks for prepositioned power conversion and distribution," in *TARDEC Ground Vehicle Systems Engineering and Technology Symposium*, 2012.

[17] B. Moridian, D. Bennett, N. Mahmoudian, W. W. Weaver, and R. Robinnett, "Autonomous Power Distribution System," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7–12, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016415843

[18] B. Moridian, D. Bennett, N. Mahmoudian, R. Robinett, and W. W. Weaver, "Design of Mobile Microgrid's Hierarchy for Power Distribution," *ASME 2015 Dynamic Systems and Control Conference*, vol. 3, Oct. 2015. [Online]. Available: http://dx.doi.org/10.1115/DSCC2015-9866

[19] B. Moridian, N. Mahmoudian, W. W. Weaver, and R. D. Robinett, "Robotic

power distribution system for post-disaster operations," in *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct. 2015, pp. 1–6.

[20] ——, "Postdisaster Electric Power Recovery Using Autonomous Vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 62–72, Jan. 2017.

[21] S. A. Darani, C. D. Majhor, W. W. Weaver, R. D. Robinett, and O. Abdelkhalik, "Optimal Positioning of Energy Assets in Autonomous Robotic Microgrids for Power Restoration," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4370–4380, Jul. 2019.

[22] G. Wang, Z. Xie, X. Mu, S. Li, F. Yang, H. Yue, and S. Jiang, "Docking Strategy for a Space Station Container Docking Device Based on Adaptive Sensing," *IEEE Access*, vol. 7, pp. 100 867–100 880, 2019.

[23] Z. Li, W. Liu, L.-E. Gao, L. Li, and F. Zhang, "Path Planning Method for AUV Docking Based on Adaptive Quantum-Behaved Particle Swarm Optimization," *IEEE Access*, vol. 7, pp. 78 665–78 674, 2019.

[24] S. Liu, M. Ozay, T. Okatani, H. Xu, K. Sun, and Y. Lin, "Detection and Pose Estimation for Short-Range Vision-Based Underwater Docking," *IEEE Access*, vol. 7, pp. 2720–2749, 2019.

[25] E. Narváez, A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "Autonomous VTOL-UAV Docking System for Heterogeneous Multirobot Team," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–18, 2021.

[26] F. Cocchioni, V. Pierfelice, A. Benini, A. Mancini, E. Frontoni, P. Zingaretti, G. Ippoliti, and S. Longhi, "Unmanned Ground and Aerial Vehicles in extended range indoor and outdoor missions," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2014, pp. 374–382.

[27] M. M. Kurdi, A. K. Dadykin, and I. Elzein, "Navigation of mobile robot with cooperation of quadcopter," in *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, Feb. 2017, pp. 30–36.

[28] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A Review of Tactile Information: Perception and Action Through Touch," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1619–1634, Dec. 2020.

[29] T. Klamt, M. Schwarz, C. Lenz, L. Baccelliere, D. Buongiorno, T. Cichon, A. DiGuardo, D. Droeschel, M. Gabardi, M. Kamedula, N. Kashiri, A. Laurenzi, D. Leonardis, L. Muratore, D. Pavlichenko, A. S. Periyasamy, D. Rodriguez, M. Solazzi, A. Frisoli, M. Gustmann, J. Roßmann, U. Süss, N. G. Tsagarakis, and S. Behnke, "Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance," *Journal*

of Field Robotics, vol. 37, no. 5, pp. 889–919, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21895

[30] A. Romay, S. Kohlbrecher, A. Stumpf, O. von Stryk, S. Maniatopoulos, H. Kress-Gazit, P. Schillinger, and D. C. Conner, "Collaborative Autonomy between High-level Behaviors and Human Operators for Remote Manipulation Tasks using Different Humanoid Robots," *Journal of Field Robotics*, vol. 34, no. 2, pp. 333–358, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21671

[31] B. P. W. Babu, R. Du, T. Padir, and M. A. Gennert, "Improving Robustness in Complex Tasks for a Supervisor Operated Humanoid," 2015. [Online]. Available: /paper/Collaborative-Autonomy-between-High-level-Behaviors-Romay-Kohlbrecher/e38cac783a07852e138b6cbc7f7828fa3bb86a25

[32] P. Lehner, S. Brunner, A. Dömel, H. Gmeiner, S. Riedel, B. Vodermayer, and A. Wedler, "Mobile manipulation for planetary exploration," in *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–11.

[33] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tethered robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 732–738, iSSN: null.

[34] A. Ortiz, J. Antich, and G. Oliver, "A particle filter-based approach for tracking undersea narrow telecommunication cables," *Machine Vision and Applications*, vol. 22, no. 2, pp. 283–302, Mar. 2011. [Online]. Available: https://doi.org/10.1007/s00138-009-0199-6

[35] X. Zhang and Q.-C. Pham, "Planning coordinated motions for tethered planar mobile robots," *Robotics and Autonomous Systems*, vol. 118, pp. 189–203, Aug. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889018309710

[36] D. Tsai, I. A. D. Nesnas, and D. Zarzhitsky, "Autonomous vision-based tethered-assisted rover docking," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 2834–2841, iSSN: 2153-0866.

[37] T. Brown, A. Stefanini, J. Sawoniewicz, I. Nesnas, and N. Georgiev, "Series Elastic Tether Management for Rappelling Rovers," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 2893–2900, iSSN: 2153-0866.

[38] P. McGarey, M. Polzin, and T. D. Barfoot, "Falling in line: Visual route following on extreme terrain for a tethered mobile robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2027–2034, iSSN: null.

[39] P. McGarey, D. Yoon, T. Tang, F. Pomerleau, and T. D. Barfoot, "Developing and deploying a tethered robot to map extremely steep terrain," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1327–1341, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21813

[40] J. Naglak, C. Greene, C. Majhor, N. Spike, J. P. Bos, and W. W. Weaver, "Autonomous Power Grid Formation for Surface Assets Using Multiple Unmanned Ground Vehicles," in *2020 IEEE Aerospace Conference*, Mar. 2020, pp. 1–8, iSSN: 1095-323X.

[41] J. E. Naglak, C. Kase, M. McGinty, C. D. Majhor, C. S. Greene, J. P. Bos, and W. W. Weaver, "Cable deployment system for unmanned ground vehicle (UGV) mobile microgrids," *HardwareX*, vol. 10, p. e00205, Oct. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468067221000341

[42] W. Brinkmann, T. M. Roehr, S. Natarajan, F. Cordes, R. U. Sonsalla, R. Szczuka, S. Bartsch, and F. Kirchner, "Design and evaluation of an end-effector for a reconfigurable multi-robot system for future planetary missions," in *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–10.

[43] P. McGarey, W. Reid, and I. Nesnas, "Towards Articulated Mobility and Efficient Docking for the DuAxel Tethered Robot System," in *2019 IEEE Aerospace Conference*, Mar. 2019, pp. 1–9.

[44] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernandez Perdomo, "ros_control: A generic and simple control framework for ROS," *The Journal of Open Source Software*, vol. 2, no. 20, p. 456, Dec. 2017. [Online]. Available: http://joss.theoj.org/papers/10.21105/joss.00456

[45] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The Office Marathon: Robust navigation in an indoor office environment," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 300–307.

[46] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 709–715.

[47] E. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. [Online]. Available: http://www-m3.ma.tum.de/foswiki/pub/MN0506/WebHome/dijkstra.pdf

[48] C. Rösmann, F. Hoffmann, and T. Bertram, "Planning of multiple robot trajectories in distinctive topologies," in *2015 European Conference on Mobile Robots (ECMR)*, Sep. 2015, pp. 1–6.

[49] T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in *Intelligent Autonomous Systems 13*, ser. Advances in Intelligent Systems and Computing. Springer, Cham, 2016, pp. 335–348. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-08338-4_25

[50] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[51] S. Siltanen, *Theory and Applications of Marker-based Augmented Reality.* VTT, Finland: VVT Technical Research Center of Finland, 2012. [Online]. Available: http://mehmetsimsek.net/slides/bm533/S3.pdf

[52] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study," *arXiv:1404.3785 [cs]*, Apr. 2014, arXiv: 1404.3785. [Online]. Available: http://arxiv.org/abs/1404.3785

[53] C. Greene, J. Naglak, C. Majhor, J. P. Bos, and W. W. Weaver, "Near Field Wireless Power Transfer via Robotic Feedback Control," in *2020 IEEE Aerospace Conference*, Mar. 2020, pp. 1–7, iSSN: 1095-323X.

[54] C. D. Majhor, J. E. Naglak, C. S. Greene, W. W. Weaver, and J. P. Bos, "Recharging of Distributed Loads via Schedule Optimization with Autonomous

Mobile Energy Assets," in *2020 IEEE Aerospace Conference*, Mar. 2020, pp. 1–9, iSSN: 1095-323X.

[55] J. Iqbal, S. Heikkila, and A. Halme, "Tether tracking and control of ROSA robotic rover," in *Robotics and Vision 2008 10th International Conference on Control, Automation*, Dec. 2008, pp. 689–693, iSSN: null.

