Towards Autonomy and Mobility for a Tethered Robot Exploring Extremely Steep Terrain

by

Patrick Michael McGarey

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto

Abstract

Towards Autonomy and Mobility for a Tethered Robot
Exploring Extremely Steep Terrain

Patrick Michael McGarey
Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto
2017

Mobile robots are well suited to explore environments considered too costly, time consuming, and hazardous for human inspection. However, a recent push to explore increasingly extreme environments has required the development of robust mobile platforms that can navigate steep, cluttered terrain, operate for extended periods, and relay information to a remote operator. Applications of these systems include terrestrial and planetary geologic survey and infrastructure inspection, where remote observation is not a viable option. This thesis chronicles the design, development, and testing of the physical platform and autonomy functions for the Tethered Robotic Explorer (TReX), a novel mapping robot capable of navigating near-vertical terrain while supported by an attached electromechanical tether; the tether provides continuous power and communication to and from the robot, but also constrains motion due to its finite length and susceptibility to entanglement in cluttered environments. A tethered robot can avoid entanglement by (i) mapping intermediate anchors (locations of obstacle-to-tether contacts), and (ii) autonomously retracing its outgoing path to sequentially unwrap the tether from obstacles. We approach (i) by formulating incremental and batch solutions to a new tethered simultaneous localization and mapping problem using tether measurements to aid odometry and map anchors, and handle (ii) using visual route following in conjunction with autonomous tether control to manage tension and assist the robot to repeat previously driven paths on steep terrain. This work concludes with a geologic surveying mission, where TReX is teleoperated to explore and map a steep, tree-covered rock outcrop in an outdoor mine.

Acknowledgements

"It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way" - Charles Dickens (1859)

This excerpt from *A Tale of Two Cities* best captures my experience as a graduate student. From formulating the initial scope of my thesis, struggling through implementation, and finally, demonstrating these ideas in the field, there have been many victories and defeats along the way. The sum total of these experiences, both professional and personal, have indelibly shaped me into a more thoughtful and effective student, researcher, and man. To that end, I want to express my gratitude and thanks to the many people who have helped me along the way. To my advisor, Tim Barfoot, I want to not only thank you for helping me to define the ideas presented in this thesis, but more so, for instilling in me the ability to be my own filter, effectively and concisely communicate my ideas, and of course, to do excellent research. I am sincerely grateful for the opportunity to work with you. To my post-doc advisor, François Pomerleau, thank you for your tireless support, thoughtful advice, and excellent photography throughout the development, testing, and field deployment of TReX. Thank you to my coauthors, Kirk MacTavish for contributing time and ideas to make TSLAM possible, Max Polzin for developing a simple/elegant tether-control strategy, and David Yoon and Tim Tang for your help in lidar-based mapping. Thanks to the rest of the ASRL team, Mike [0], Braden, Katarina, Jon, Chris, Sean, Peter, Mike [1], Kai, Tyler, Mona, Hengwei, and Nan for always lending a helping hand. Thank you to Fulbright Canada for your generous support of my research. Thank you to my friends, family, and sister, Kelly, for treating me like a rock-star/rocket-scientist – it really helped when I needed it most. To Michael, Tom, and Joan, your support is not forgotten, it pushes me forward every day. To my mother, Julie, a life-long cheerleader, thank you for always supporting my crazy ideas and believing in my path. To my wife and love, Tricia Roscoe, thank you for always listening to me and providing endless support and encouragement. Also, thank you for serving as the editor-in-chief on all of my papers (if you find any typos not on this page let her know). Above all, this thesis would not have come together without all of you – seriously.

Contents

List of Figures

Acronyms

TReX : Tethered Robotic Explorer.
TSLAM : Tethered Simultaneous Localization and Mapping.
SLAM : Simultaneous Localization and Mapping.
VO : Visual Odometry.
VT&R : Visual Teach & Repeat.

Notation

a	:	Symbols in this font are real scalars.
$\mathbf{a}$	:	Symbols in this font are real column vectors.
$\mathbf{A}$	:	Symbols in this font are real matrices.
$\mathcal{N}(\mathbf{a}, \mathbf{B})$	:	Normally distributed with mean $\mathbf{a}$ and covariance $\mathbf{B}$.
$\mathbf{1}$	:	The identity matrix.

Chapter 1

Introduction

1.1 Motivation

We are motivated to deploy mobile robots in the investigation of extreme environments considered too dangerous, time consuming, and costly for human exploration. Mobile robots are well suited to access hard-to-reach areas to aid in the geologic survey of cliffs, caves, and crevices, as well as for infrastructure inspection of mines, dams, and buildings. Figure 1.1 provides example applications for robotic exploration in extreme environments. Deploying robots in these environments requires the development of new approaches that extend the current capabilities and autonomy of conventional robotic systems (Schenker et al., 2003). For example, the exploration of steep cliffs on Mars, where geologic history is directly exposed, is well beyond the capacity of currently deployed rovers (Matthews and Nesnas, 2012). Therefore, in this thesis, we address the development of a system that enables a conventional wheeled robot, like those deployed on Mars, to access extremely steep terrain for the purpose of mapping. In particular, we focus on mapping applications that are not suitable for remote or aerial observation, like steep, cluttered areas occluded by structure or vegetation. Moreover, we address the development of autonomy for these systems as a building block towards the autonomous exploration of extreme environments.

A beneficial approach to the exploration of steep terrain involves tethering, which leverages an electromechanical tether attached to a rover to provide support, power, and communication while driving (Wettergreen et al., 1993). We note the important distinction between robots that use tethers as a means of power and communication alone (e.g., underwater robots with neutrally buoyant tethers), and those that additionally exploit the tether's tensile strength for support on steep terrain. Tethering allows a

Figure 1.1: *Exploring the Extreme*: From surveying exposed rock layers on Mars to study formation history, to mapping mine stopes to determine yield, to performing structural inspection, mobile robots are ideal for accessing hard-to-reach areas.

robot equipped with sensors to explore and collect data in normally inaccessible areas for extended periods of time as a result of harnessing off-board power. In comparison, 'untethered' robots (e.g., aerial vehicles), which rely on finite, on-board power, have limited time to map an environment before charging is required. Wettergreen et al. (1993) introduced the first tethered robot capable of exploring steep areas (e.g., volcanoes in Antarctica and Alaska). Huntsberger et al. (2007) and Matthews and Nesnas (2012) provide modern examples of tethered climbing robots for space exploration.

Despite the stated benefits of tethering, the principal disadvantage is that navigation is inherently more complicated; in cluttered environments the tether will come into contact with obstacles and form intermediate anchors (Sinden, 1990). Without careful consideration of anchors, the tether may become entangled and immobilize the robot. The solution to the entanglement problem is two fold: (i) detect and map anchors while driving, and (ii) ensure that the robot retraces its outgoing path in order to sequentially detach from added anchors.

1.2 Research Objectives

The central focus of this research concerns the development of a custom, tethered robot and algorithms formulated to address autonomy in extremely steep, cluttered environments for tethered robots in general. We focus on the following research objectives.

- Systems Design: Enhance tethered robot mobility.
- Tethered Autonomy: Avoid tether entanglement.
- Environment Mapping: Map extremely steep, cluttered terrain.

1.2.1 Systems Design

Drawing on the strengths and weaknesses of prior tethered systems, we have designed and built the Tethered Robotic Explorer (TReX), which is a mapping platform that is capable of managing tether on board and generating 3D scans of the environment. Tether management (reeling in/out) and rotation of an attached 2D lidar are accomplished using a single actuator mounted to a passively rotating tether deployment arm. The tether arm makes TReX unique amongst tethered systems, in that the body of the robot can rotate continuously in place while the arm passively aligns with the current anchor due to tension. This design feature makes it possible to turn on steep terrain and drive tangent to the slope in order to cover/map more area during a single traverse. The tether arm also enables the measurement of the tether's length and bearing-to-anchor, which are used to prevent tether entanglement as explained by the following objective.

1.2.2 Tethered Autonomy

We explore two solutions to the problem of tether entanglement in cluttered environments to enable tethered autonomy. (i) In order to map intermediate anchors, we formulate the first solutions to the tethered simultaneous localization and mapping (TSLAM) problem. In TSLAM, we are interested in estimating the robot's trajectory and map of anchors given nonvisual tether length, bearing-to-anchor, and wheel velocity measurements. While the setup resembles a range-bearing SLAM problem, in TSLAM we must account for the fact that tether length, which can be thought of as a pseudo range measurement, is a function of all anchors contacting the tether, which has implications on the structure of the problem. We show that an efficient solution can still be formulated, as we compare the accuracy of both incremental and batch methods with respect to ground truth using data collected from TReX. (ii) To further reduce the risk of entanglement, we enable TReX to autonomously retrace its outgoing path and sequentially detach from anchors using the idea of visual route following. Specifically, we use the well-tested Visual Teach & Repeat (VT&R) algorithm to backtrack along a set of manually taught paths in both steep and cluttered environments. VT&R relies on a path tracker, which converts localization errors into appropriate wheel actions. However, when the robot loses wheel traction and slips on steep terrain, the path tracker will fail to produce wheel actions that enable autonomous route following. To account for this, we make no modifications to the underlying VT&R algorithm, and instead use a new tether control strategy; the controller selects a slip-minimizing tension, which depends on the vehicle's inclination with respect to gravity, allowing the robot to drive as if it were untethered on moderate terrain and climb steep slopes when wheel traction is significantly reduced (i.e.,tether assistance is required).

1.2.3 Environment Mapping

Finally, we detail a mapping deployment with TReX to investigate a steep rock outcrop at an outdoor mine site in Northern Ontario, Canada. Due to the complexity of the terrain, the robot was manually piloted on a series of steep paths to map a contiguous area with exposed bedrock that spans over 150 meters and is partially covered by forest and ground vegetation. Figure 1.2 illustrates a typical mapping campaign and shows an image from the experiment. The field test served to evaluate both the mapping and system capabilities of TReX. Mapping with TReX involves a 2D lidar mounted on its tether spool, which rotates to generate a 3D scan as the robot drives and deploys tether. Since scanning requires vehicle motion, we must account for scan distortion by estimating the robot's trajectory during a single scan before merging it into the map. Using data collected from the experiment, two existing approaches to handle scan distortion are compared; (i) a continuous-time, lidar-only method that accommodates for asynchronous measurements using a constant-velocity motion prior, and (ii) a camera-aided approach that leverages visual odometry. Once scan distortion is rectified, scans are aligned into a global map of the environment using Iterative Closest Point (ICP) matching. The results include a series of point-cloud maps that are compared to a ground-truth map as well as a discussion of lessons learned from the deployment.

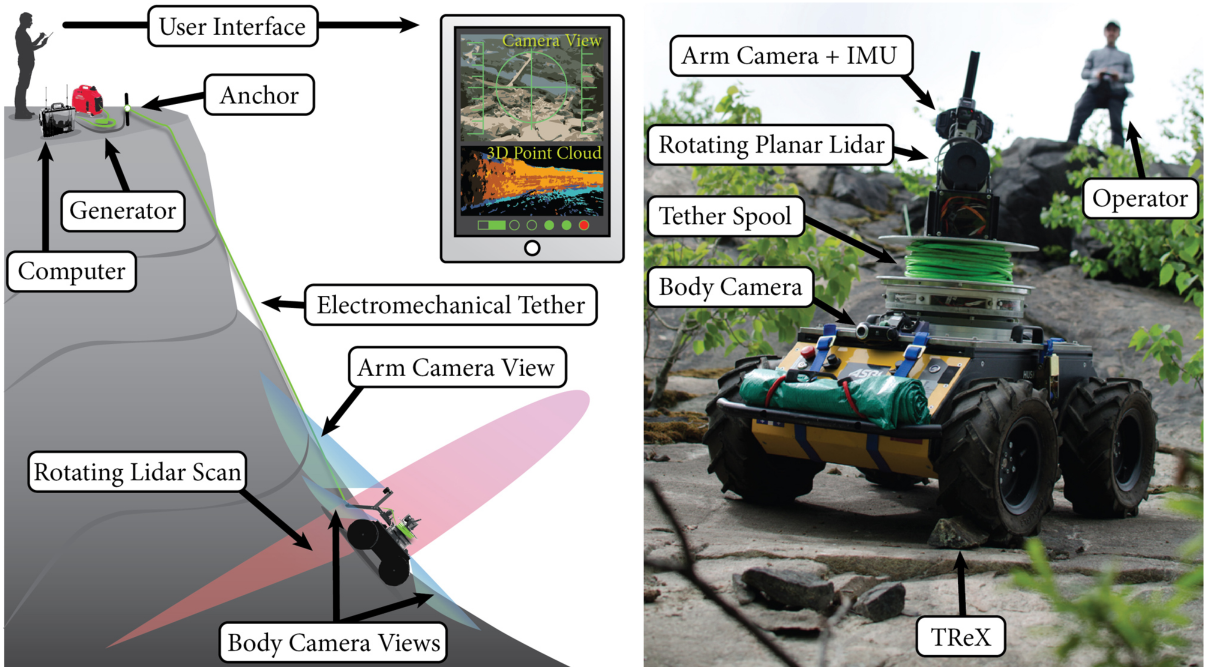


Figure 1.2: *Mission Concept*: The *left* image illustrates a geologic mapping deployment with TReX. The *right* image was taken during an actual field deployment.

1.3 Novel Contributions

This thesis makes the following novel contributions towards robotics research with an emphasis on tethered, mobile systems.

- The first tethered robot design capable of continuous rotation under tension.
- The first formulation of the TSLAM problem with incremental and batch solutions.
- The first demonstration of autonomous route following on steep, cluttered terrain.
- The deployment of TReX to map steep terrain in an outdoor environment.

1.4 Thesis Structure

This thesis is structured to address each of the aforementioned research objectives. First, Chapter 2 details the systems design and initial testing of the TReX prototype. Chapters 3 and 4 address tethered autonomy. Chapter 3 proposes incremental and batch solutions to the TSLAM problem and compares their accuracy on real data with respect to ground truth. Chapter 4 outlines our visual route following pipeline, proposes a tether control strategy, and evaluates results from experiment. Chapter 5 explains the environment mapping pipeline and presents results and lessons learned from a large-scale field test. The structure of each chapter commonly includes a motivation on the topic, a review of prior works, a description and methodology of the system and solution to be evaluated, experimental results validating the proposed method, a list of novel contributions, and references to associated publications and videos. We conclude with Chapter 6, which provides closing remarks, engineering lessons, and proposes potential avenues for future work on tethered systems.

1.5 Associated Publications

The following, first-author papers have appeared for publication and comprise the technical content of this thesis.