[56] M. Krishna, J. Bares, and E. Mutschler, "Tethering system design for Dante II," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, Apr. 1997, pp. 1100–1105 vol.2.

[57] P. Younse, "Launchable and retrievable tetherbot exploration system," in *Space Exploration Technologies*, vol. 6960. International Society for Optics and Photonics, Apr. 2008, p. 69600J. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/6960/69600J/Launchable-and-retrievable-tetherbot-exploration-system/10.1117/12.781708.short

[58] S. Krause and S. Cain, "UAV Pre-Study for In-Air-Capturing Maneuver," in *2020 IEEE Aerospace Conference*, Mar. 2020, pp. 1–14, iSSN: 1095-323X.

[59] M.-H. Khooban, M. Gheisarnejad, N. Vafamand, M. Jafari, S. Mobayen, T. Dragicevic, and J. Boudjadar, "Robust Frequency Regulation in Mobile Microgrids: HIL Implementation," *IEEE Systems Journal*, vol. 13, no. 4,

pp. 4281–4291, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8721112/

[60] J. Naglak, T. Barrett, N. Bondi, S. Krusinski, M. McGinty, T. Moon, J. Bos, and W. Weaver, "(Submitted) AC/DC Power Connector for Unmanned Ground Vehicle(UGV) Mobile Microgrids," *Hardware X*, 2021.

[61] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87–116, Oct. 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0734189X88800331

[62] M. McCauley, "AccelStepper: AccelStepper library for Arduino." [Online]. Available: http://www.airspayce.com/mikem/arduino/AccelStepper/

[63] D. Austin, "Generate stepper-motor speed profiles in real time," Jul. 2014. [Online]. Available: http://web.archive.org/web/20140705143928/http://fab.cba.mit.edu/classes/MIT/961.09/projects/i0/Stepper_Motor_Speed_Profile.pdf

[64] N. Metropolis, "The Beginning of the Monte Carlo Method," *Los Alamos Science*, 1987.

[65] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3, iSSN: null.

[66] A. Renawi, M. A. Jaradat, and M. Abdel-Hafez, "ROS validation for non-holonomic differential robot modeling and control: Case study: Kobuki robot trajectory tracking controller," in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, Apr. 2017, pp. 1–5, iSSN: null.

[67] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2016, pp. 96–101.

[68] D. Meltz and H. Guterman, "RobIL — Israeli program for research and development of autonomous UGV: Performance evaluation methodology," in *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, Nov. 2016, pp. 1–5, iSSN: null.

[69] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. von Stryk, "Hector Open Source Modules for Autonomous Mapping and Navigation with Rescue Robots," in *RoboCup 2013: Robot World Cup XVII*, ser. Lecture Notes in Computer Science, S. Behnke, M. Veloso, A. Visser, and R. Xiong, Eds. Berlin, Heidelberg: Springer, 2014, pp. 624–631.

[70] C. E. Agüero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo, E. Krotkov, and G. Pratt, "Inside the Virtual

Robotics Challenge: Simulating Real-Time Robotic Disaster Response," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 494–506, Apr. 2015.

[71] "Open Dynamics Engine." [Online]. Available: http://ode.org/ode-latest-userguide.html#sec_3_11_1

[72] P. D. Cenek, N. J. Jamieson, and M. W. McLarin, "Frictional Characteristics of Roadside Grass Types," in *International Surface Friction Conference, 2005*, Christchurch, New Zealand, 2005.

[73] C. Majhor, "Optimal Mission Planning of Autonomous Mobile Agents for Applications in Microgrids, Sensor Networks, and Military Reconnaissance," *Dissertations, Master's Theses and Master's Reports*, Jan. 2020. [Online]. Available: https://digitalcommons.mtu.edu/etdr/1009

Appendix A

Visual Docking for Mobile Microgrid Power Transfer: Simulation and Validation

A.1 Introduction

Mobile microgrids combine the functionality of independent power grids with multi-domain, multi-purpose, autonomous Unmanned Ground Vehicles(UGV). Applications with human risk or human denied environments are especially valuable, which

range from military forward operating bases or disaster response, to planetary infrastructure. Micro power grids incorporate AC/DC conversion, load balancing or load shedding, power transmission, energy generation with grid storage, and grid optimization or control. Additionally, mobile microgrids must optimize for both autonomous vehicle constraints and the power grid requirements. Power grid hardware is deployed on the mobile robots, which configure according to mission requirements and any fixed load or power source infrastructure. If conditions change, the mobile grid elements should be able to optimally reconfigure.

Very little work is available in the literature regarding the deployment of power grids on mobile robots. For the purpose of mobile microgrids, docking supports a wireless power transfer task, as well as wired electrical connection coupling. These tasks depend on *mm*-scale docking precision, with maximum of any misalignment angle $< 10°$.

Robot Operating System (ROS) is an open-source ecosystem of common drivers, algorithms, and tools that support autonomous robotic development. ROS has become closely integrated with an open simulation engine called Gazebo, which supports many common robotic platforms and sensing systems [65]. Comparison between Gazebo simulation and actual robot trajectory performance is explored in [66, 67], but is limited to an indoor environment. Difference in actuator effort between gazebo and an off-road test environment for a differential drive robot is evaluated [68]. A

standardized evaluation environment for rescue UGVs is available in Gazebo [69]. Despite these efforts, the literature does not explore a mobile microgrid application, apart from the previous results reported by Michigan Technological University [16, 17, 18, 19, 20, 21, 40, 53, 54].

While there is ample work comparing ROS with Gazebo simulation to actual robot performance, no work is available to determine how well ROS with Gazebo will simulate the real-world conditions of mobile microgrid formation. The contribution of this work is a comparison between the performance of the visual docking controller for mobile microgrid power transfer as simulated in Gazebo using a Monte-Carlo approach, with the true behavior of the system in equivalent outdoor demonstration trials. The results obtained by each experiment are compared, and the suitability of Gazebo as a simulation tool for mobile microgrid formation is discussed.

The contents of this report are as followes: first the author details the Microgrid UGV platform, then specifications of the ROS and Gazebo systems will be provided. The docking controller will be described and the Monte-Carlo parameters will be defined. After the test environments in Gazebo and outdoors have been documented, the results will be reported. Finally, we will discuss conclusions and future work.

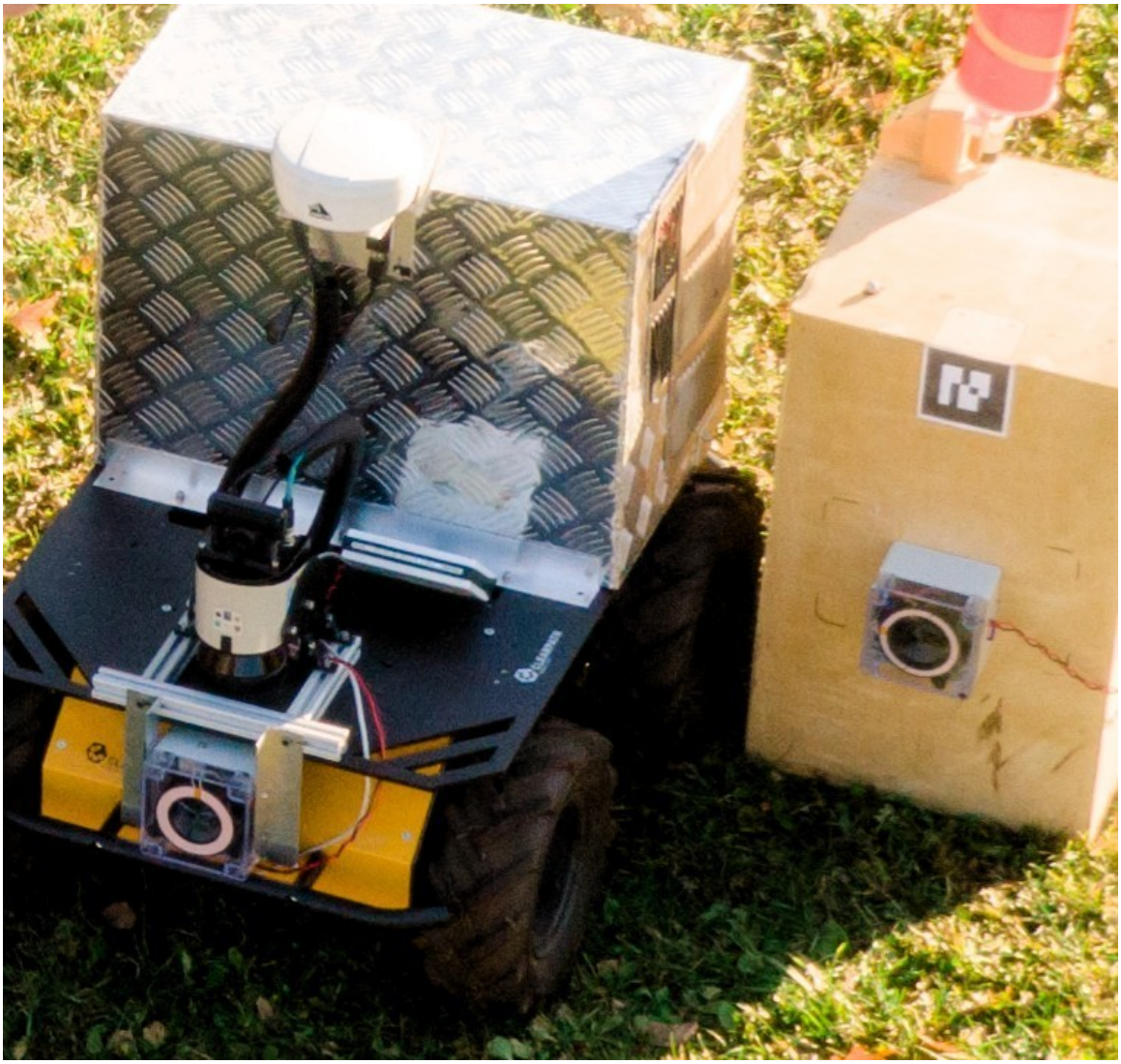


Figure A.1: Ruggedized UGV microgrid agents are equipped with a sensor suite appropriate for their operating environment, as well as power hardware. In this example, the UGV carries a generator and transfers power via wireless charging coil. The load coil location is indicated by an AR marker.

A.2 Microgrid UGV Configuration

Clearpath Husky UGVs constitute the fleet of development vehicles for the work reported in this paper, one of which is shown in Figure A.1. These Husky UGVs

shipped with a Mini-ITX computer, running an Intel i5 processor at $2.9GHz$ and Ubuntu 14.04. Sensors include LORD MicroStrain 3DM-GX3-25 9-DOF IMU and NovAtel SMART6-L GPS receiver, as well as the SICK LMS-111 2D LiDAR.

Microgrid Husky's are equipped with a range of power grid hardware. The husky used in the work has a gas Honda $1100W$ generator mounted in a grounded enclosure. Other agents have grid attached storage, electrical bus, or power cabling and connection coupling hardware. Another feature of this microgrid robot is wireless power transfer primary coil, mounted directly above the front bumper of the UGV. Details of this implementation have been published elsewhere [53], but it is important to note that the quality of the docking pose, in regard to both displacement and angular error, greatly effects the efficiency of the wireless power transfer.

A.3 Visual Controller for Docking

Visual docking is performed using a combination of open-source algorithms available in ROS, and set of task-specific calculations based on visual feedback from an Augmented Reality(AR) marker adjacent to the docking goal at a load. ROS platform elements can be replicated based on the configurations from parameter files.

A.3.1 ROS move_base Implementation

A description of the ROS waypoint navigation method and a summary of the important configuration parameters for the move_base elements of the navigation stack as deployed for this testing is provided in [40].

A.3.2 Iterative Visual Feedback

Microgrid UGV agents need to be able to achieve accurate positioning adjacent to infrastructure or other robotic agents before initiating power coupling. This docking controller needs a continuous exteroceptive source of localisation data. A viable solution is to implement a visual feedback method using a calibrated [50] monocular camera and Augmented Reality (AR) tracking of fiducial markers [51]. A fixed-eye camera is installed on each UGV. A zero-turn pan method to acquire the augmented reality marker in the camera frame when needed during the docking maneuvers.

After the rendezvous waypoint is achieved, the docking controller checks to see if the AR marker is in the camera field of view. If it cannot locate the marker, it engages a pan controller until the marker is acquired. Once acquired, it calculates a docking pose offset from the marker and passes this pose (x,y,θ) to the ROS Navigation stack as a waypoint goal. This waypoint goal is more accurate to the true pose of

the docking target than any pre-planned rendezvous waypoint. Once the waypoint controller completes, the UGV is relatively near to the AR marker, which presents the best estimate of its position. It is advantageous to recalculate the docking pose offset and generate an updated waypoint goal.Finally, the agent visual servos to the final docking pose. The pose error, e_x and e_y, between the robot chassis pose and the docking pose in a $2-$D cartesian frame is calculated using the AR marker frame in ROS tf. A proportional feedback loop is engaged, where $\dot{x}$ and $\dot{\theta}$, the linear and angular velocity commands, are calculated $\dot{x} = P_x\ e_x$ and $\dot{\theta} = P_\theta\ e_y$ where P_x and P_θ were tuned empirically. This behavior executes until e_x drops below a threshold distance. At this time, the docking method exits and signals the mission controller to proceed to the next behavior.

A.4 Monte-Carlo Validation Method

One practice for assessing robustness is to introduce a stochastic disturbance into a system over many trials and observe the system response [64]. This Monte-Carlo type method can be leveraged to validate the autonomous mobile microgrid docking controller. Ideal mission configurations may not expose failure modes and edge cases, so a Monte-Carlo test regime should be designed to explore the limits of the capability of the controller. Another benefit of the Monte-Carlo regime is that it can deliver stochastic performance measures that are important for mission planning [54].

Docking of the UGV is a multi-step process, subject to different failure modes at each step. Under ideal conditions, a rendezvous waypoint is offset from a docking goal by $1m$ along the target normal, with the docking camera facing the target AR marker. The validation test seeks to understand the effect of offset from this ideal pose along the target normal, orthogonal to this normal, and also in yaw orientation. The limits for the region of investigation are the camera resolution (maximum distance), and the assumptions of the kinematic planner regarding obstacle avoidance (minimum distance). A half-washer shaped region that satisfied these requirements was allocated about the target and populated with uniform random rendezvous waypoints. The minimum radius about the target face was 1 m, the maximum radius was 3 m. Angles were constrained to $1/2\ \pi$ either side of the normal vector pointing in to the marker. An automated test routine was developed that included a short reset traverse away from the final docking pose, Figure A.2.

A secondary Monte-Carlo test was designed to asses the traversal cost between rendezvous waypoints. This test consisted of uniformly distributed pose goals throughout an $8mx8m$ region within the test environment. The traversal cost is generalized to time between poses normalized by the scalar distance between them.

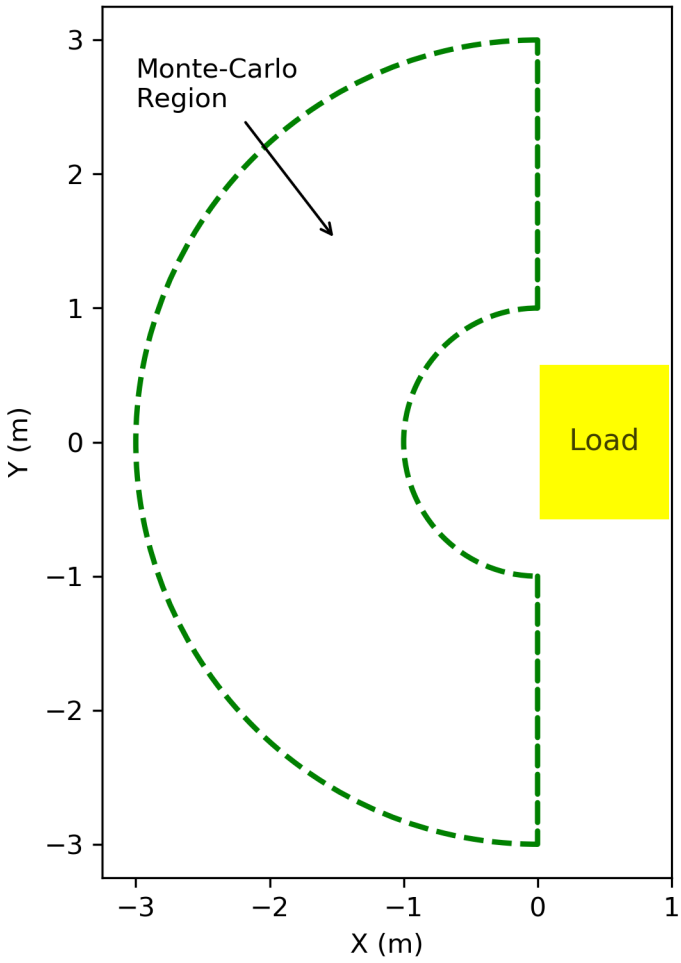


Figure A.2: Pose goals were distributed about the face of the docking goal in a uniform random distribution throughout a half-washer shaped region

A.4.1 Gazebo Simulation Environment

An outdoor environment was simulated in Gazebo. Grass material was used from the open source Darpa Robotics Challenge Simulation (DRCsim) [70], with surface friction coefficients of $\mu = 0.5$ and $\mu^2 = 0.5$. These coefficients are a simplification of the friction cone model into a friction pyramid [71]. Early evaluation of the comparison between this environment and the performance of the UGV in the real-world grass surface showed some discrepancy, possibly related to a poor match for the coefficient μ. Based on an evaluation of μ for car tires on different varietals of dry and wet grass in [72], a parameter search was performed for $\mu, \mu^2 = [0.36, 0.5, 0.77, 1.54]$. Similar to the actual outdoor test configuration, a fixed load was positioned at the local origin, with an AR tag indicating the docking goal. No provision was made at this time to simulate the interaction physics of the wireless power transmission. This environment is shown in Figure A.3.