- McGarey et al. (2015). System Design of a Tethered Robotic eXplorer (TReX) for 3D Mapping of Steep Terrain and Harsh Environments. In the *2015 International Conference on Field and Service Robotics (FSR).*

- McGarey et al. (2016). The Line Leading the Blind: Towards Nonvisual Localization and Mapping for Tethered Mobile Robots. In the *2016 IEEE International Conference on Robotics and Automation (ICRA).*

- McGarey et al. (2017a). TSLAM: Tethered Simultaneous Localization and Mapping for Mobile Robots. In the *International Journal of Robotics Research (IJRR).*

- McGarey et al. (2017b). Falling in line: Visual Route Following on Extreme Terrain for a Tethered Mobile Robot. In the *2017 IEEE International Conference on Robotics and Automation (ICRA).*

- McGarey et al. (2017c). Field Deployment of the Tethered Robotic eXplorer to Map Extremely Steep Terrain. In the *2017 International Conference on Field and Service Robotics (FSR).*

1.6 Associated Videos

- Intro to TReX: https://youtu.be/Q2gOOhK45lY
- Mechanical Design: https://youtu.be/iQYULj8TLWk
- Building TReX: https://youtu.be/i7e7iHxMmu0
- TSLAM (Particle Filter): https://youtu.be/7ehPxdtYWrA
- TSLAM (Batch Method): https://youtu.be/mzlHJEa3z3Y
- Tethered VT&R: https://youtu.be/qqIkfSabtZs
- TReX in the Field: https://youtu.be/VakpChosVNE
- Mapping Extreme Terrain: https://youtu.be/9r10kC7GTmc

Chapter 2

System Design of TReX

2.1 Motivation

The exploration of steep terrain by mobile robots has been a topic of interest for decades. Prior to the wide-spread adoption of light-weight, energy-dense batteries, many conventional ground robots remained tethered for powering purposes. Even today, due to the limitations of on-board battery storage, roboticists opt to use external power (e.g., an extension cord connecting a robot to wall power) for extended laboratory experiments. Tethering, via an electromechanical tether, is not only ideal for leveraging external power, but also for reliably transmitting data and providing mechanical support on extreme terrain. However, outside of underwater applications where tethering is standard, tethers are not widely used for ground robots because tether management is an unsolved problem in challenging environments (Nagatani et al., 2013). Sinden (1990) first introduced the need for tether management in complex environments as a means to prevent entanglement. Since then, a variety of tethered robots have been developed and tested on steep terrain, but few have addressed tether management as a general problem for mobile robots. Furthermore, a lack of demonstrated autonomy and advanced mobility on steep terrain has slowed both interest and progress in tethered systems. In response, we have developed a new tethered platform specifically to investigate tether management, autonomy, and most importantly, mapping on extremely steep terrain. The Tethered Robotic Explorer (TReX), as seen in Figure 2.1, is the first tethered robot capable of lateral motion on steep terrain while under tension. Additionally, TReX is equipped with sensors that enable tethered autonomy and 3D mapping (see Chapters 3, 4, and 5). In this chapter, we introduce the systems design for TReX, drawing comparisons to prior

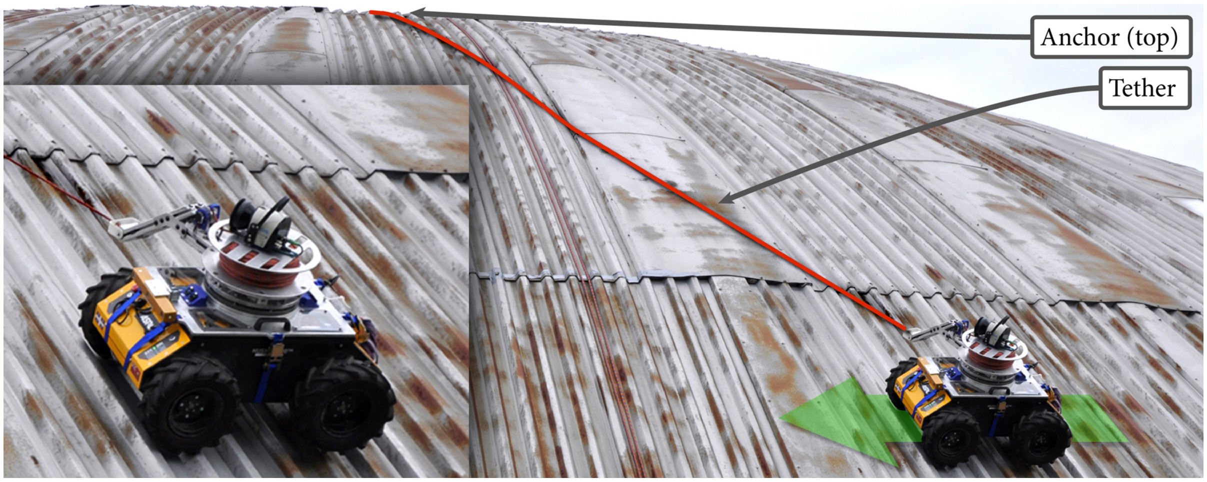


Figure 2.1: *Tethered Robotic Explorer*: TReX traverses the exterior of a dome, demonstrating lateral motion under tension. The heading is shown by a green arrow.

approaches, as well as present results from evaluation and verification of the system.

This chapter proceeds as follows. Section 2.2 presents a brief history of tethered, climbing robots. Section 2.3 details our design and fabrication approach for TReX. Section 2.4 provides results from system verification and calibration tests. Section 2.5 states novel contributions. Concluding thoughts and future extensions for the TReX platform are available in Chapter 6.

2.2 Related Work

Dante I and II (shown in Figure 2.2) were the first tethered climbing systems to be designed and deployed in extreme terrain (Wettergreen et al., 1993). Their unique, eight-legged 'walking' configuration allowed for traversing steep, snow-covered volcano craters and, through testing, demonstrated the challenges of tethered mobility for the first time. During several field deployments, issues related to mobility under tension and tether management resulted in extensive damage to the platform. Following Dante, The Teamed Robots for Exploration and Science on Steep Areas (TRESSA) project outfitted a conventional, wheeled rover with a pair of tethers in order to explore steep slopes (Huntsberger et al., 2007). The dual-tether system provided additional stability, yet made tether management inherently more complex. In fact, tether deployment was coordinated by two separate anchor robots located at the top of a cliff. Additionally, by not managing tether on board, the cables were exposed to abrasion due to dragging,

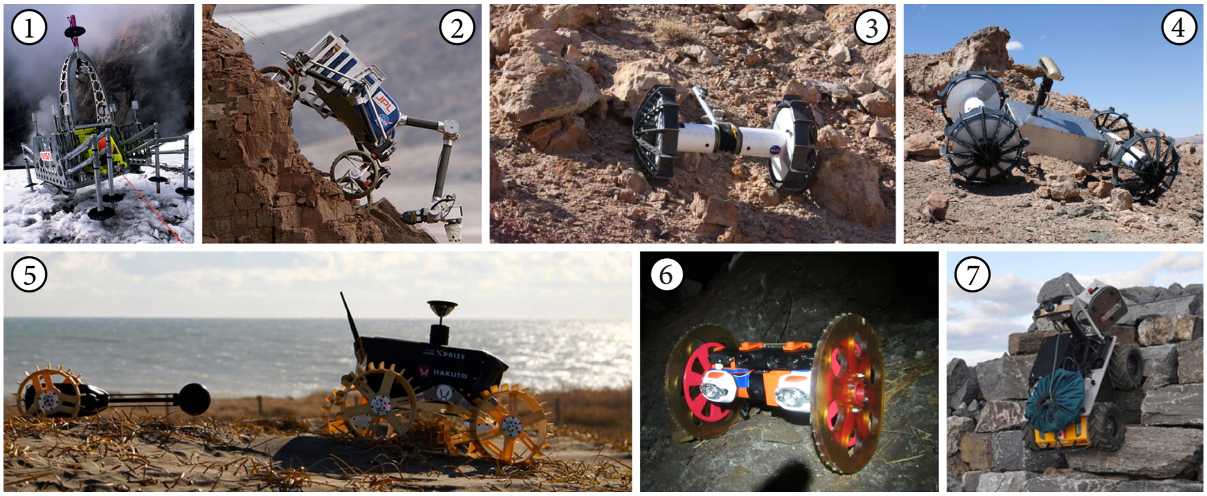


Figure 2.2: *Review of Tethered Robots*: (1) Dante II (Bares and Wettergreen, 1999), (2) TRESSA (Huntsberger et al., 2007), (3) Axel II and (4) DuAxel (Matthews and Nesnas, 2012), (5) Tetris and Moonraker (Britton et al., 2015), (6) VolcanoBot (JPL/CalTech), and (7) vScout (Stenning et al., 2015)

which in turn, decreased the effective range of the robot. More recently, Matthews and Nesnas (2012) developed the Axel robot, which returns to the idea of on-board tether management and uses a novel dual-wheel, self-righting design as shown in image (3) of Figure 2.2. Image (4) shows a variation of the design, which uses two Axel robots to construct a redundant four-wheeled platform, where one of the Axel robots serves as the 'anchor' while the other descends to explore. Axel's design has inspired the Moonraker and Tetris robots (a lunar rover concept from Tohoku University), and VolcanoBot (a volcanic vent mapping prototype from JPL/Caltech), which are shown in images (5-6). The key issue with Axel is that the potential for embedding mapping sensors is limited by design; the current system has a forward-looking stereo camera centered between the wheels with limited space for additional sensors. With the idea that conventional rover platforms are better equipped for sensor integration, the vScout prototype was developed as a precursor to TReX (Stenning et al., 2015). The prototype, shown in image (7), was capable of reeling, but not managing, its tether. The motivation for the TReX design is meant to address a common limitation of all prior systems, which is a lack of advanced mobility on steep terrain; no prior system has demonstrated the capability of turning significantly outside the direction of applied tension to drive laterally with respect to the slope. This disadvantage implies a greater risk of tether entanglement when navigating around obstacles, and also limits the robot to drive in a straight line away from its anchor, which decreases mapping efficiency.

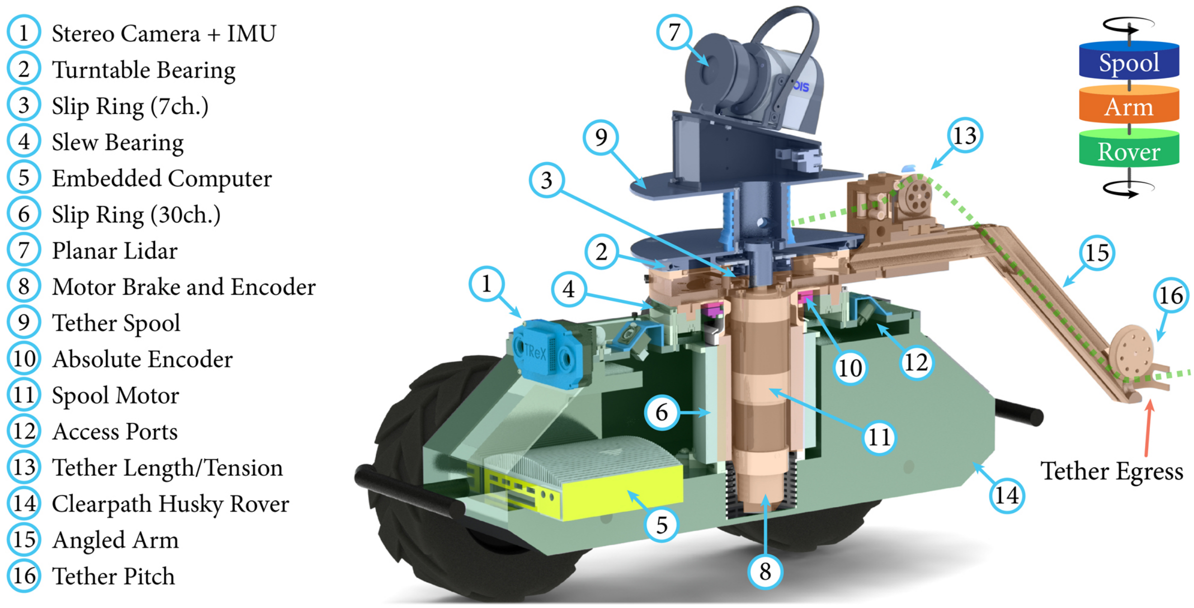


Figure 2.3: *TReX Cut View*: The rotating elements of TReX are highlighted by color. The tether arm (orange) rotates passively in the direction of applied tension. The spool (blue) is actuated with respect to the tether arm. The result is that the rover (green) can rotate in place on steep slopes in order to drive laterally on the steep terrain.

2.3 Design Methodology

2.3.1 Continuous Rotation

In order to navigate laterally on steep slopes, we have developed a tethered robot design that allows for continuous rotation while under tension. The design leverages a tether management payload that attaches to a conventional, wheeled rover and passively rotates about its center (yaw) axis. The payload is mounted to a Clearpath Husky A200 rover, which is a four-wheeled, skid-steered robot base. The tether, which terminates at the rotational center of the robot and tether management payload, is deployed through a tether arm that passively rotates in the direction of applied tension. The tether arm is also angled at the tether's egress point, which has the effect of providing stability on steep terrain by aligning[1] the tensional force with a virtual line passing through the vehicle's center-of-mass. The tether is reeled in and out using a motorized spool, which is not coupled in any way to rover rotation. To visualize the complex rotational freedom of the TReX design, Figure 2.3 provides an annotated cut view with color-coded, rotating elements highlighted. To accomplish the rotation, we have mounted the tether

[1]The tether arm length can be adjusted, i.e., 'trimmed', for better alignment.

management payload on a slew bearing. Power and data are transmitted between the rover base and payload using a multi-channel slip ring with a hollow center. The spool motor is conveniently suspended within the hollow center of the slip ring and is fixed only to the base of the tether arm. Power is transmitted through the slip ring and into the motor in order to rotate a shaft that is coupled to the tether spool. The spool rotates on a separate turntable bearing and is electrically linked to the tether arm by a second, hollow-center slip ring, which allows the motor shaft to pass through. An electronics compartment at the top of the spool distributes power to the top-mounted lidar (light detection and ranging sensor), and is the connection point for the tether. The tether passes AC power and Ethernet data into the electronics compartment. The AC power is used to charge an on-board battery, while the Ethernet signal is linked to an embedded computer located in the rover base. The end result is that TReX can rotate continuously on steep terrain, provided sufficient wheel traction, while the tether spool rotates to manage tension and generate 3D scans using the attached 2D lidar. For more design detail, see the supplemental figures provided in Appendix A.

2.3.2 Comparison to Other Systems

Figure 2.4 compares the mobility of TReX with past systems. TReX's ability to rotate continuously under tension on steep terrain is beneficial for obstacle navigation and efficient mapping. Each platform is evaluated by the following metrics.