A.4.2 Outdoor Test Environment

At this time, only a grass test region has been explored. The region is portrayed in Figure A.4. The vehicles EKF filter was utilized to locate the load at the $0m, 0m$ origin point associated with a UTM datum point in the test region. The load then

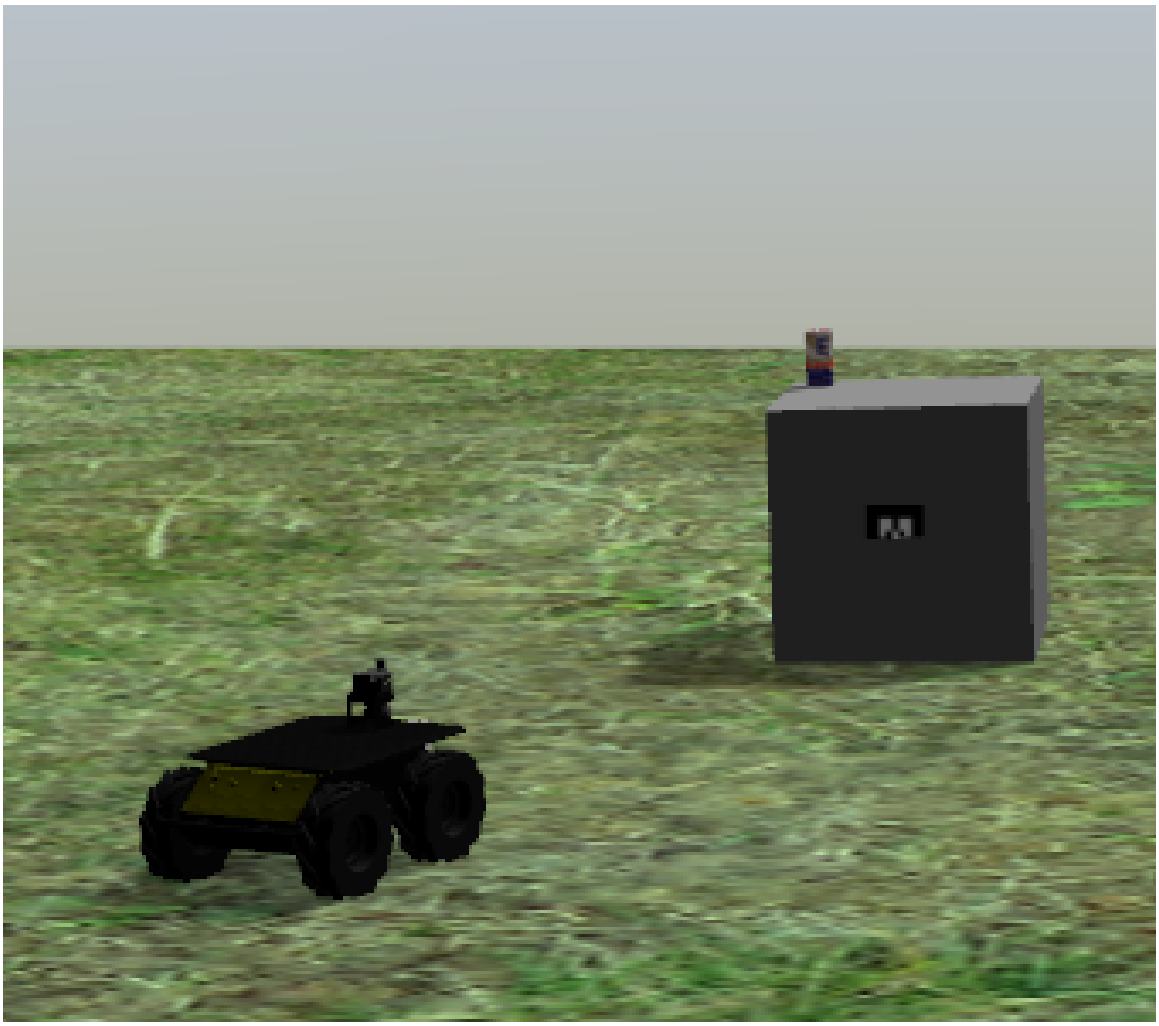


Figure A.3: An outdoor environment with a fixed load was modeled in Gazebo, with a flat grass surface.

remained in that location for the duration of the Monte-Carlo trials.

Figure A.4: The outdoor test environment was a generally open and covered in grass.

A.4.3 3 Load Recharging

To further leverage the capabilities of this mobile microgrid simulator, the tool was applied to a real-time schedule for maintaining State of Charge (SoC) on three distributed loads, using wireless power transmission. The parameters for traversal and docking time cost determined by a limited indoor Monte-Carlo scheme were introduced into an optimizer, outside the scope of this work [73], and a schedule for the robot was calculated. The robot then was deployed to visit the 3 loads according to the schedule in simulation over a 24 hr time period.

Table A.1
Success rate and arc for multiple values of μ on a simulated grassy surface.

μ	% **Success**	**Arc** (°)
0.36	45	
0.5	50	100
0.77	60	
1.56	60	

A.5 Results

Monte-Carlo simulation of waypoint navigation and visual docking is performed in simulation and in corresponding outdoor environments. These results reveal similarities and differences between the vehicle performance in simulation vs. outdoor deployment. Specifically, mean and standard deviation for each stage of docking is reported, as well as overall success rate and limitations in the extents of rendezvous region between the docking UGV and the goal load.

A.5.1 Simulation

The Monte-Carlo simulation in Gazebo consisted of 80 docking attempts across four parameter configurations for the friction coefficient μ, shown in Figure A.8.

Success rate and empirical success arc for each of the values of μ is shown in table A.1.

Consider the results for $\mu = 0.5$ Only 50% of these attempts were successful, indicated by black arrows. From these successful approaches, a region comprised of a 100° arc centered on the docking goal can be considered a viable location for a rendezvous waypoint. Some important differences between these results and the results from the outdoor trials should be elucidated. First, the pan control responded differently in the simulated grass compared to actual grass. The amount of rotation of the UGV chassis attained in the simulation was somewhat limited compared to in the outdoor environment. This is likely the cause of the limited arc of successful docking in the simulated environment. Finally, in simulation we do not observe from the data a hard limit to the distance from the AR marker which the tracking algorithm is able to resolve. In fact, there is a software limit for this distance, and a noise parameter for the simulated camera image. Compared the real conditions outdoors, the simulated camera shows an unrealistic range which it is able to acquire the AR marker and perform the docking maneuver.

Traversal cost is estimated to be $0.227 + / - 0.078 m/s$ in the grassy simulation environment, with $\mu, \mu^2 = 1.54$. Monte-Carlo waypoint poses and the UGV trajectory is shown in Figure A.9.

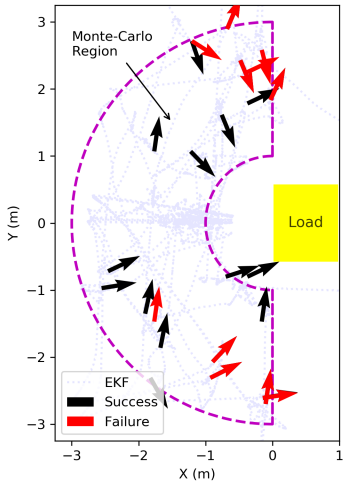


Figure A.5: Trials for $\mu = 0.36$

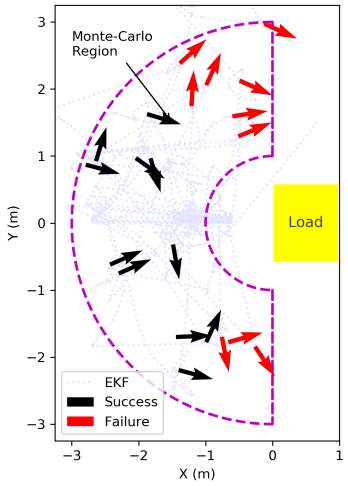


Figure A.6: Trials for $\mu = 0.5$

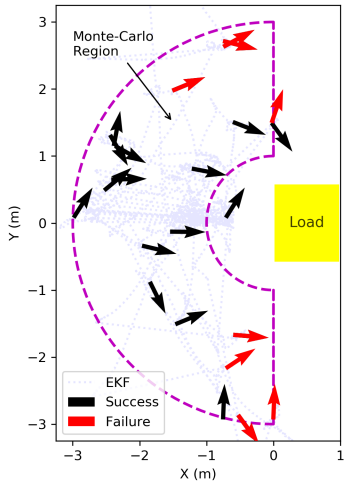


Figure A.7: Trials for $\mu = 0.77$

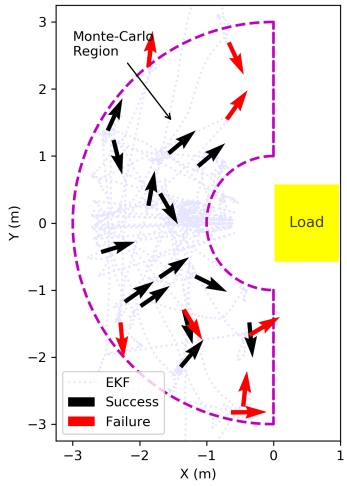


Figure A.8: Trials for $\mu = 1.54$

A.5.2 Outdoor Validation

Outdoor implementation of the Monte-Carlo validation method produced a set of 18 docking approaches. For goal poses in the uniform-random distribution of $3m$

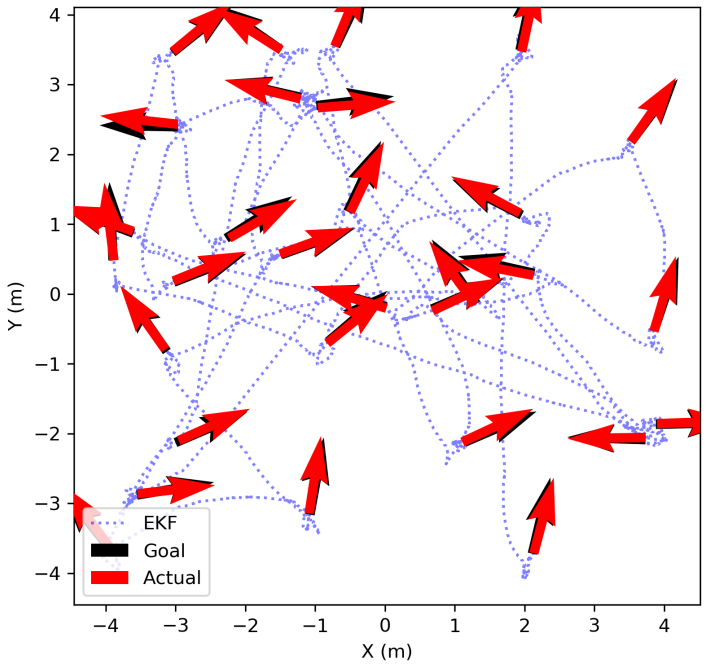


Figure A.9: Monte-Carlo evaluation of traversal time cost between waypoint poses in the Gazebo simulation environment.

about the AR marker, the docking success rate was 73%. This region was chosen to encompass, but exceed, the expected success region for the docking method. Figure A.10 shows the docking region, as well as a reduced region which shows 92% docking success rate within a 2.25m radius, 160° arc, about the AR marker. On average, it took 83s from the time that the UGV achieved the rendezvous waypoint until it completed its docking approach, with a standard deviation of 43s.

These data demonstrate two failure modes of the visual AR tracking method. First, there is a maximum range for which the method can successfully resolve the AR

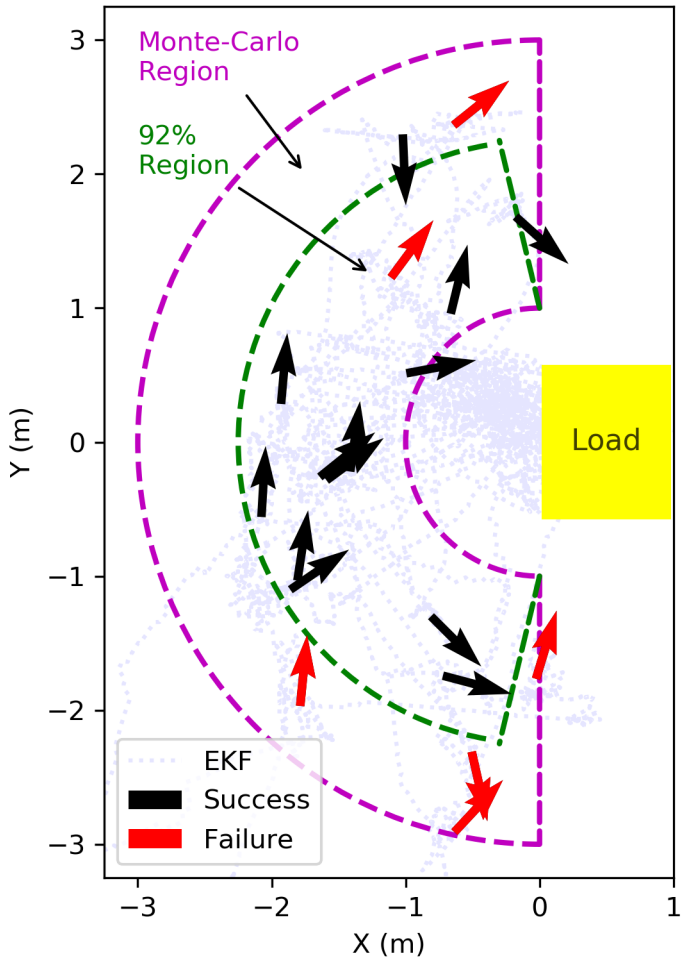


Figure A.10: Outdoor Monte-Carlo validation results consisted of 18 docking trials over a region that was expected to exceed the capability of the controller.

marker. This is distance is directly related to marker size and to camera resolution. Second, there is some limit of viewing angle between camera and AR marker where the skew of the marker defeats the tracking algorithm. Operationally, there are concrete improvements that can be made to enhance the range at which the docking method

can resolve rendezvous uncertainty. A dual marker scheme could be employed, with a large marker for initial localization, and a smaller marker which will remain in the camera frame during the final docking approach. Alternatively, a more sophisticated camera could be installed, especially one with higher resolution and better optical properties.

There are some differences between the simulation and the outdoor results. Expectation held that the simulation results would have a higher success rate, but that has not proved to hold with the limited results for grass. One divergence between simulation and an outdoor environment lies in the disturbance from the tire-ground interaction, which is captured by a single friction parameter in the simulation. Actually, this interaction is non-linear, that is to say that a single parameter does not represent the interaction well over small distances or individual disturbances (bumps, changes in moisture content). These disturbances are especially important during final precision manoeuvres.

Traversal cost is estimated to be $0.257 + / - 0.115 m/s$ in the grassy simulation environment, with $\mu, \mu^2 = 1.54$. Monte-Carlo waypoint poses and the UGV trajectory is shown in Figure A.11. Black arrows indicate the waypoint pose goal, with red arrows indicating the pose attained within the achievement tolerances of the planner.

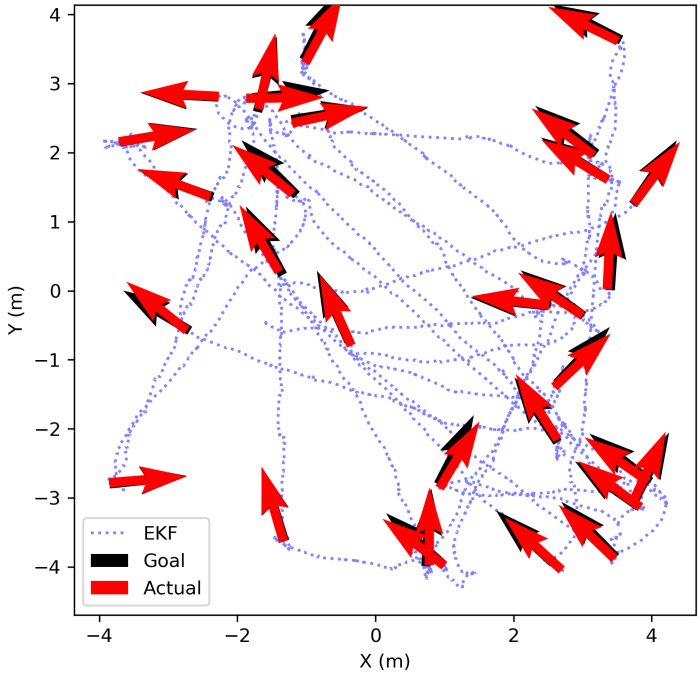


Figure A.11: EKF filtered trajectories of the UGV transiting between random waypoints in an outdoor, grassy, environment.

A.5.3 3 Load Recharging

Monte-Carlo evaluations of navigation time cost, and docking time cost and repeatability, were determined in an indoor 3 load environment, Figure A.13. This Monte-Carlo study provides parameters inputs to mission planning and power optimization tools, available in [73]. Based on these parameters, an indoor, single-UGV power

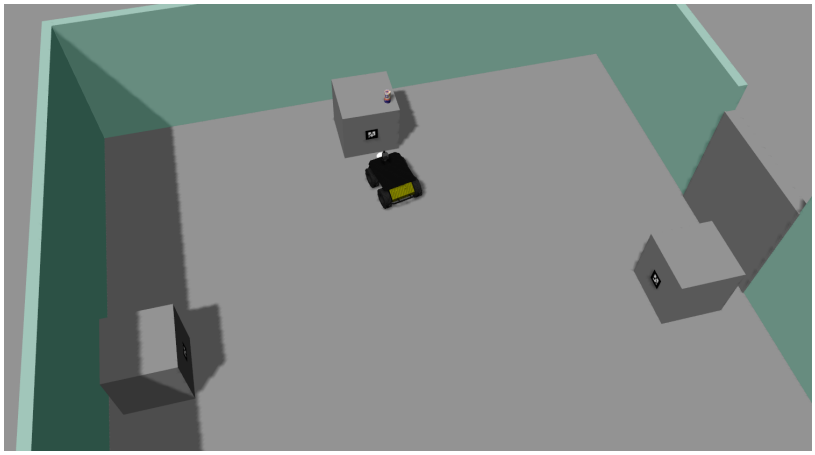


Figure A.12: A UGV with wireless power transmission approaches one of 3 loads to recharge it in simulation.

source, 3 DC load, wireless recharging schedule was calculated and deployed in real-time simulation, Figure A.12. The performance of the UGV in the simulated environment corresponded well to the model used to calculate the optimized recharging schedule. The Gazebo simulated system achieves charging goals more quickly than expected, thus the loads maintain a higher State of Charge(SoC) than estimated. This is in line with other simulation comparisons where the simplified disturbances yield slightly improved performance of the navigation and docking controllers.