- *Rotational Freedom*: Prior systems do not rotate significantly outside the direction of applied tension. TReX has a passively rotating tether arm that allows rotational freedom rotation on steep terrain, provided sufficient wheel traction.
- *Passing Obstacles*: Tether-to-obstacle contacts create intermediate anchors, which must be removed in order for the robot to return safely. TReX has the ability to rotate and drive laterally around an obstacle instead of surmounting it.
- *Climbing Obstacles*: Despite the advanced rotational ability of TReX, there are still challenges to navigating extreme terrain. While Dante and Axel are tailored for navigating over large obstacles, TReX's size and limited ground clearance can cause the robot to get stuck if no alternative path exists around a difficult obstacle.
- *Coverage Area*: The ability to drive laterally on steep terrain allows TReX to sweep back and forth in order to cover more area in a single traverse. Other systems are limited to travel mostly linear paths, which means that the anchor must be relocated in order to cover and map new areas.

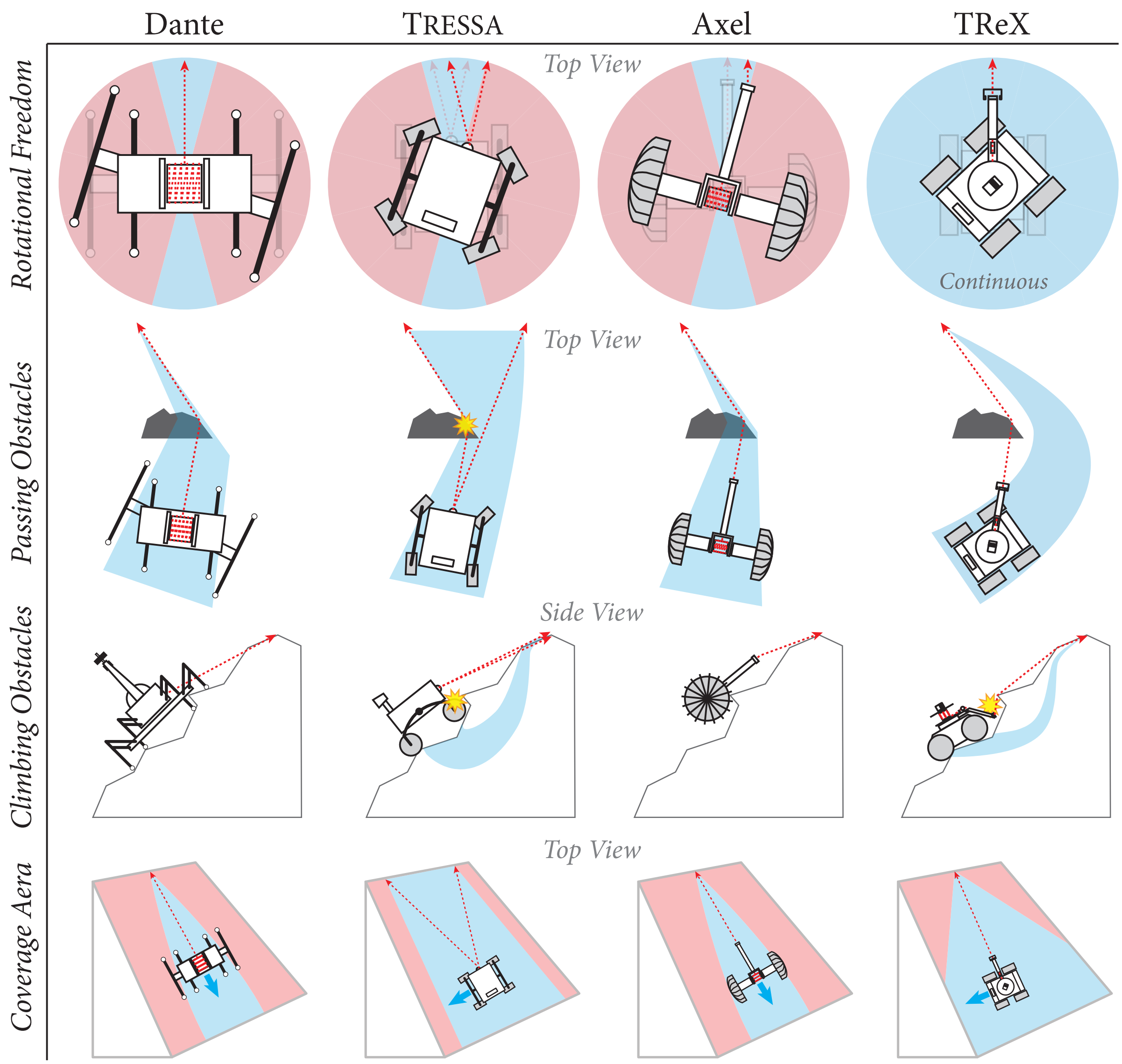


Figure 2.4: *Mobility Comparison to Other Systems*: The mobility of Dante, TRESSA, Axel, and TReX are qualitatively compared in the illustrations above. We note that these illustrations are not informed by prior research, and instead, are created by looking at the design of past systems to make a cartoon mobility comparison. Rows represent mobility metrics and columns correspond to tethered systems. All vehicles (except TRESSA) manage tether on board. Tethers are dashed red lines and tether-to-obstacle contacts are yellow stars. The light blue and red shaded regions represent feasible and infeasible rotations/paths respectively. Blue arrows indicate vehicle heading. Overall, the TReX design enables continuous rotation under tension on steep terrain, which improves obstacle navigation, increases coverage area, and makes mapping more efficient.

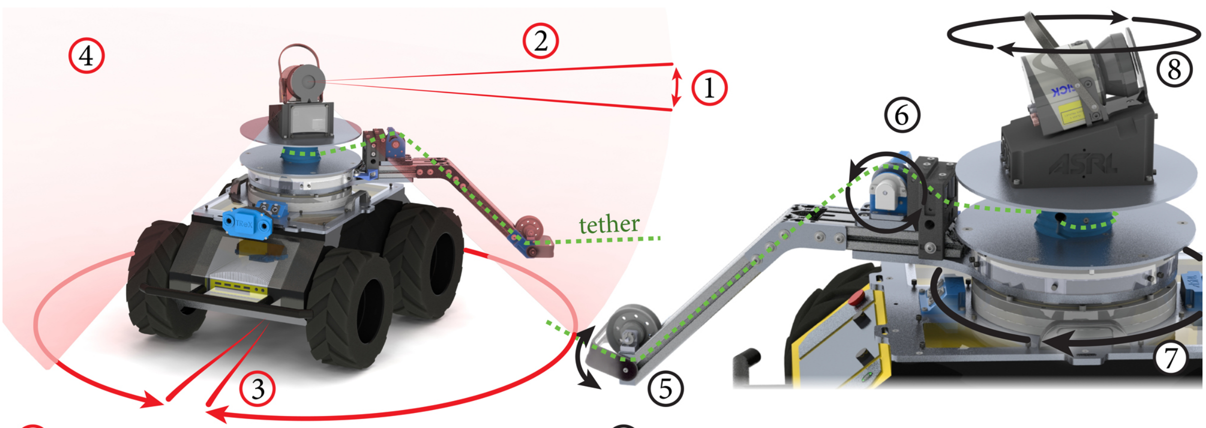


Figure 2.5: *TReX Sensor Configuration*: The *left* image provides 3D mapping specifications, while the *right* indicates the tether and payload orientation sensor arrangement.

2.3.3 Sensor Configuration

The sensor configuration of the TReX platform is illustrated in Figure 2.5. TReX has the ability to make a complete 3D scan of the environment through the rotation of an attached 2D lidar. Due to the single-actuator design of the tether management payload, 3D scanning is dependent on the deployment of tether through vehicle motion. During lidar rotation, the worst-case, azimuth scan spacing [2] is 1.7°, which is a function of the maximum spool rate (0.23 revolutions/second) and the lidar's scan frequency (50 Hz). The upward-facing orientation of the lidar allows for scanning enclosed environments, like caves and crevices, but also works for planar surfaces, like cliffs and dam walls. Figure 2.5 also illustrates how measurements of the payload's orientation and attached tether are made. To create accurate maps of the environment, we need to know the orientation of the rotating lidar with respect to the robot base at all times. To accomplish this, we use a series of encoders to measure (i) the angle of the tether arm with respect to the robot base, and (ii) the angle of the spool with respect to the tether arm. We also use (i) as a tether bearing-to-anchor measurement in conjunction with an encoder that measures tether length to aid in localizing obstacle-to-tether contacts (see Chapter 3). Tether pitch is measured at the egress of the tether arm, and offers a basic measurement for tether

[2] In spherical coordinates, the worst-case azimuth scan spacing is the angle between a pair of points observed between two consecutive scans at the same elevation or polar angle.

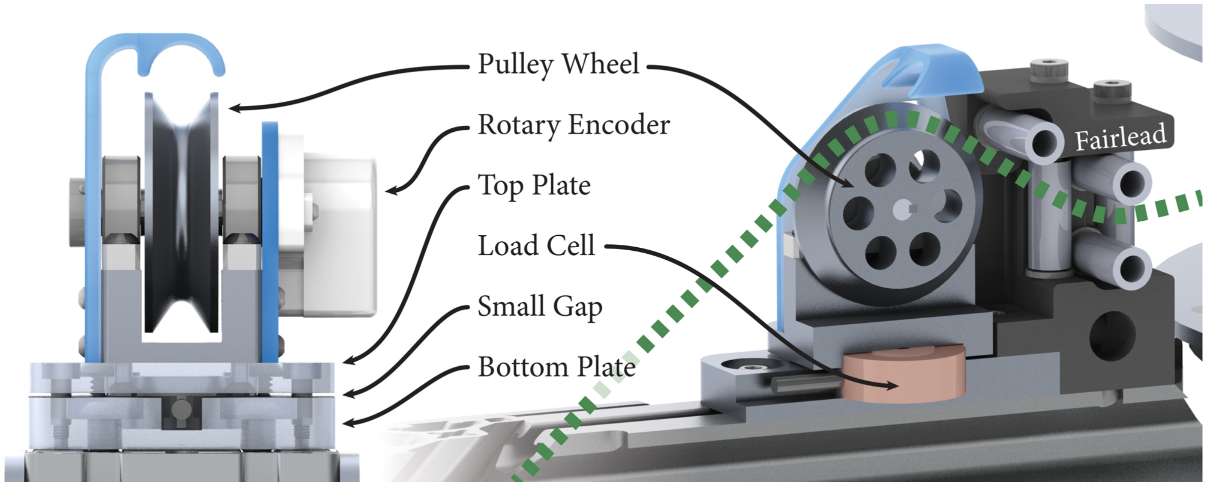


Figure 2.6: *Tether Length & Tension Sensor*: The pulley mechanism has an encoder for measuring length, and a piezoelectric load cell for sensing force due to tension.

tension. To accurately measure tension for the purpose of automatic tension control, we use a piezoelectric sensor integrated into a pulley mechanism that also measures tether length, as shown in Figure 2.6.

2.3.4 The Evolution of TReX

Prior to TReX, a prototype, vScout, was designed within our lab and used a Clearpath Husky rover with a modular winch payload for ascending/descending a climbing rope (Stenning et al., 2015). With lessons learned from vScout, TReX was developed. The evolution of the TReX design, from initial concept to field-tested system, is illustrated in Figure 2.7. The earliest design called for multiple actuators in the tether management payload as a means to decouple winching from on-board spooling and enable self righting. However, we reduced the scope of this design to use a single actuator (not including the rover's wheel motors) for the sake of mechanical feasibility and control simplicity. The first iteration of the completed TReX system used a climbing rope, which limited the operation time to 20 minutes. Later, an electromechanical tether [3] was integrated. The tether allows for power (up to 1,200 W) and wired data transmission (up to 100 Mb/s) while supporting the weight of the robot (up to 900 kg force). The tether's diameter (9 mm) limits the on-board carrying capacity and drivable range of the robot to under 50 m. We note that the location of the stereo camera and inertial sensor have changed to accommodate for different experiments. For example, the stereo camera can be mounted

[3]Falmat XtremeNet Deep-Water Ethernet Cable – Model: FM022208-03-2K.

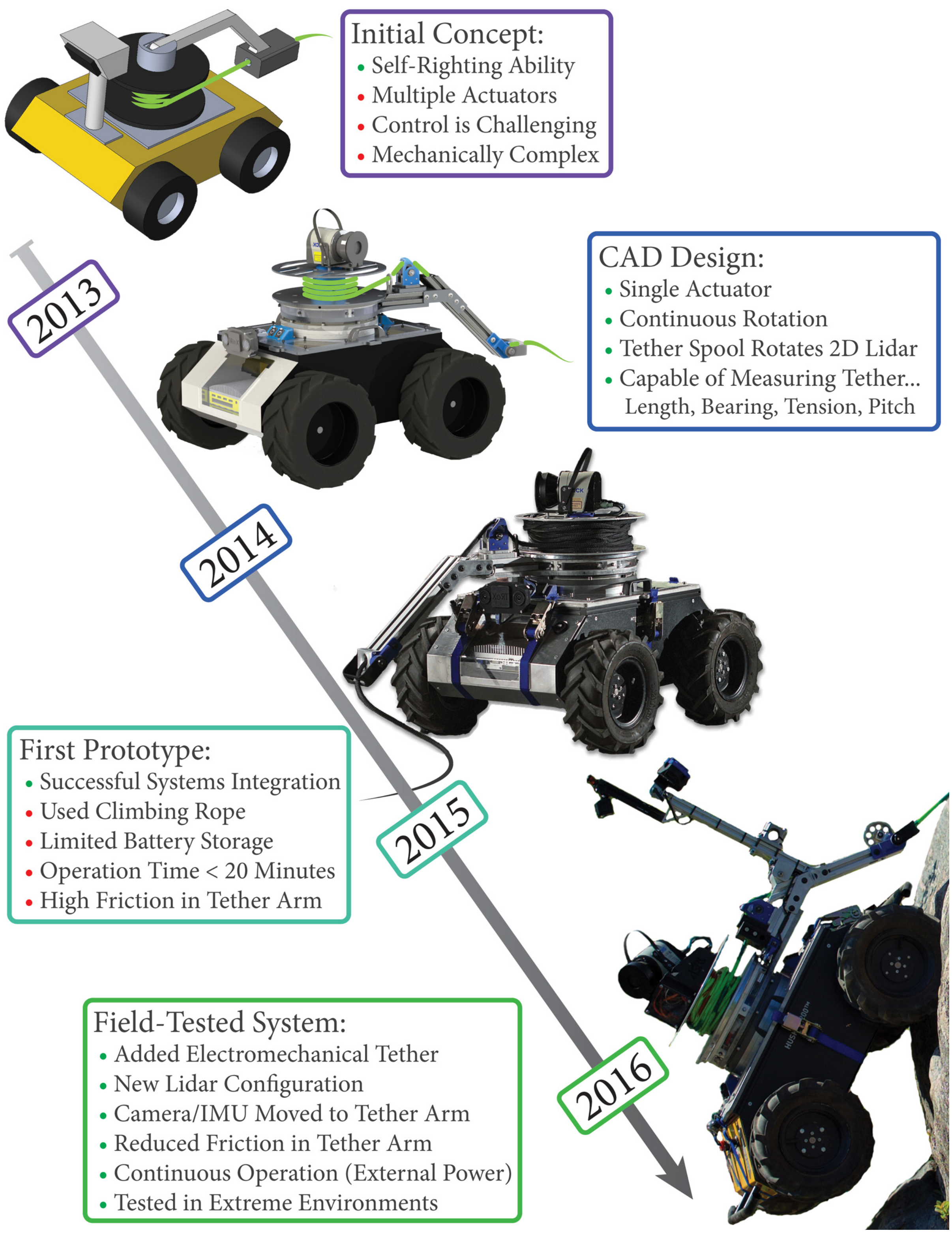


Figure 2.7: *The Evolution of TReX*: From initial concept to field-tested system. Green dots are features and achievements, while red dots denote problems and limitations