A.6 Conclusion

This report uses a series of Monte-Carlo scenarios to show that ROS and the Gazebo simulation environment can be used to model and control the rendezvous and docking

Figure A.13: 3 loads in a laboratory environment, with charging UGV making a wireless power transmission docking approach.

of UGV robots for mobile microgrid formation. Mission planning and microgrid optimization can now be predicated on real-world parameters for traversal cost, docking time cost, and docking success rate. Rendezvous waypoint selection is informed by the 92% docking success region determined through outdoor testing. Additionally, specific planning scenarios can be evaluated in the Gazebo simulation environment with confidence that the real-world implementation will correspond to the simulated behavior.

To extend this work to encompass the entire control method for mobile microgrids, the electrical connection coupling and cable deployment systems can be modeled in Gazebo. The entire system should be validated through a similar Monte-Carlo approach. Additional future adaptation would be the introduction of scenario-specific infrastructure within the environment, as well as terrain features associated with the mission environment.

Appendix B

Sample Code

For the readers reference, the controllers outlined in Chapter 5 are defined in the following code:

B.1 Cable Visual Tracking

```
#!/usr/bin/env python

import cv2
import numpy as np
from scipy.stats import linregress as lr
from matplotlib import pyplot as plt
import rospy
```

```
from sensor_msgs.msg import Image
from cv_bridge import CvBridge, CvBridgeError
from std_msgs.msg import Float32
import tf
from tf.transformations import euler_from_quaternion


class cableSegmenter():
  """cableSegmenter visually tracks the power cable"""
  def __init__(self):
    rospy.init_node('cableVisualTracking', anonymous=↩
       True)
    rate=rospy.Rate(1)

    self.br = tf.TransformBroadcaster()

    #ROS publisher for cable overlay image
    self.cablepub = rospy.Publisher('cable',Image, ↩
       queue_size=1)
    self.cableazpub = rospy.Publisher('cable_az',Image, ↩
       queue_size=1)
    self.anglepub = rospy.Publisher('spool/angle',↩
       Float32, queue_size=10)
    self.azimuthpub = rospy.Publisher('spool/azimuth',↩
       Float32, queue_size=10)

    #ROS subscriber to image topic with bridge to OpenCV
    self.bridge=CvBridge()

    #subscriber to the image topic
    image_sub = rospy.Subscriber('usb_cam_side/image_raw↩
       ',Image, self.callbackROI)
    imagerear_sub = rospy.Subscriber('usb_cam_rear/↩
       image_raw',Image, self.rearcallbackROI)
```

```
    self.lower=np.array([90,150,100])      #For HSV, Hue ↩
       range is [0,179], Saturation range is [0,255] and ↩
       Value range is [0,255]
    self.upper=np.array([98,250,255])    #[47,200,150]
    self.rearlower=np.array([90,150,100])      #For HSV, ↩
       Hue range is [0,179], Saturation range is [0,255] ↩
       and Value range is [0,255]
    self.rearupper=np.array([98,250,255])    #↩
       [47,200,150]

    self.slope = 0.0
    self.intercept = 0
    self.azimuth = 0.0

    print("Here we go...")

    while not rospy.is_shutdown():
      #let the callback do the work as long as ROS is ↩
         running
      self.cableTransformer()
      rate.sleep()


def estAngle(self,image):

    x,y = np.nonzero(image)
    if x.any():
      linfit = lr(y,x)
      return linfit.slope, int(linfit.intercept)
    else:
      return np.nan,np.nan
```

```
def callbackROI(self,data):
  try:
    frame=self.bridge.imgmsg_to_cv2(data, "bgr8")#↩
       desired_encoding="passthrough")
  except CvBridgeError as e:
    print(e)

  #find the cable
  hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)

  mask=np.zeros(hsv.shape[:2], np.uint8)
  mask[-50:,0:50] = 255

  mean, stddev = cv2.meanStdDev(hsv,mask=mask)
  scd = 0.75
  self.lower=np.array([np.floor(mean[0]-stddev[0]*scd)↩
     ,np.floor(mean[1]-stddev[1]*scd),
      np.floor(mean[2]-stddev[2]*scd)])      #For HSV, ↩
         Hue range is [0,179], Saturation range is ↩
         [0,255] and Value range is [0,255]
  self.upper=np.array([np.ceil(mean[0]+stddev[0]*scd),↩
     np.ceil(mean[1]+stddev[1]*scd),
      np.ceil(mean[2]+stddev[2]*scd)])

  mask2=np.zeros(hsv.shape[:2], np.uint8)
  mask2[:,-400:] = 255
        dangleROI_hsv = cv2.bitwise_and(hsv,hsv,mask =↩
            mask2)

  maskYellow=cv2.inRange(dangleROI_hsv,self.lower,self↩
     .upper)

  blur = cv2.GaussianBlur(maskYellow ,(25,25),0)
  edge=cv2.Canny(maskYellow,100,400)
```

```
lines = cv2.HoughLines(edge, 1, np.pi / 180, 20)
# Draw the lines
if lines is not None:
  for i in range(0,1):# len(lines)):
    rho = lines[i][0][0]
    theta = lines[i][0][1]
    a = np.cos(theta)
    b = np.sin(theta)
    x0 = a * rho
    y0 = b * rho
    pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
    pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))
    wline = cv2.line(edge, pt1, pt2, (255,0,0), 3, ↩
       cv2.LINE_AA)
  self.slope = lines[0][0][1]
  pt1 = (640-525,0)
  pt2 = (640-525,480)
  wline = cv2.line(edge, pt1, pt2, (255,0,0), 1, cv2↩
     .LINE_AA)
  self.cablepub.publish(self.bridge.cv2_to_imgmsg(↩
     cv2.addWeighted(dangleROI_hsv,1,cv2.addWeighted(↩
     cv2.cvtColor(
    wline,cv2.COLOR_GRAY2BGR),1,cv2.cvtColor(cv2.↩
       bitwise_and(hsv,hsv,mask = mask),
    cv2.COLOR_HSV2BGR),1,0),1,0)))

  self.slope = np.clip((np.pi/2.)-self.slope,0,np.pi↩
     /2.)
else:
  self.slope=np.nan
self.anglepub.publish(self.slope)
```

```
def rearcallbackROI(self,data):
  try:
    frame=self.bridge.imgmsg_to_cv2(data, "bgr8")#↩
       desired_encoding="passthrough")
  except CvBridgeError as e:
    print(e)

  #find the cable
  hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)

  self.rearlower=np.array([self.lower[0]-10,self.lower↩
     [1]-50,self.lower[2]-50])
  self.rearupper=np.array([self.upper[0],self.upper↩
     [1]+100,np.array([254])])#self.upper[2]+100])

  mask2=np.zeros(hsv.shape[:2], np.uint8)
  mask2[325:400,:] = 255
        azROI_hsv = cv2.bitwise_and(hsv,hsv,mask = ↩
           mask2)


  maskYellow=cv2.inRange(azROI_hsv,self.rearlower,self↩
     .rearupper)

  blur = cv2.GaussianBlur(maskYellow ,(9,9),0)
  edge=cv2.Canny(maskYellow,100,400)

  lines = cv2.HoughLines(edge, 1, np.pi / 180, 25)

  if lines is not None:
    for i in range(0,1):# len(lines)):
      rho = lines[i][0][0]
      theta = lines[i][0][1]
      a = np.cos(theta)
```

```
        b = np.sin(theta)
        x0 = a * rho
        y0 = b * rho
        pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
        pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))
        wline = cv2.line(edge, pt1, pt2, (255,0,0), 3, ↩
           cv2.LINE_AA)
      if lines[0][0][0] >= 0:
        self.azimuth = lines[0][0][1]
      else :
        self.azimuth = lines[0][0][1]-np.pi

      self.cableazpub.publish(self.bridge.cv2_to_imgmsg(↩
         cv2.addWeighted(azROI_hsv,1,cv2.addWeighted(cv2.↩
         cvtColor(
        edge,cv2.COLOR_GRAY2BGR),1,cv2.cvtColor(hsv,
        cv2.COLOR_HSV2BGR),1,0),1,0)))
    else:
      self.azimuth=np.nan
      self.cableazpub.publish(self.bridge.cv2_to_imgmsg(↩
         cv2.addWeighted(azROI_hsv,1,cv2.addWeighted(cv2.↩
         cvtColor(
        edge,cv2.COLOR_GRAY2BGR),1,cv2.cvtColor(hsv,
        cv2.COLOR_HSV2BGR),1,0),1,0)))

    self.azimuthpub.publish(self.azimuth)


  def cableTransformer(self):
    if not np.isnan(self.slope) and not np.isnan(self.↩
       azimuth):
      D = np.arcsinh(np.tan(self.slope))*(np.divide↩
         (2.57,1))  # theta = tan^(-1)(sinh(mu_c/T_0))
```

```
        self.br.sendTransform((D*np.cos(self.azimuth), D*-↩
           np.sin(self.azimuth), -.673),
               tf.transformations.quaternion_from_euler(0, ↩
                  0, -self.azimuth),
                            rospy.Time.now(),
               "cable",     # child
               "fairlead"      # parent
               )


if __name__ == '__main__':
  try:
    cableSegmenter()
  except rospy.ROSInterruptException:
    pass
```