System Specifications

Mass	92 kg (including tether)
Dimension	1 x 0.7 x 1 m (L x W x H)
Power	24 VDC / 120 VAC
Platform	Clearpath Husky A200
Actuator	MMP 170 Nm Torque
Camera	Skybotix VI-Sensor
Lidar	Sick LMS 151
Computer	Logic Nuvo-3100VTC
DAQ	LabJack T7
Tether	Falmat FM022208-03-2K
Length	45 m (limits range)

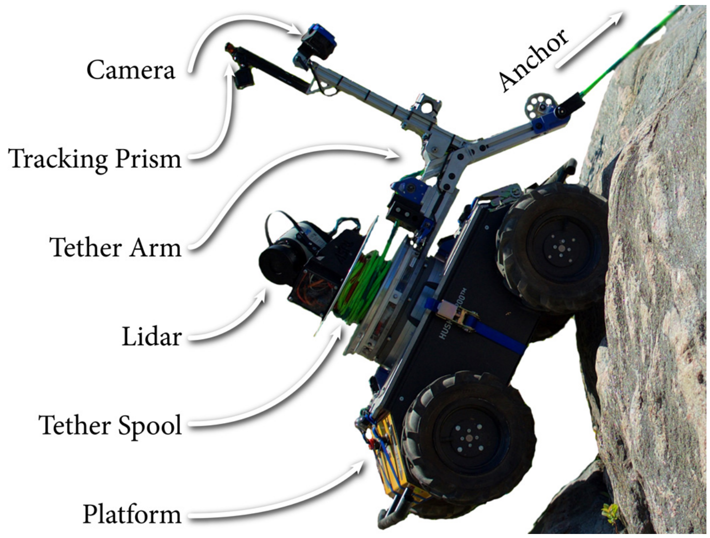


Figure 2.8: *TReX System Specifications*

(i) low on the robot body for navigating in tight spaces, (ii) on the robot's tether arm to leverage passive actuation of the camera for the purpose of scene stabilization, or (iii) on a camera mast attached to the robot to make VT&R work without modification (prevents continuous rotation under tension). Figure 2.8 provides system specifications and highlights features of the TReX system.

2.4 System Evaluation

Upon completion of the first TReX prototype (shown in Figure 2.7), we conducted a series of tests designed to evaluate the capabilities of the system. In particular, we looked into the advanced mobility of TReX on steep terrain while under tension and the performance of the customized tension measurement sensor. For detail on mapping, see Chapter 5.

2.4.1 Advanced Mobility on Steep Terrain

In order to test TReX's ability to rotate-in-place and travel laterally on steep terrain, the robot was manually piloted on an outdoor dome structure. Figure 2.9 provides a time-lapse image collected during the test, which demonstrates that the robot can turn in place and drive horizontally on steep, slippery terrain. The advanced mobility of TReX allows for more efficient mapping of the terrain, provided sufficient wheel traction.

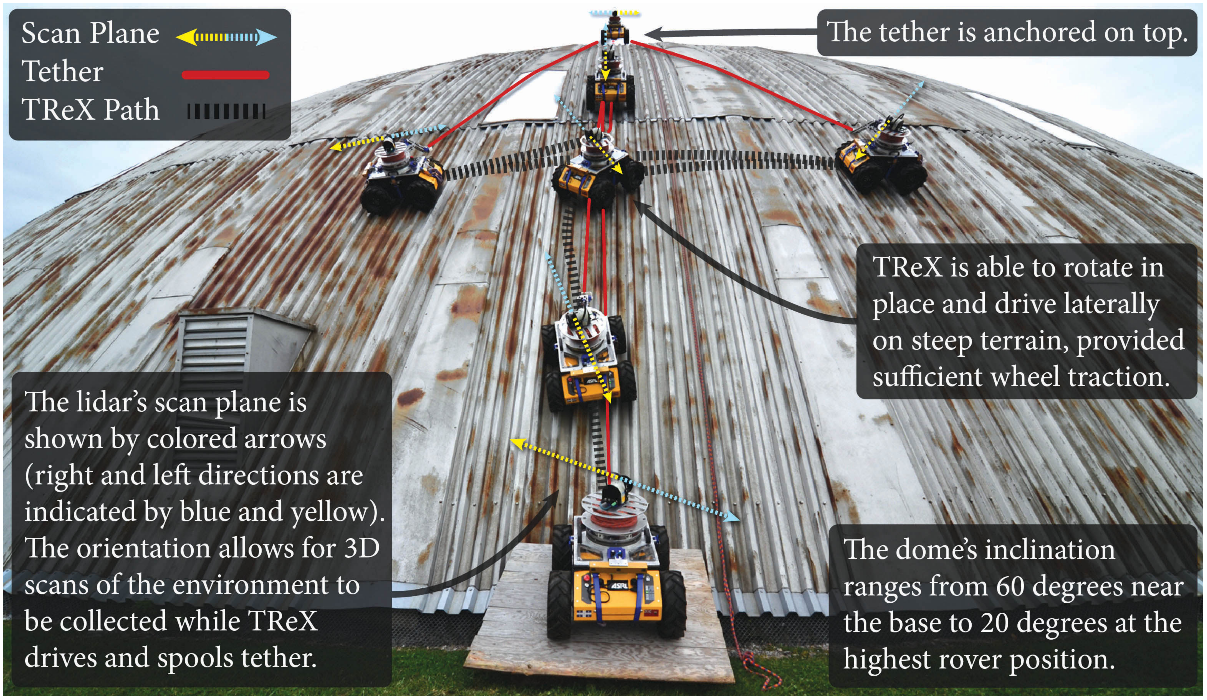


Figure 2.9: *Steep-Terrain Mobility Test*: TReX is piloted on the exterior of a metallic structure to demonstrate in-place rotation and lateral driving while under tension.

2.4.2 Measuring Tether Tension

Measuring the tension of an attached tether is a critical component to tether management. When the robot navigates flat, cluttered environments, keeping a taut tether prevents entanglement and eases spooling. On steep terrain, the tether supports the weight of the robot and prevents slippage. In either case, we want to ensure tether tautness regardless of the terrain and allow the robot to drive as if untethered. Prior to implementing the tether-control strategy detailed in Chapter 4, we evaluated the performance of the tension measurement sensor by (i) implementing a simple feedback controller on flat ground, (ii) utilizing an inclined plane to examine sensor response at varying slopes, and (iii) using a pull test to recalibrate the sensor after design changes.

Flat-Ground Test: We use a basic feedback controller to evaluate force sensor response on smooth, flat ground. A time-lapse image of the test is shown in Figure 2.10. Although the tether appears taut throughout, there is a difference in sensor output when reeling in/out, as shown by Figure 2.11. This difference is related to measurement hysteresis between loading and unloading conditions and sensor saturation at low tensions.

Inclined-Plane Test: An inclined-plane test was conducted in order to evaluate the tension sensor's response to varying slopes. Figure 2.12 shows the test setup and re-

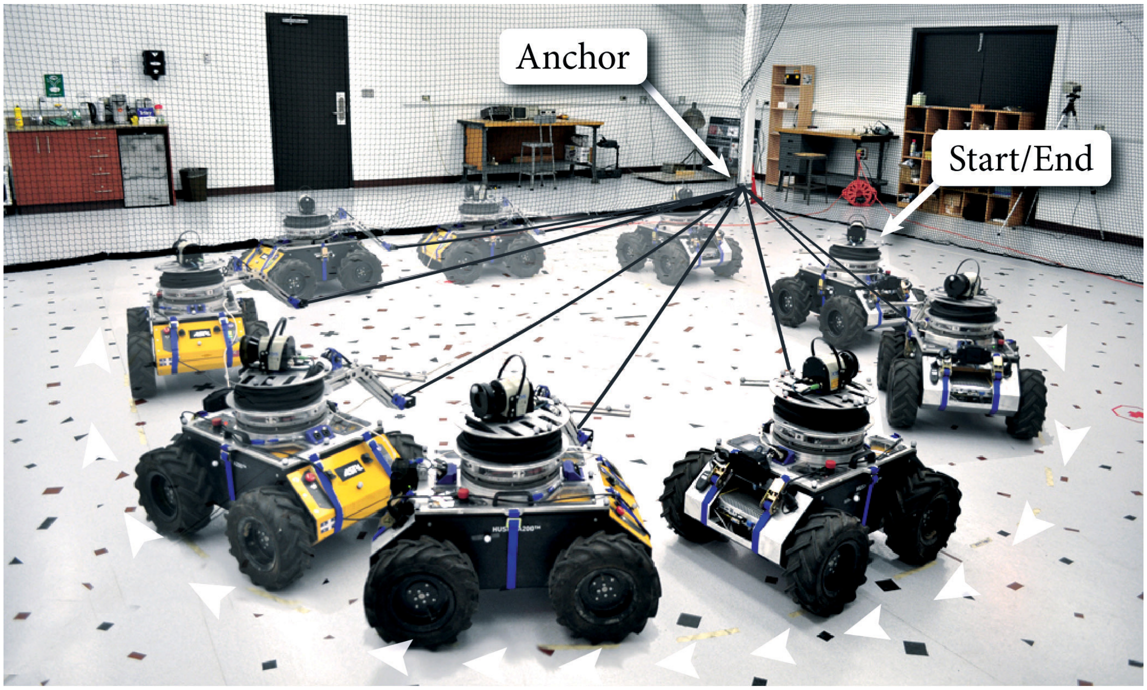


Figure 2.10: *Initial Tether-Control Test:* TReX was manually driven in two, clockwise loops on a flat, slippery surface while anchored to a beam. The purpose of the test is to evaluate the performance of an integrated tension measurement sensor. During the test, a simple proportional feedback controller was used to maintain a taut tether. Results from the test are provided in Figure 2.11.

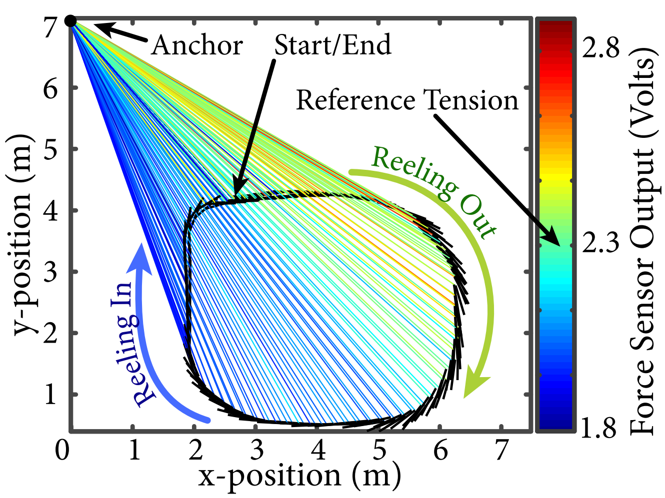


Figure 2.11: *Initial Tether-Control Results:* The ground-truth pose of the robot and orientation of the tether were recorded by a *Vicon* Motion Capture System. Colored lines drawn between the anchor and robot poses (black heading vectors) illustrate the discrepancy in sensor response between periods of reeling in and out. Due to saturation at low tension and measurement hysteresis, the tether is slack when reeling in.

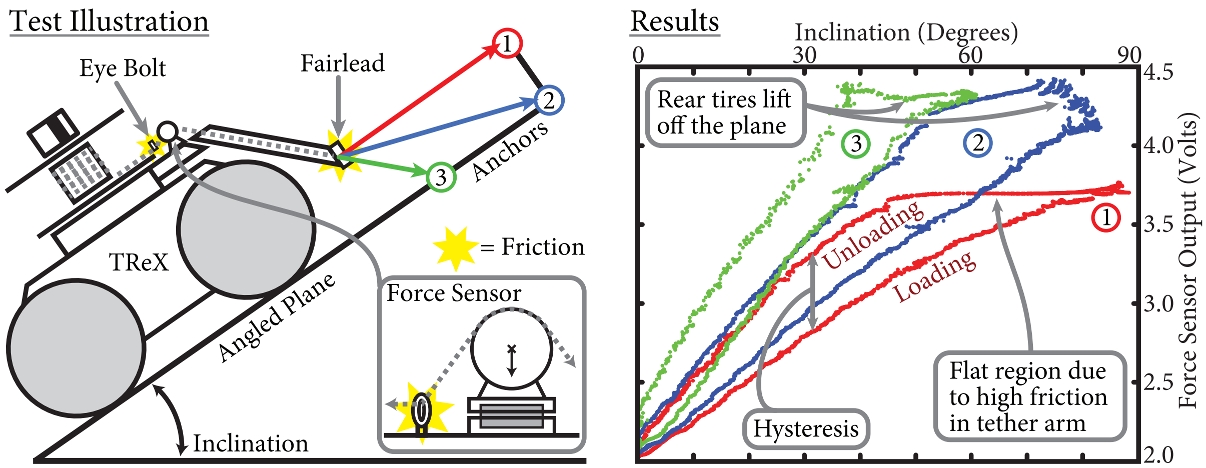


Figure 2.12: *Inclined-Plane Test:* The *left* figure illustrates the test setup. The angle of inclination is varied to observe tension sensor response to changing slopes while TReX rests stationary on a plane. Three tests are performed, each with different anchors, in order to evaluate the impact of friction in the design (shown as yellow stars). To avoid influencing tension measurement, we did not remove the eye bolt. The *right* figure shows the result of the test. At low inclinations, the sensor response is mostly linear with hysteresis between instances of loading (bottom curve) and unloading (top curve). The sensor response becomes more nonlinear with increasing inclination. Varying the anchor improves linearity, but hysteresis is still significant.

sults. Three variations of the test were conducted with different anchor attachments to determine how system friction contributes to measurement hysteresis. In all tests, we observe significant hysteresis between loading and unloading states. The worst case occurs when attached to the first anchor; the 'flat' response between 90 to 50 degrees inclination is caused by high friction between the tether and fairlead, which prevents the tether from slipping smoothly. For the second and third anchors, the linearity of the response improves due to reduced friction between the tether and fairlead.