B.2 Cable Control

```
#!/usr/bin/env python

import numpy as np
import xlsxwriter
from datetime import datetime
import rospy
from std_msgs.msg import Float32
from tf.transformations import euler_from_quaternion
from teb_local_planner.msg import FeedbackMsg, ↩
   TrajectoryMsg, TrajectoryPointMsg
from nav_msgs.msg import Odometry
from geometry_msgs.msg import Twist
```

```
class PID():
  """PID (P,I,D,IUpper,ILower) control class with anti-windup limits"""
  def __init__(self,P,I,D,ILower,IUpper,Lband,Uband):
    self.integralE = 0.0
    self.prevErrorE = 0.0
    self.kp = float(P)
    self.ki = float(I)
    self.kd = float(D)
    self.IU = float(IUpper)
    self.IL = float(ILower)
    self.LDB = float(Lband)
    self.UDB = float(Uband)

  def update(self,measured,target):
    ## return feedback update
    error = float(target) - float(measured)
    if error < self.UDB and error > self.LDB:
      error = 0.0
      self.integralE = 0.0
#     print("deadband")
    derivative = error - self.prevErrorE
    self.prevErrorE = error
    self.integralE = np.clip(self.integralE + error,self.IL,self.IU)
#   print(self.integralE)
    return self.kp * error + self.ki * self.integralE + self.kd * derivative


class ctlwb():
  """Logging for control signal plotting"""
```

```
def __init__(self):
  wbname = "/home/administrator/controllerlogs/" + str(datetime.now()) + ".xlsx"
  print wbname
  try:
    self.workbook = xlsxwriter.Workbook(wbname)
    self.worksheet = self.workbook.add_worksheet()
  except Exception as e:
    print(repr(e))
  self.row=0
  self.timecol=0
  self.ekfcolx = 1
  self.ekfcoly = 2
  self.ekfcoltheta = 3
  self.ffcol = 4
  self.fbcol = 5
  self.Danglecol = 6
  self.Dazimuthcol = 7
  self.Vlincol = 8
  self.Vangcol = 9

  self.x = None
  self.y = None
  self.theta = None

  try:
    self.worksheet.write(self.row,self.timecol,"time")
    self.worksheet.write(self.row,self.ekfcolx,"ekf x")
    self.worksheet.write(self.row,self.ekfcoly,"ekf y")
    self.worksheet.write(self.row,self.ekfcoltheta,"ekf theta")
    self.worksheet.write(self.row,self.ffcol,"FF")
```

```
      self.worksheet.write(self.row,self.fbcol,"FB")
      self.worksheet.write(self.row,self.Danglecol,"↩
         Dangle")
      self.worksheet.write(self.row,self.Dazimuthcol,"↩
         Azimuth")
      self.worksheet.write(self.row,self.Vlincol,"Vlin")
      self.worksheet.write(self.row,self.Vangcol,"Vang")
    except Exception as e:
      print(repr(e))
    self.row += 1

    rospy.Subscriber('odometry/filtered',Odometry,self.↩
       ekfback)

  def ekfback(self,data):
    self.x=data.pose.pose.position.x
    self.y=data.pose.pose.position.y
    self.theta = euler_from_quaternion([data.pose.pose.↩
       orientation.x,data.pose.pose.orientation.y,
      data.pose.pose.orientation.z,data.pose.pose.↩
         orientation.w],axes='sxyz')[2]

  def log(self,FF,FB,Dangle,azimuth,Vlin,Vang):
    if np.isnan(Dangle):
      Dangle="nan"
    if np.isnan(azimuth):
      asimuth="nan"
    try:
      self.worksheet.write(self.row,self.timecol,rospy.↩
         get_time())

      self.worksheet.write(self.row,self.ekfcolx,self.x)
      self.worksheet.write(self.row,self.ekfcoly,self.y)
```

```
        self.worksheet.write(self.row,self.ekfcoltheta,↩
           self.theta)

        self.worksheet.write(self.row,self.ffcol,FF)
        self.worksheet.write(self.row,self.fbcol,FB)
        self.worksheet.write(self.row,self.Danglecol,↩
           Dangle)
        self.worksheet.write(self.row,self.Dazimuthcol,↩
           azimuth)
        self.worksheet.write(self.row,self.Vlincol,Vlin)
        self.worksheet.write(self.row,self.Vangcol,Vang)

      except Exception as e:
        print(repr(e))
      self.row += 1


class cableController():
  """cableController outputs velocity goals for the ↩
     stepper driver"""
  def __init__(self):

    self.hz = 15

    self.rate=rospy.Rate(self.hz)

    #publisher for cable extrude velocity goals
    self.distancepub = rospy.Publisher('spool/distance',↩
       Float32, queue_size=1)

    twist_sub = rospy.Subscriber('cmd_vel',Twist, self.↩
       twistback)
```

```
        angle_sub = rospy.Subscriber('spool/angle',Float32, ↩
            self.angleback)
        azimuth_sub = rospy.Subscriber('spool/azimuth',↩
            Float32, self.azimuthback)

        self.angle = 0.0
        self.azimuth = 0.0
        self.path = []  #array of trajectory points with ↩
            time_from_start
        self.Vlin = 0.0
        self.Vang = 0.0

        self.angleSP = np.pi*55./180   #theory suggests 73 ↩
            degrees from horizontal, or 64 as limit

        self.anglePID = PID(0.75,0.1,0.0,-20.,10.,0.,0.08)

        self.stopflag = 1
        self.steve=ctlwb()



    def run(self):
        if self.stopflag == True:
            try:
                FF = self.feedForward()
                FB = self.feedBack()
                self.distancepub.publish(-(FF+FB))  #the ↩
                    direction in the driver is backwards...
#               print(D)

                self.steve.log(FF,FB,self.angle,self.azimuth,↩
                    self.Vlin,self.Vang)
```

```
    except KeyboardInterrupt:
      pass

def stop(self):
  self.distancepub.publish(0.0) #stop
  self.steve.workbook.close()
  self.rate.sleep()
  self.stopflag = 0
  print("Spool STOPPED")

def feedForward(self):
  fudge = 0.2
  if np.isnan(self.azimuth):
    return (2+fudge)*self.Vlin
  else:
    return  2*((1+fudge)*self.Vlin+np.pi*0.36*self.↩
       Vang*np.sign(self.azimuth))

def feedBack(self):
  if np.isnan(self.angle) or self.angle == 0.0 or self↩
     .angle == 1.5708:
    return 0
  else:
    return self.anglePID.update(self.angle,self.↩
       angleSP)

def angleback(self,data):
  self.angle = data.data
  self.run()

def azimuthback(self,data):
  self.azimuth = data.data

def twistback(self,data):
```

```
    self.Vlin = data.linear.x
    self.Vang = data.angular.z


if __name__ == '__main__':
  try:
    rospy.init_node('cableFeedbackControl', ↩
       disable_signals=True, anonymous=True)
    bob = cableController()
    bob.run()
  except Exception as e:
    print(repr(e))
```