Pull Test: In response to friction-related problems, the tether arm design was changed; the eye bolt and fairlead where replaced with passive pulleys. In addition, an electromechanical tether replaced the climbing rope used in previous tests. As illustrated by Figure 2.13, a simple pull test was performed to re-evaluate the force sensor. The pull test improves on the angled-plane test because we can evaluate the full range of tension allowable by the tether and keep the vehicle stationary, which limits the influence of wheel friction during calibration. While the robot's tether spool is completely capable of 'dead lifting' the weight of the robot, it was not possible to accurately test the tension beyond 70° of inclination on the angled plane because the robot's front tires would

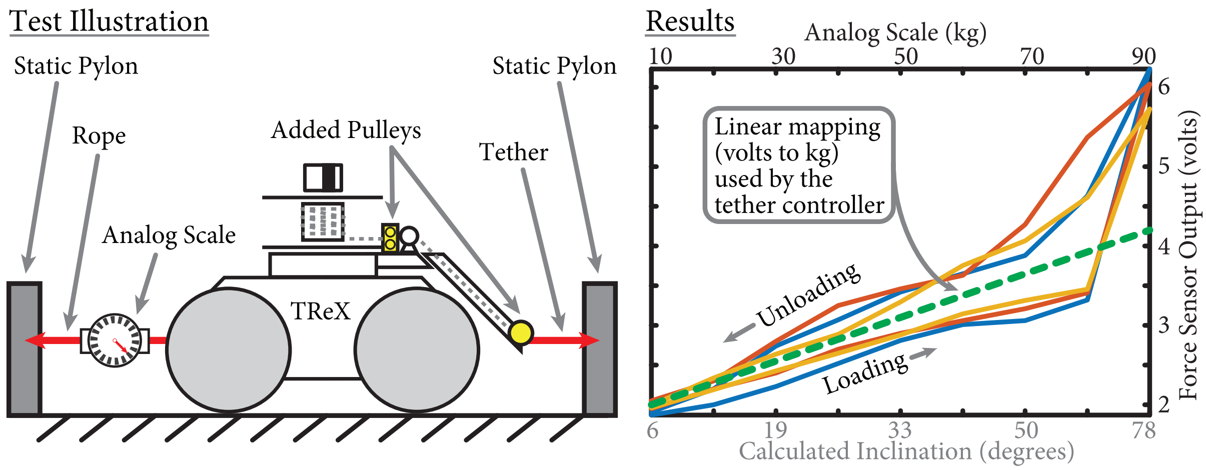


Figure 2.13: *Pull Test:* The *left* figure illustrates the test setup. With the robot anchored on one side by a rope attached to a pylon, the tether (attached to a second pylon) is progressively pulled tighter. The force sensor's voltage is recorded at 10 kg increments, which are determined using an analog scale placed in line with the rope. The *right* figure shows the result from three cycles of the test. The response is still influenced by hysteresis. A linear mapping is shown, which is used to maintain tension at flat to mild inclinations. For comparison, we take the arcsine of the ratio between the analog scale reading and TReX's total weight to get a calculated inclination.

leave the plane and cause a 'wobbling' effect, which prevented calibration over the full expected range of tension. However, the 'wobbling' effect is a concern for stability[4] on steep terrain and could limit the safe region of exploration. During the simple pull test, tension was gradually increased while the voltage output of the sensor was recorded in 10 kg increments (read from an analog scale). At 90 kg (the mass of the robot), the tension was decreased in order to record the unloading response. Three cycles of the test were performed to observe the impact of hysteresis on sensor output. As shown, the sensor response over multiple cycles results in a similar hysteresis pattern. Despite some hysteresis, the output is improved from the angled-plane test due to the improved testing method and reduced system friction.

For the tether controller presented in Chapter 4, we use a linear mapping for sensor response as indicated in Figure 2.13. We are able to do so because the controller leverages feedback and feedforward control; feedback is prominent in flat to mild inclines where sensor response is roughly linear, while feedforward dominates in steep areas when sensor response is strongly influenced by hysteresis.

[4]We did not perform a comprehensive stability analysis on the platform.

2.5 Novel Contributions

Although much effort went towards the mechanical design and testing of TReX, the main novelty of this chapter and the system presented is the achievement of enhanced tethered robot mobility through in-place, continuous rotation while under tension.

2.6 Associated Publication

- McGarey et al. (2015). System Design of a Tethered Robotic Explorer (TReX) for 3D Mapping of Steep Terrain and Harsh Environments. In the *2015 International Conference on Field and Service Robotics (FSR)*.

2.7 Associated Videos

- Intro to TReX: https://youtu.be/Q2gOOhK45lY
- Mechanical Design: https://youtu.be/iQYULj8TLWk
- Building TReX: https://youtu.be/i7e7iHxMmu0

Chapter 3

TSLAM: Tethered Simultaneous Localization & Mapping

3.1 Motivation

A tether attached to a mobile robot can provide support, power, and communication, but also constrains navigation (Sinden, 1990). In cluttered environments, the robot's tether will come into contact with obstacles and form intermediate anchors as shown in Figure 3.1. To detach from added anchors, excluding the initial, the robot must backtrack along its outgoing path in order to allow for each anchor to be removed in sequence (Teshnizi and Shell, 2014). One way to approach this problem is to detect and localize the position and sequence of any active anchors (i.e., obstacles currently touching the tether) with respect to the robot's trajectory. Accordingly, we propose tethered simultaneous localization and mapping (TSLAM), which is a variant of SLAM that leverages nonvisual sensors to jointly solve for the robot's pose and position/sequence of anchors. TSLAM is nonvisual because it fuses tether length and bearing-to-anchor (i.e, tether arm angle) measurements with wheel odometry, which could allow it to work in dark and dusty environments. TSLAM is comparable to range/bearing SLAM performed on a single landmark (i.e., anchor) with one critical difference; the tether length measurement (i.e., range) is a function of all active anchors in contact with the tether. Normally in SLAM, each measurement is a function of the robot pose and just one landmark, which is no longer true in TSLAM. To account for this difference, we can either (i) approximate measurement independence by assuming that uncertainty is limited to the currently observed anchor, or (ii) account for uncertainty across all anchors since tether measure-

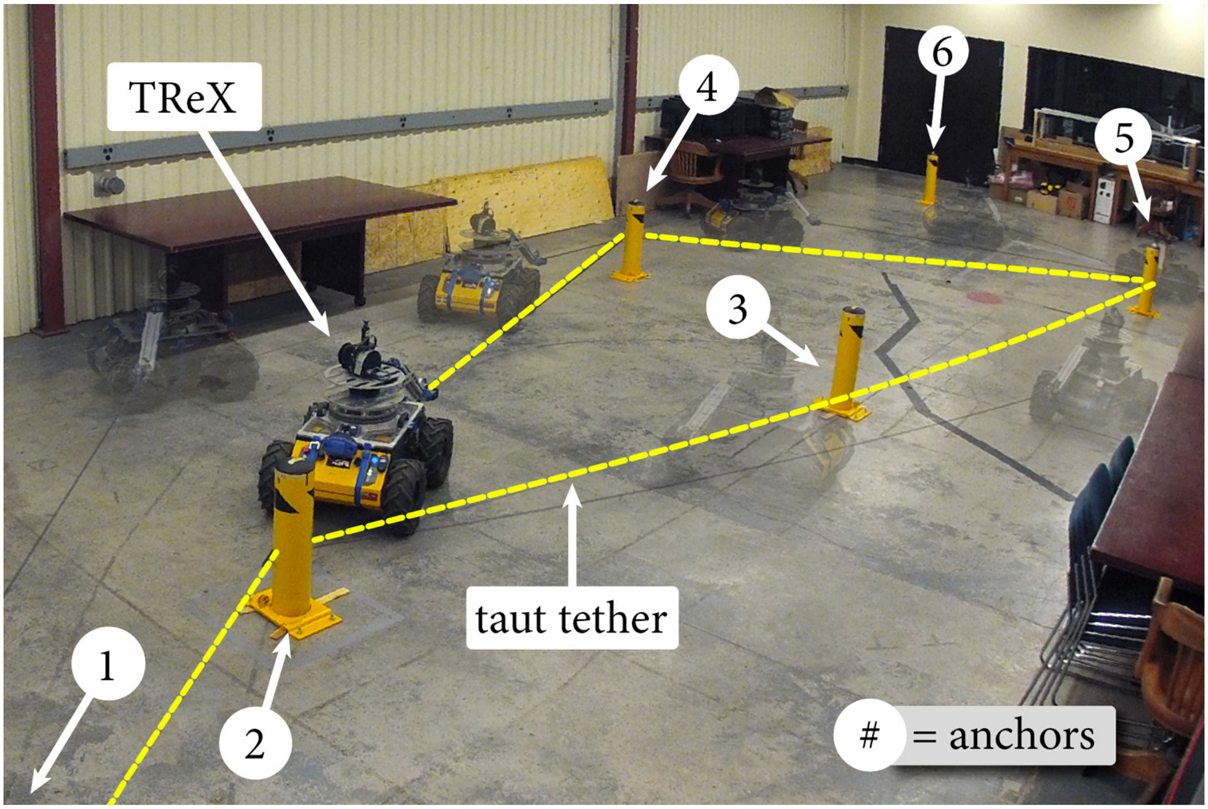


Figure 3.1: *The Problem with Tethers*: As TReX navigates through cluttered environments, its tether will come into contact with obstacles and form intermediate anchors. To avoid tether entanglement, we want to estimate the robot's pose and positions of anchors using nonvisual tether length, bearing-to-anchor, and wheel odometry measurements.

ments can now involve all the active anchors. In visual SLAM, if we were to assume that (ii) is true and that measurements are dependent, solving the problem would not only be inefficient, but potentially intractable, because thousands of landmarks may be observed. However, in TSLAM, we can exploit the fact that only tens of anchors will be added and observed in a single traverse, which allows for efficiently solving the problem. We propose two methods to solve the planar (2D) TSLAM problem using (i) an online particle filter based on FastSLAM that approximates measurements as independent, and (ii) an efficient batch formulation that allows for anchor measurements to be dependent. We test each solution to the TSLAM problem on both simulated and experimental data collected by TReX. The results show that each method reduces dead-reckoned errors from odometry and allows for anchors to be detected and localized. Overall, the batch approach performs best because it allows for measurements to depend on all the anchors and has superior outlier rejection to cope with noisy sensor data.

This chapter proceeds as follows. Section 3.2 highlights related works on SLAM and tether-based localization. Section 3.3 details the TSLAM problem and setup. Section 3.4 evaluates TSLAM on a variety of datasets. Section 3.5 discusses lessons learned from experiment. Section 3.6 offers concluding remarks and future extensions for TSLAM. Section 3.7 states novel contributions.

3.2 Related Work

With the goal of preventing tether entanglement, we want to develop an approach to landmark-based SLAM that exploits the tether sensors on our mobile robot. The first solution to the general SLAM problem, which leverages an Extended Kalman Filter (EKF), was proposed by Smith et al. (1990). However, the inspiration for SLAM owes itself to the classic bundle adjustment problem for automated photogrammetry introduced by Brown (1958). Triggs et al. (2000) was responsible for later popularizing this work. Subsequently, modern advances to landmark-based SLAM were made in surveys by Durrant-Whyte and Bailey (2006) and Bailey and Durrant-Whyte (2006). In this chapter, we will detail two solutions to the tether-based SLAM problem; (i) an online, particle-filter approach inspired by FastSLAM from Montemerlo et al. (2002), and (ii) a batch approach that resembles graph-SLAM from Thrun and Montemerlo (2006), which is a popularized version of a technique originally introduced by Lu and Milios (1997). What makes TSLAM different from SLAM is that tether measurements are a function of the robot's pose and potentially all active anchors, whereas in SLAM, measurements are a function of the robot's pose and just a single landmark. While tether measurements have been exploited to aid in robot localization in the past (Kumar and Richardson, 2008; Corominas-Murtra and Tur, 2013; Rajan et al., 2014), no prior formulation allows for mapping anchor points using such measurements.

3.3 TSLAM

Prior to detailing each TSLAM approach, we must first set up the general problem and define common variables that will be used throughout this chapter. The TSLAM problem is best illustrated by Figure 3.2, which shows the tethered robot's trajectory

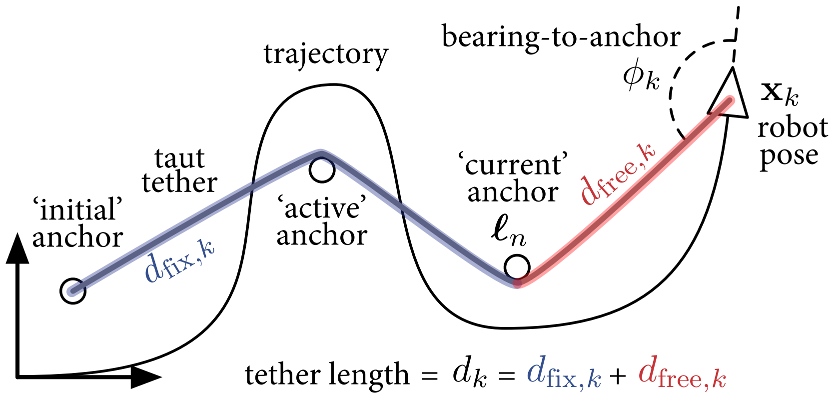


Figure 3.2: *The TSLAM problem*: The robot's pose and anchor positions are unknown and will be estimated using tether measurements (e.g., length and bearing-to-anchor) and wheel odometry. The taut tether's length can be divided into 'fixed' and 'free' components to distinguish between tether wrapped around active anchors and tether joining the robot's pose to the current anchor. However, we can only measure the total tether length and must estimate the anchors to determine 'fixed' and 'free' lengths.

through obstacles on a plane. We note that TSLAM is formulated in 2D[1], given the nonvisual sensors involved. We define the robot's trajectory as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \tag{3.1}$$

with k representing a discrete time-step index and K being the maximum time. The robot's pose on the plane is described by a 2D position, x_k and y_k, and orientation, θ_k. As usual in SLAM, we assume the initial robot pose is known by setting $\mathbf{x}_0 = (0, 0, 0)$. An ordered list represents anchors that are in contact with the tether according to

$$\boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_1 \\ \vdots \\ \boldsymbol{\ell}_N \end{bmatrix}, \quad \boldsymbol{\ell}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix}, \tag{3.2}$$

with n being the 'current' anchor index (first along tether from robot) and N the total number of anchors observed. Each anchor has a location x_n and y_n in the plane. We make the assumption that the tether remains taut while driving, which results in straight-line distances between anchors, which can be represented by zero-radius points. We

[1] Extending TSLAM to 3D would be possible but is beyond the scope of this work.

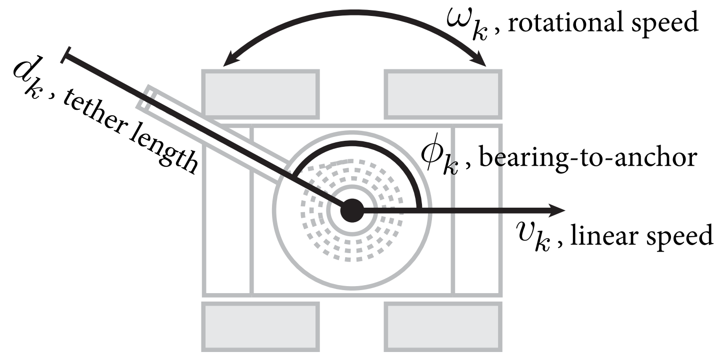


Figure 3.3: *Tethered-Robot Model*: This model shows measurements that are collected by TReX to use as inputs to the TSLAM problem. The robot base is a four-wheeled, skid-steered platform that can be modeled kinematically as a unicycle. Tether measurements at time k involve the bearing-to-anchor, ϕ_k, and deployed tether length, d_k. Body-centric linear and rotational velocities, v_k and ω_k, are used to compute wheel odometry.

acknowledge that the point representation is used to simplify the problem as a first approach. In reality, physical anchors have a non-zero radius (e.g., trees and rocks). Section 3.5 discusses the impact of the point representation on our experiments. Tether measurements, as illustrated in Figure 3.3, are expressed as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} d_k \\ \phi_k \end{bmatrix}, \tag{3.3}$$

where each measurement is a nonlinear function of the robot's pose and all active anchors:

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}) + \mathbf{n}_k, \tag{3.4}$$

with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ being the zero-mean, Gaussian noise with covariance $\mathbf{R}_k$. The first component of the tether measurement is the bearing-to-anchor, ϕ_k, which is modeled as

$$\phi_k = \operatorname{atan2}(y_n - y_k, x_n - x_k) - \theta_k. \tag{3.5}$$

The bearing involves just the current anchor, $\boldsymbol{\ell}_n$, and robot pose, $\mathbf{x}_k$. Tether length, d_k, comprises 'fixed' and 'free' lengths, with

$$d_k = d_{\text{fix},k} + d_{\text{free},k} \ . \tag{3.6}$$

The fixed length, a function of the 'active' anchors, is modeled as a sum of normal distances between anchors (depends on tether history),

$$d_{\text{fix},k} = \|\boldsymbol{\ell}_1 - \boldsymbol{\ell}_2\| + \ldots + \|\boldsymbol{\ell}_{n-1} - \boldsymbol{\ell}_n\|\,, \tag{3.7}$$

where 'active' describes anchors currently in contact with the tether. Anchors $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_n$ (i.e., the initial and current anchor) are always active. In the case that $\boldsymbol{\ell}_n$ is $\boldsymbol{\ell}_1$, then only one anchor is active, and its fixed length must be zero. Due to the way that tether history (i.e., the order that anchors are added) is determined for each solution to TSLAM, the sequence of anchors between $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_n$ can be represented differently depending on the estimation method used. We detail how each method handles this sequence in Sections 3.3.1 and 3.3.2.

The free tether length is more simply the distance between the current anchor and robot:

$$d_{\text{free},k} = \left\|\boldsymbol{\ell}_n - \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right]\right\|. \tag{3.8}$$

The fact that the fixed and free lengths contribute to the measurement, d_k, is what makes TSLAM different from range-bearing SLAM; In TSLAM, each measurement is a function of *not just one*, but *all* the active anchors. Intuitively, this makes sense since perturbing any of the anchors touching the tether will alter the total tether length. In order to compute wheel odometry, we use body-centric velocities, $\mathbf{v}_k$:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}, \tag{3.9}$$

where v_k and ω_k are the linear-translational and rotational velocity as illustrated in Figure 3.3. These velocities are sampled in constant time and are used to propagate a nonlinear motion model and calculate odometry. The motion model is of the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k), \tag{3.10}$$

where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is zero-mean, Gaussian process noise with covariance $\mathbf{Q}_k$. Given the definitions above, in TSLAM we want to solve the joint likelihood,

$$p(\mathbf{x}, \boldsymbol{\ell}|\mathbf{y}, \mathbf{v}), \tag{3.11}$$

where the robot's trajectory and anchors are estimated given nonvisual measurements from tether and odometry. In the following sections, two solutions to the TSLAM problem are detailed: (i) an online particle filter, and (ii) an efficient batch formulation. Appendix B.1 provides expanded math formulations for TSLAM.

3.3.1 Particle-Filter Method

One way to approach the online TSLAM problem is to use a particle filter to capture uncertainty in the robot's motion through obstacles. Specifically, a close encounter with an obstacle can cause the robot to be uncertain about the detection and or wrapping direction of its tether on a new anchor, which results in a multi-modal distribution of uncertainty. A particle filter is tailored to handle multi-modal problems, where the trajectory of the robot around an obstacle will have direct implications on whether an anchor is detected or not. Over time, particles representing a belief of the robot's trajectory that pass on the correct side of an obstacle will be favored to survive because they best represent the state of the robot and map of anchors. We use a particle-filter implementation that is based on FastSLAM (Montemerlo et al., 2002), where particles are used to represent the robot's trajectory and Gaussians are used to bookkeep and update landmark (i.e., anchor) locations. FastSLAM derives from Monte Carlo Localization (MCL) (Thrun et al., 2001), which allows for samples (i.e., robot poses), each with a different belief of the map (i.e., anchors), to jointly represent the state. The fundamental advantage of the FastSLAM formulation is that the joint likelihood can be factored accordingly,

$$p(\mathbf{x}, \boldsymbol{\ell} | \mathbf{y}, \mathbf{v}) = \underbrace{p(\mathbf{x} | \mathbf{y}, \mathbf{v})}_{\text{particles}} \underbrace{p(\boldsymbol{\ell} | \mathbf{x}, \mathbf{y})}_{\text{Gaussians}} , \tag{3.12}$$

where anchors can be represented by Gaussians with a mean for position, and a covariance, which tracks uncertainty. To adapt, we write our anchor list, $\boldsymbol{\ell}$, as

$$\boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_1 \\ \vdots \\ \boldsymbol{\ell}_N \end{bmatrix}, \quad \boldsymbol{\ell}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n), \tag{3.13}$$

where N is the total number of active anchors and individual Gaussians describe an anchor's 2×1 mean position, $\boldsymbol{\mu}_n$, and 2×2 covariance, $\boldsymbol{\Sigma}_n$. Our iterative, FastSLAM pipeline is illustrated in Figure 3.4.

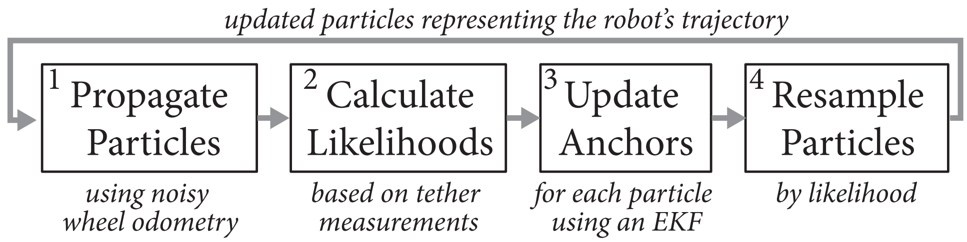


Figure 3.4: *Particle-Filter Pipeline*: The particle filter is based on a FastSLAM implementation and is iterated over time to estimate the robot's trajectory and positions of anchors. Each particle can bookkeep uncertainty and update the anchors using the same EKF-map update step as used in FastSLAM.

Adapting FastSLAM:

Modifications were necessary to make FastSLAM work for tether measurements. Normally, landmark measurements are assumed to be conditionally independent in FastSLAM (i.e., measurements only depend on a single landmark and robot pose), which enables the continued factorization of $p(\boldsymbol{\ell}|\mathbf{x},\mathbf{y})$ into a product of individual Gaussians (i.e., one for each landmark). Since tether length measurements depend on all the active anchors, we make an approximation that anchors are independent in order to preserve the factorization. The approximation impacts the measurement likelihood as follows,

$$p(\mathbf{y}_k|\mathbf{x}_k,\boldsymbol{\ell}) \approx p(\mathbf{y}_k|\mathbf{x}_k,\boldsymbol{\ell}_n) \ , \tag{3.14}$$

where only the current anchor, $\boldsymbol{\ell}_n$, influences the measurement at the current time, k. Figure 3.5 illustrates the impact of this approximation on updating the list of anchors. It also follows that the anchor map likelihood becomes

$$p(\boldsymbol{\ell}|\mathbf{x},\mathbf{y}) \approx \prod_{n=1}^{N} p(\boldsymbol{\ell}_n|\mathbf{x},\mathbf{y}) \ , \tag{3.15}$$

where N is the total number of active anchors. The approximation has the effect of limiting uncertainty to the current anchor at any given time, which restricts the algorithm's ability to make updates to older parts of the map, and makes it susceptible to localization errors caused by outlier measurements. However, the effect of the approximation should be limited unless the robot comes back to the same anchor a second time (i.e., large-scale loop closure). The batch method, described in Section 3.3.2, was formulated to account for conditionally dependent measurements and better handle outliers.

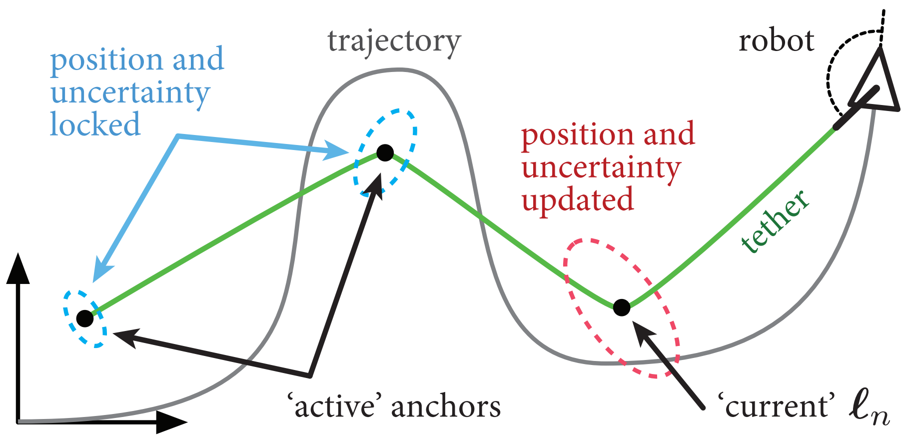


Figure 3.5: *Updating Anchors*: This illustration shows that the approximation made to preserve measurement independence limits the ability to update the position and uncertainty to just the current anchor, $\boldsymbol{\ell}_n$. The 'active' anchors remain locked to updates until they are reobserved on the robot's return trajectory.

Anchor Detection:

Given the online nature of particle filter approach, new anchors must be detected on-the-fly and placed in a sequential list, which gives the tether's obstacle interaction history. To account for the fact that tether length measurements are a function of that list, we revisit the idea of dividing that measurement into 'fixed' and 'free' components. Since each particle maintains its own anchor list and tether history, we can use equation (3.7) to divide the measured length into components. We can leverage the 'free' component of tether length, along with a direct bearing-to-anchor measurement, to determine the initial position for a previously unobserved anchor. Consider the scenario shown in Figure 3.6, where a tethered robot is driving past a potential anchor. We can use the ellipse model shown in Figure 3.7 to leverage tether, odometry, and prior anchor measurements to identify a new candidate anchor by exploiting basic geometry. Specifically, the robot's current position, $\mathbf{x}_k$, and anchor, $\boldsymbol{\ell}_n$, form the foci of an ellipse, whose boundary is defined by an extra amount of 'free' tether. A unique position on the ellipse boundary can be inferred with knowledge of the current robot heading and bearing-to-anchor measurement, which has deviated from what we would expect.

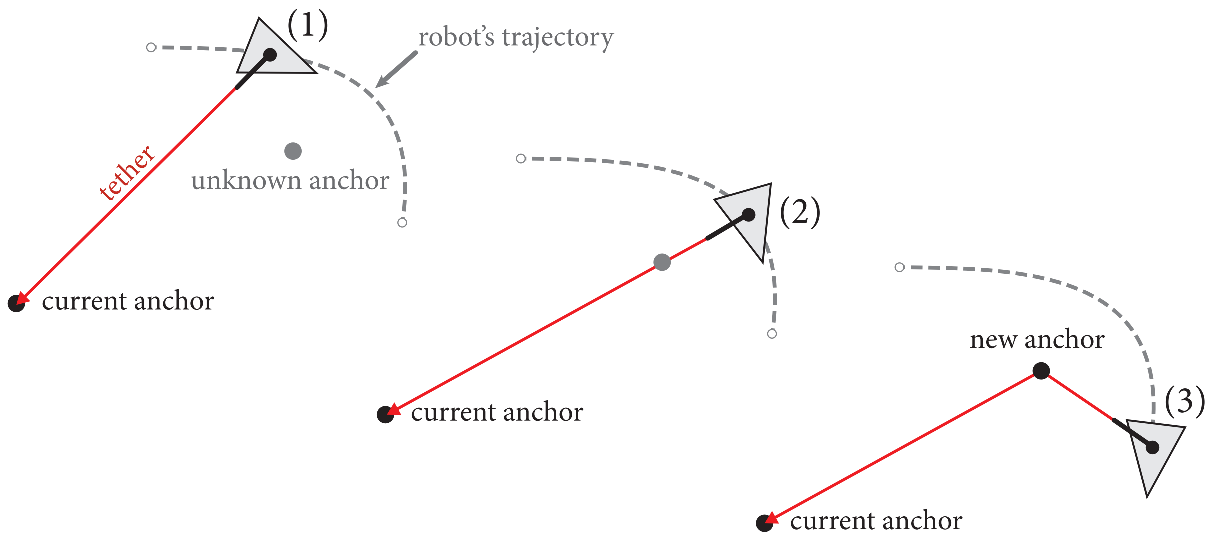


Figure 3.6: *Encountering an Unknown Anchor*: As a tethered robot passes an unknown anchor (grey circle) in this time-lapse sequence, the tether length and bearing-to-anchor measurement change unexpectedly, which occurs because the measurement is now to a newly added anchor and not the prior anchor, as expected.