B.3 Low-Level Micro-Controller

```
/*
 * Autonomous Mobile Microgrid cable drive stepper ↩
    control
 */

#if (ARDUINO >= 100)
#include <Arduino.h>
#else
#include <WProgram.h>
#endif

#include <ros.h>
#include <geometry_msgs/Twist.h>
#include <std_msgs/Float32.h>
#include <avr/io.h>
```

```
#include <AccelStepper.h>                    //using accel ↩
   stepper library, more info at https://www.airspayce.↩
   com/mikem/arduino/AccelStepper/

#define RPWM_Output 10                        //linear ↩
   actuator PWM pins
#define LPWM_Output 11
#define actuatorPWM 255                       //linear ↩
   actuator PWM speed

boolean debug = 0;

float mpstosps = 400.0/(3.1415*0.052);      // meters/s ↩
   to steps/s for 400 steps/rev and drive diameter 52mm
AccelStepper Spool(1, 2, 4);                // initiates ↩
   accel stepper pins 2 and 4 with a stepper driver

float UGVspeed = 0.0;                         //chassis ↩
   speed from vehicle
float UGVangle = 0.785;                        //cable ↩
   angle of the dangle from vehicle
float CableDistance = 0;                     //distance ↩
   goal for cable
float eint = 0;                                 //↩
   integrator
float elast = 0;                                  //e(k↩
   -1)

int ActuatorDistance = 0;
float eintA = 0;                             //for Actuator↩
    PID
float elastA = 0;
```

```
ros::NodeHandle  nh;

/*void servo_cb( const geometry_msgs::Twist& cmd_msg){
 //  digitalWrite(13,HIGH);
 UGVspeed = cmd_msg.linear.x+1.65*abs(cmd_msg.angular.z);      //scalar speed of cable fairlead
 }

 void angle_cb( const std_msgs::Float32& angle_msg){
 digitalWrite(13,HIGH);
 UGVangle = angle_msg.data;      //cable angle of the dangle
 }
 */
void distance_cb( const std_msgs::Float32& distance_msg){
  digitalWrite(13,HIGH);
  CableDistance = distance_msg.data;//*mpstosps;      //cable distance next time interval
}

void actuator_cb( const std_msgs::Float32& actuator_msg){
  digitalWrite(13,HIGH);
  ActuatorDistance = constrain(actuator_msg.data*1000,0,305);      //convert m extension to mm for linear actuator
}

//ros::Subscriber<geometry_msgs::Twist> sub("spool/cmd_vel", servo_cb);
//ros::Subscriber<std_msgs::Float32> suba("spool/angle", angle_cb);
```

```
ros::Subscriber<std_msgs::Float32> subd("spool/distance"↩
   , distance_cb);
ros::Subscriber<std_msgs::Float32> subLD("actuator/↩
   distance", actuator_cb);



void setup(){
  pinMode(13, OUTPUT);               //indicator light
  pinMode(7, INPUT_PULLUP);           //button switches ↩
     manual control from spool to actuator
  pinMode(RPWM_Output, OUTPUT);
  pinMode(LPWM_Output, OUTPUT);

  if (debug){
    Serial.begin(9600);
  }
  else{
    nh.initNode();
    //  nh.subscribe(sub);
    //  nh.subscribe(suba);
    nh.subscribe(subd);
    nh.subscribe(subLD);
  }

  Spool.setPinsInverted(0,0,0);
  Spool.setMinPulseWidth(3);         //DM542T requires at↩
      least 2.5 us pulse
  Spool.setAcceleration(300);                    //sets ↩
     max rate speed can change steps/s/s
  Spool.setMaxSpeed(4000);//2450);     // sets max motor↩
      speed
}
```

```
void loop(){
  if(!debug){
    nh.spinOnce();
  }

  boolean but = digitalRead(7);

  /*  if (UGVspeed<0.0001 && UGVspeed>-0.0001){
   setManualVelocity();
   }
   else{
   angleFeedback();
   //    cableVelocity(UGVspeed);
   }*/
  if (CableDistance==0.0 && but){
    setManualVelocity();
  }
  else if (!but){
    setManualExtension();
  }
  else {
//    Spool.moveTo(CableDistance);
    cableVelocity(CableDistance);
    actuatorFeedback(ActuatorDistance);
  }

  Spool.run();

  if (debug){
    Serial.print("   current speed ");
    Serial.print(Spool.speed());
    Serial.println();
```

```
  }
  //  delay(1);
}



void cableVelocity(float mps){
  int theFast = mps*mpstosps;
  if (debug){
    Serial.print("spool sps ");
    Serial.print(theFast);
  }
  Spool.setSpeed(theFast);                // sets speed of↩
      motor to desired speed
}

void setManualVelocity(){
  digitalWrite(13,LOW);
  int sensorValue = analogRead(A4);      //spool speed ↩
     pot on this pin
  float mps = (constrain(map(sensorValue,0,450,-1000,0)↩
     ,-1000,0) +
    constrain(map(sensorValue,573,1023,0,1000),0,1000))↩
      /1000.0;  //pot sets -1 to 1 m/s speed, with ↩
      deadband
  cableVelocity(mps);
}

void angleFeedback(){
  int windup = 10;                                //↩
     integrator anti-windup
  int mwindup = -10;
  float angleSP = 0.785;                           //angle ↩
     SetPoint
```

```
  float P = 0.5;              //Proportional Gain
  float I = 0.1;                                  //Integral↩
      Gain
  float D = 0.0;                                  //↩
     Derivative Gain

  float e = UGVangle - angleSP;                        //↩
     dangle error
  eint = e + eint;                                //↩
     integrate error
  eint = constrain(eint,mwindup,windup);     //anti-↩
     windup
  float G = P*e + I*eint + D*(e - elast);     //calculate↩
      PID on cable angle
  elast=e;                                     //record ↩
     last error

  float fbSpeed = UGVspeed + G;               //sum angle↩
      gain with feedforward speed

  cableVelocity(fbSpeed);
}

void extend(){
  if (debug){
    Serial.println("Extending!");
  }
  analogWrite(LPWM_Output, 0);
  analogWrite(RPWM_Output, actuatorPWM);
  delay(1000);
}

void retract(){
  if (debug){
```

```
    Serial.println("retracting...");
  }
  analogWrite(RPWM_Output, 0);
  analogWrite(LPWM_Output, actuatorPWM);
  delay(1000);
}

void stopActuator(){
  if (debug){
    Serial.println("STOPPING!!");
  }
  analogWrite(RPWM_Output, 0);
  analogWrite(LPWM_Output, 0);
}

void actuatorFeedback(int setP){    //input SP in ↩
   integer mm
  int actuatorValue = analogRead(A3);
  int FB = map(actuatorValue,1018,101,0,305);  //pot ↩
     sets full retract to full extend, sp in mm

  int windup = 25;                              //↩
     integrator anti-windup
  int mwindup = -25;
  float P = 100;             //Proportional Gain
  float I = 0.1;                                 //Integral↩
      Gain
  float D = 0;                                  //Derivative↩
      Gain

  float e = setP-FB;                      // error
  eintA = e + eintA;                             //↩
     integrate error
```

```
  eintA = constrain(eintA,mwindup,windup);     //anti-↩
     windup
  float G = P*e + I*eintA + D*(e - elastA);     //↩
     calculate PID
  if (abs(e)<=1){
    G=0;
    if(debug){
      Serial.println("Deadband");
    }
  }
  elastA=e;                                     //record ↩
     last error

  int rG = constrain(G,0,255);
  int lG = abs(constrain(G,-255,0));
  analogWrite(RPWM_Output, rG);
  analogWrite(LPWM_Output, lG);

  if (debug){
    Serial.print(" actuator ");
    Serial.print(actuatorValue);
    Serial.print(" FB ");
    Serial.print(FB);
    Serial.print(" SP ");
    Serial.print(setP);
    Serial.print(" G ");
    Serial.print(G);
    Serial.print(" rPWM ");
    Serial.print(rG);
    Serial.print(" lPWM ");
    Serial.print(lG);
  }
}
```

```
void setManualExtension(){
  digitalWrite(13,HIGH);
  int sensorValue = analogRead(A4);      //spool speed ↩
     pot on this pin
  int mm = map(sensorValue,0,1023,0,305);  //pot sets ↩
     full retract to full extend, sp in mm
  digitalWrite(13,LOW);
  actuatorFeedback(mm);
}
```

Appendix C

Letters of Permission

IEEE:

Authors may include their articles in a thesis or dissertation.

Include the following copyright notice in the references: "© 20XX IEEE. Reprinted, with permission, from [full citation of original published article]."

When posting your thesis on your university website, include the following message:

"In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [name of university or educational entity]'s products or services. Internal or personal use of this material is permitted. If interested

in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation."

Only the accepted version of your article, not the final published version, may be posted online in your thesis.

Elsevier:

Author rights:

The below table explains the rights that authors have when they publish with Elsevier, for authors who choose to publish either open access or subscription. These apply to the corresponding author and all co-authors.

Use and share their works for scholarly purposes (with full acknowledgement of the original article):

5. Include in a thesis or dissertation (provided this is not published commercially)

Institution rights:

Regardless of how the author chooses to publish with Elsevier, their institution has the right to use articles for classroom teaching and internal training. Articles can be used for these purposes throughout the author's institution, not just by the author:

Theses and dissertations which contain embedded final published articles as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publication on ScienceDirect

From: https://www.elsevier.com/about/policies/copyright/permissions

Can I include/use my article in my thesis/dissertation?

Yes. Authors can include their articles in full or in part in a thesis or dissertation for non-commercial purposes.

Can I use material from my Elsevier journal article within my thesis/dissertation?

As an Elsevier journal author, you have the right to Include the article in a thesis or dissertation (provided that this is not to be published commercially) whether in full or in part, subject to proper acknowledgment; see the Copyright page for more information. No written permission from Elsevier is necessary.

This right extends to the posting of your thesis to your university's repository provided that if you include the published journal article, it is embedded in your thesis and

not separately downloadable.