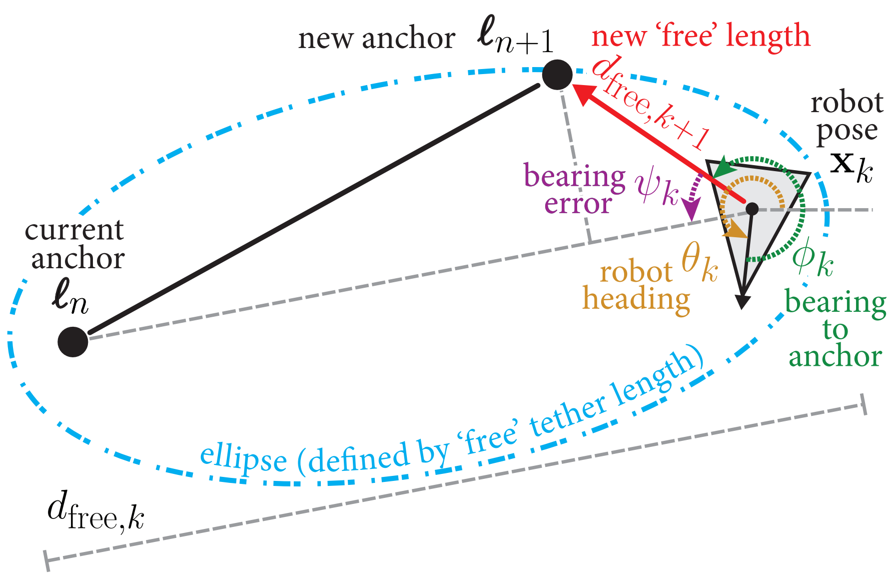


Figure 3.7: *Ellipse Measurement Model*: The ellipse represents all possible locations for a new anchor, $\boldsymbol{\ell}_{n+1}$. The inputs to the model are the 'free' length, $d_{\text{free},k}$, robot's pose, $\mathbf{x}_k$, and current anchor, $\boldsymbol{\ell}_n$. The combined bearing-to-anchor, ϕ_k, and robot heading, θ_k, point to a unique location on the ellipse. The error between the actual and expected bearing-to-anchor measurement, ψ_k, is used to find the new 'free' distance, $d_{\text{free},k+1}$, to the new anchor. The mean position of the new anchor is calculated using equation (3.16).

The steps to detect a new anchor are as follows:

1. calculate $d_{\text{free},k} = d_k - d_{\text{fix},k}$
2. a new anchor has been detected if $d_{\text{free},k}$ is greater than the distance between $\mathbf{x}_k$ and $\boldsymbol{\ell}_n$ (if not, the formed ellipse is ill-conditioned),
3. set up the ellipse with values from $\mathbf{x}_k$, $\boldsymbol{\ell}_n$, $d_{\text{free},k}$, and ϕ_k according to Figure 3.7,
4. compute the position of the new anchor according to equation (3.16).

The mean position of a new anchor, $\boldsymbol{\ell}_{n+1}$, is initialized as

$$\boldsymbol{\mu}_{n+1} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - d_{\text{free},k+1} \begin{bmatrix} \cos(\phi_k + \theta_k) \\ \sin(\phi_k + \theta_k) \end{bmatrix}, \tag{3.16}$$

which requires solving for $d_{\text{free},k+1}$, the new 'free' length after a new anchor is added, according to

$$d_{\text{free},k+1} = \frac{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - (d_{\text{free},k})^2}{2(d_{\text{free},k}) - \left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos \psi_k}, \tag{3.17}$$

where ψ_k, is the offset or error between the expected and measured bearing-to-arm measurement in the global frame as shown in Figure 3.7.

Applying Particle Weights:

As per FastSLAM, particle weights are calculated according to their uncertainty. The weight for an individual particle, m, is given by the likelihood function,

$$w_k^m \approx \int \underbrace{p(\mathbf{y}_k | \mathbf{x}_k^m, \boldsymbol{\ell}_n^m)}_{\substack{\text{observation} \\ \text{model}}} \underbrace{p(\boldsymbol{\ell}_n^m | \mathbf{x}^m, \mathbf{y}^m)}_{\substack{\text{Gaussian from} \\ \text{last update}}} \, d\boldsymbol{\ell}_n^m . \tag{3.18}$$

Weights are calculated for M total particles and are used when resampling to select (with greater likelihood) particles that best represent the state given the measurements.

Adding, Updating, and Removing Anchors:

In order to represent the likelihood of adding, updating, or removing an anchor from a given particle's list, we take a probabilistic approach and split each particle into three different 'copies' of itself just after resampling. The copied particles will start with the same initial pose as their parent particle, but will have different beliefs about their anchor list; the first copy adds a new anchor according to the ellipse measurement model, the second will simply update its belief of the current anchor, and the third will remove an anchor from the list and perform an update on the previous anchor. If the given particle has a list of n current anchors, then it will make copies of itself with anchor lists of length $n+1$, n, and $n-1$. The particle that removes an anchor is performing loop closure by reobserving and updating its belief about a previous anchor. Splitting particles in this manner has the benefit of allowing anchors to be detected even when measurements are noisy, which causes the ellipse model to be degenerate in some cases; a particle with a poor approximation of the anchor list/map will be unlikely to survive successive resampling. This method is also effective in cases where the decision to add an anchor is delayed due to the trajectory of the robot. For example, if the robot drives straight just after encountering an obstacle and later turns, it may take time before the measurements are indicative of a new anchor and manifest themselves in the particle weights.

Any updates to the belief of an anchor are made using a Kalman filter similar to the FastSLAM update. This step differs from conventional EKF-SLAM, which performs an update on both the robot's trajectory and landmarks. As per FastSLAM, we only need to compute a low-dimensional (2D) update to the mean and covariance for a given Gaussian representation of an anchor because particles are used to represent the robot's trajectory. As such, our approach requires that a sufficient number of particles are used to capture the robot's trajectory. Particle quantity is tied to our expectation for odometry process noise, which, in practice, is between 500-1000 particles; this accounts for the fact that the number of particles will triple in between resampling events (e.g., 1500-3000 particles). Changes to the anchor list are only allowed after resampling, which, depending on frequency, means that anchors are updated for a majority of particle-filter iterations. Additional details and math regarding the anchor initialization and Kalman update process can be found in Appendix B.1.

3.3.2 Batch Method

With lessons learned from the particle-filter implementation, the batch approach was formulated to (i) consider the impact of uncertainty across all anchors at once, and (ii) be robust to outlier measurements. The batch method also has the advantage of using all the measurements simultaneously, which means that a uni-modal, Gaussian representation can be used for measurement uncertainty. This representation is possible in batch because we better handle the detection of anchors during initialization and do not require maintaining multiple hypotheses of the anchor list like the particle-filter approach. What follows is an explanation of our batch methodology, where, along the way, important differences are highlighted from the particle-filter approach.

In batch TSLAM, we use the standard maximum-a-posteriori (MAP) approach to convert (3.11) into the optimization problem,

$$\{\mathbf{x}^\star, \boldsymbol{\ell}^\star\} = \arg\min_{\mathbf{x},\boldsymbol{\ell}} J(\mathbf{x}, \boldsymbol{\ell}), \tag{3.19}$$

where $J(\mathbf{x}, \boldsymbol{\ell}) = -p(\mathbf{x}, \boldsymbol{\ell}|\mathbf{v}, \mathbf{y}) + C$ is the objective function. The constant, C, denotes terms in the negative log-likelihood, which are independent of $\mathbf{x}$ and $\boldsymbol{\ell}$. We use the Gauss-Newton approach to solve the MAP optimization. Accordingly, we start with an initial estimate for the states, $\mathbf{x}_{\text{op}}$ and $\boldsymbol{\ell}_{\text{op}}$, and compute perturbations to each as

$$\mathbf{x} = \mathbf{x}_{\text{op}} + \delta\mathbf{x}, \quad \boldsymbol{\ell} = \boldsymbol{\ell}_{\text{op}} + \delta\boldsymbol{\ell}, \tag{3.20}$$

where δ represents a small perturbation to the state, which minimizes the cost of equation (3.19). As per Gauss-Newton, optimal perturbations, $\delta\mathbf{x}^\star$ and $\delta\boldsymbol{\ell}^\star$, are found by iteratively solving a linear system of equations:

$$\underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \delta\mathbf{x}^\star \\ \delta\boldsymbol{\ell}^\star \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}}_{\mathbf{b}}. \tag{3.21}$$

Perturbations are applied to the state as

$$\mathbf{x}_{\text{op}} \leftarrow \mathbf{x}_{\text{op}} + \delta\mathbf{x}^\star, \quad \boldsymbol{\ell}_{\text{op}} \leftarrow \boldsymbol{\ell}_{\text{op}} + \delta\boldsymbol{\ell}^\star, \tag{3.22}$$

and the batch process iterates until convergence.

In conventional SLAM, the $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ components from equation (3.21) are normally block-tridiagonal and block-diagonal respectively, as illustrated in Figure 3.8. The conventional approach is to exploit the block-diagonal sparsity of $\mathbf{A}_{22}$ using a Schur complement (Zhang, 2006) or sparse Cholesky decomposition (Dereniowski and Kubale, 2004) to solve the system of equations efficiently. Recall that $\mathbf{A}_{22}$ is block-diagonal be-

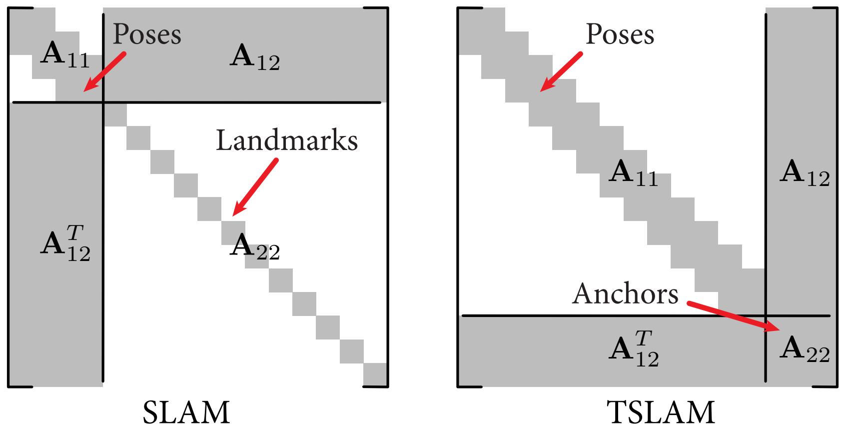


Figure 3.8: *Sparsity Comparison*: In comparing the sparsity patterns for the $\mathbf{A}$ matrix from equation (3.21) between batch SLAM and TSLAM, we find a clear discrepancy; for TSLAM, the $\mathbf{A}_{22}$ block, which is normally sparse in SLAM, is dense. However, we can leverage the fact that tens of anchors will be encountered in a traverse versus thousands of visual landmarks and, instead, exploit the block-tridiagonal sparsity of $\mathbf{A}_{11}$.

cause measurements are normally a function of one robot pose and landmark, regardless of the number of landmarks observed at any given time (i.e., measurements are independent). In TSLAM, measurements are dependent on all the active anchors, which changes the sparsity of $\mathbf{A}_{22}$, as shown in Figure 3.8. The $\mathbf{A}_{11}$ block remains block-tridiagonal because odometry is still computed between two consecutive robot poses at a time. A real example of the A-matrix sparsity from experiment is provided in Figure 3.21.

Alternatively, our batch approach exploits the block-tridiagonal sparsity of $\mathbf{A}_{11}$ to efficiently solve equation (3.21). The solve is still efficient because the robot is limited to travel by a finite length of tether, which means we will encounter significantly fewer anchors than robot poses during a single traverse: $N \ll K$. This is the opposite of conventional SLAM, where visual landmarks can vastly exceed the number of robot poses. In TSLAM, $\mathbf{A}_{11}$ is exploited using a sparse Cholesky decomposition of $\mathbf{A}$:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T, \tag{3.23}$$

where $\mathbf{L}$ is the sparse, lower-triangular Cholesky factor used to solve $\mathbf{Lc} = \mathbf{b}$ for $\mathbf{c}$ and $\mathbf{L}^T\mathbf{z} = \mathbf{c}$ for the optimal perturbation, $\mathbf{z}$. In terms of efficiency, the decomposition complexity is $O(N^3 + N^2K)$ per batch iteration. For comparison, in a conventional SLAM formulation, the complexity would be $O(K^3 + K^2N)$, where N would represent the maximum landmarks observed, and the sparsity of measurements would be exploited instead. Further details on the Cholesky decomposition are provided in Appendix B.1.

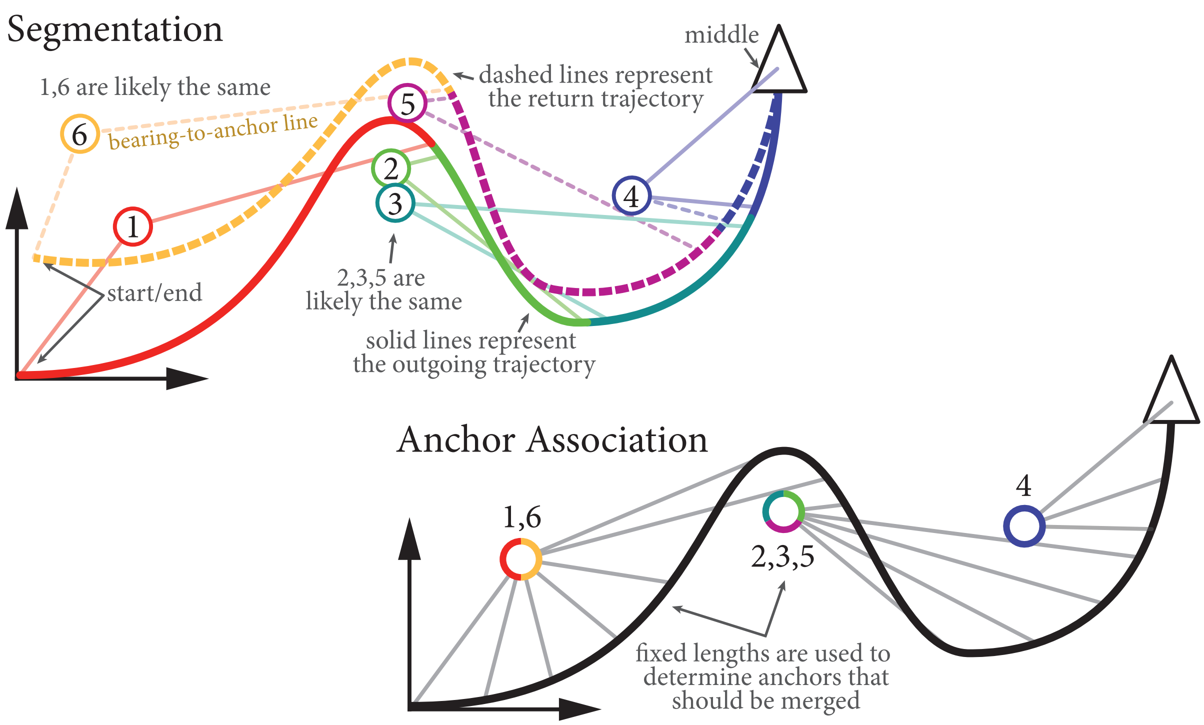


Figure 3.9: *Segmentation and Anchor Association Example*: Segmentation is used to determine sets of robot poses (i.e., subtrajectories) that have a common current anchor (i.e., first along the tether from the robot). The subtrajectories are color coded and the anchors are numbered in order of observation. Due to odometry drift, we detect more anchors than we should. To account for this, we merge anchors that are likely to be the same, which is analogous to loop closure in SLAM.

Anchor Detection:

With respect to anchor detection, the batch method differs from the online approach because we must account for the following situations:

(i) *New Anchor*: Detect the locations of active anchors using all the measurements,

(ii) *Old Anchor*: Detect when anchors are revisited in order to perform loop closure.

We handle (i) by segmenting the robot's trajectory into subtrajectories of poses that share a common anchor as in Figure 3.9. Segmentation of the trajectory is inspired by the 'clique' technique proposed by Howard (2008), who showed the benefit of grouping sets by mutually consistent features. We perform segmentation without considering (ii) and assume that every time the tether touches a new anchor it is a new observation. We handle (ii) after minimizing the distance between anchors with similar fixed tether lengths, which is an indicator that a 'new' anchor is likely a revisited anchor. We handle

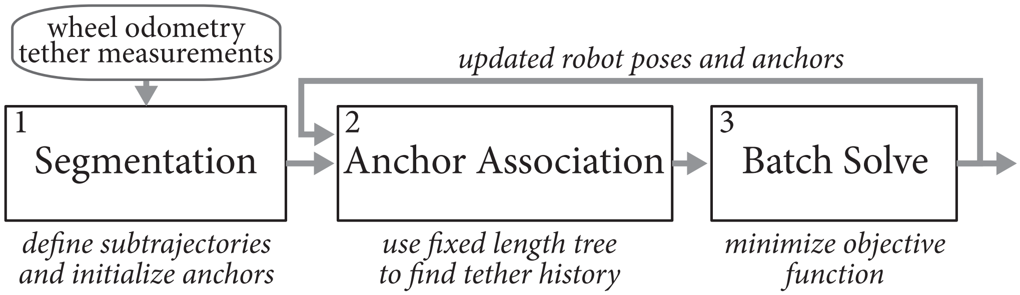


Figure 3.10: *Batch Pipeline*: 1) Segment the robot's trajectory into subtrajectories that share the same current anchor point. Currently, this step happens just once upon initialization because it is computationally expensive. 2) Associate anchors that are likely to be the same observation and use the concept of a fixed length tree to determine the tether interaction history. 3) Find optimal perturbations to the state during the batch solve, which minimize the cost from equation (3.19). Repeat steps 2 and 3 until convergence.

loop closure by merging the position of multiple anchors into a single hypothesis using the concept of switchable constraints from Sünderhauf and Protzel (2012), which we implement as a dynamic-covariance-scaling robust cost function as per Agarwal et al. (2013). Loop closure has the effect of repairing the trajectory to account for dead-reckoned odometry as illustrated by the anchor association step in Figure 3.9. Our batch pipeline is illustrated in Figure 3.10. Further details on the segmentation and association processes will be provided as follows.

Segmentation:

Segmenting the trajectory into poses that share common current anchors is analogous to the discretization process proposed by Teshnizi and Shell (2014) that occurs when a tether (attached to a robot) comes into contact with an obstacle. Segmentation of the robot's trajectory into subtrajectories also allows for detecting the location of candidate anchors before the batch solve begins. First, we make the assumption that the wheel odometry is accurate over short distances and use it as a prior on the robot's trajectory. Then, we perform RANSAC (Fischler and Bolles, 1981) to detect candidate anchor positions using just the bearing-to-anchor measurements and robot poses from odometry. As illustrated by Figure 3.11, the detection is made by selecting two robot poses at random and finding the point where lines drawn along their bearing-to-anchor measurement (i.e., tether arm heading) intersect. Our approach to anchor detection approach is inspired by single-point RANSAC introduced by Scaramuzza et al. (2009). The benefit of this

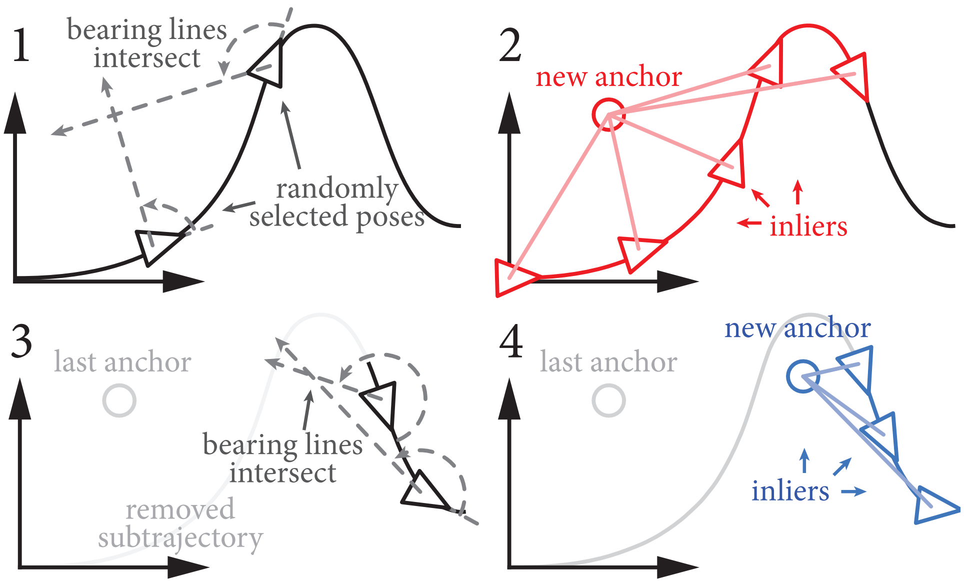


Figure 3.11: *RANSAC Segmentation*: With odometry as a prior on the robot's trajectory, we segment poses into subtrajectories as follows. 1) Select two random poses and determine where lines drawn along their bearing-to-anchor measurement intersect. 2) Candidate anchors with the most number of inliers are selected. 3) Consecutive inliers form a subtrajectory, which are removed from the pool and the process iterates. 4) Segmentation completes when all robot poses are assigned to an anchor and subtrajectory.

anchor detection scheme over the particle-filter approach is that RANSAC (i) is robust to outliers, (ii) doesn't require the use of tether length measurements, and (iii) uses all the measurements to make a decision.

In practice, we select anchor candidates having the greatest number of inliers and associate them to a subtrajectory of robot poses. Then, to improve the result, we perform a local optimization (Chum et al., 2003) to resolve anchor positions using all the measurements from a given subtrajectory. This preprocessing step is analogous to detecting feature tracks in visual odometry. Segmentation is performed only once prior to the batch solve because it is computationally expensive, although, it would be possible to iterate this process as well through future speedups.

Anchor Association:

In batch TSLAM, loop closure is handled by associating and merging multiple anchors that are likely the same. Since odometry drifts over time and is used as a trajectory prior, the segmentation step will initialize anchors in physically different locations even though they are the same (see Figure 3.9). Accordingly, we cannot naively associate and

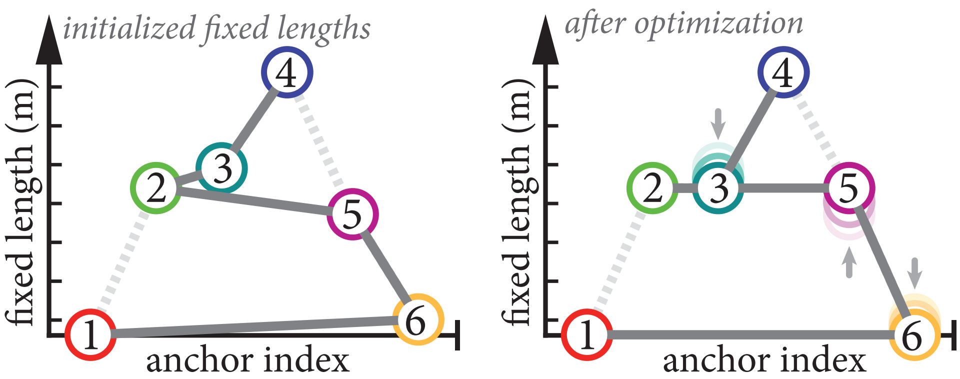


Figure 3.12: *Fixed-Length Tree*: In the left example, fixed lengths have been initialized for each anchor in the example problem from Figure 3.9. The tether history from any anchor back to the initial anchor can be found by descending along the solid line. For the example problem, the history from the fourth anchor is the sequence 4, 3, 2, 5, 6, 1. The right example shows the effect of including fixed lengths in the optimization and merging similar anchors (i.e., matched anchors are horizontally aligned in the tree).

merge anchors that are closely spaced. A better approach is to use the concept of 'fixed' tether length (introduced earlier) to identify anchors that should be merged; even though multiple observations of an anchor can be made in different locations, their associated fixed lengths will be roughly similar. For any subtrajectory and anchor pair, an average fixed length can be initialized without knowledge of the tether history. For an example pose at time k, we can find a single fixed length measurement:

$$d_{\text{fix},k} = d_k - d_{\text{free},k}\,, \tag{3.24}$$

where the average fixed length is the mean for all poses in the given subtrajectory.

Calculating the average fixed length for each anchor allows for determining the tether history from any anchor back to the initial fixed anchor, which is used to fill in the fixed-length model from equation (3.7). In batch, we allow for non-sequential ordering of the tether history because the same anchor can be represented by multiple points. In order to determine the tether history sequence, we use a binary search tree inspired by Bentley (1975), which is populated with fixed tether lengths as shown in Figure 3.12. The illustrated tree accompanies the example from Figure 3.9 and explains how tether history is determined by descending along the tree.

We make one modification to the anchor representation for batch TSLAM to allow

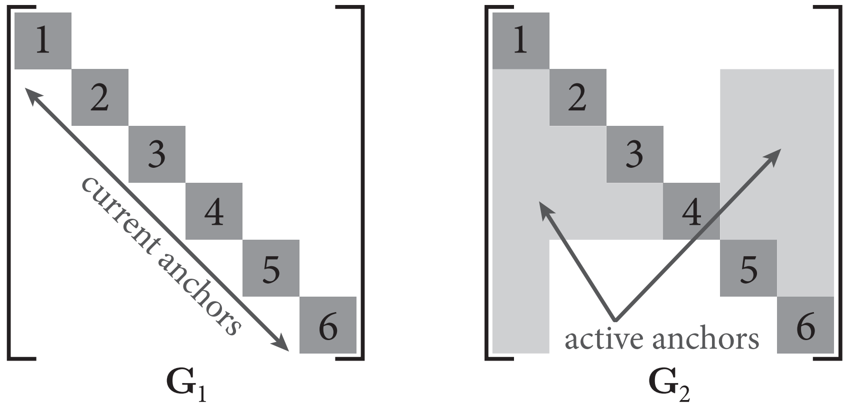


Figure 3.13: *Jacobian Sparsity*: For the example problem from Figure 3.9, $\mathbf{G}_1$ and $\mathbf{G}_2$ are Jacobians of the measurement model with respect to robot poses and anchors respectively. These Jacobians are used to assemble the $\mathbf{A}$ matrix from equation (3.21). $\mathbf{G}_1$ is block diagonal because it involves the robot poses and just the current anchor. $\mathbf{G}_2$ is more dense because it depends on the tether history determined in Figure 3.12.

the fixed length to be optimized as part of the state:

$$\boldsymbol{\ell}_n = \begin{bmatrix} x_n \\ y_n \\ d_{\text{fix},\boldsymbol{\ell}_n} \end{bmatrix} , \tag{3.25}$$

where $d_{\text{fix},\boldsymbol{\ell}_n}$ is the average fixed length for anchor $\boldsymbol{\ell}_n$.

The tether history has an impact on the $\mathbf{A}$ matrix from the linear system we want to solve in equation (3.21), which has components

$$\mathbf{A}_{12} = \mathbf{G}_1^{-T}\mathbf{R}^{-1}\mathbf{G}_2 \ , \quad \mathbf{A}_{22} = \mathbf{G}_2^{-T}\mathbf{R}^{-1}\mathbf{G}_2 \ , \tag{3.26}$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ are the Jacobians of the measurement model with respect to all robot poses and anchor positions respectively, and $\mathbf{R}^{-1}$ is the block-diagonal, inverse covariance on measurement noise. $\mathbf{G}_1$ is still block-diagonal because the Jacobian with respect to robot poses involves just the 'free' length of tether and bearing-to-anchor (see equations (3.5) and (3.8)). Conversely, $\mathbf{G}_2$ depends on the tether history and all active anchors, which impacts the sparsity. For the example problem from Figure 3.9, sparsity is illustrated for $\mathbf{G}_1$ and $\mathbf{G}_2$ in Figure 3.13. Due to the tether history, $\mathbf{G}_2$ is more dense.

In order to perform loop closure through anchor association and merging we make the assumption that (i) anchors positions are static, and (ii) the outgoing and return trajectories are similar. If the robot retraces its outgoing path and unwraps from all but the initial anchor, then all anchors will be reobserved during a traverse, which guarantees loop closure. Accordingly, associated anchors with similar lengths will be 'pulled' together during the optimization, causing the outgoing and return trajectories to align. In practice, fixed lengths, robot poses, and anchor positions are optimized, then later, a robust cost function is introduced between each link in the fixed length tree. The additional cost terms have the effect of merging closely spaced anchors with similar fixed lengths into one place. The robust cost is a switchable constraint (Sünderhauf and Protzel, 2012), implemented as dynamic covariance scaling (Agarwal et al., 2013), which has a narrow region of influence, is robust to outliers, and offers fast convergence (MacTavish and Barfoot, 2015). For example, the influence is negligible for anchors that are far apart, and strong for nearby anchors that can be merged by similar fixed lengths. This robust cost is added as an additional term in the overall optimization problem. For detail on how dynamic covariance scaling is implemented, see Appendix B.1.

The batch approach to data association improves on the particle-filter method because loop closures are not permanent and can be reversed if they reduce the cost function. Alternatively, the particle filter makes sequential associations on-the-fly that cannot be reversed until anchors are removed from the list in the order they were added. This approach causes early mapping mistakes to adversely impact the remaining estimation, which is especially problematic after subsequent resampling events have reduced particle map diversity (i.e., many particles will share the same parent particle and map of anchors over time). We note that an edge case exists for physically distinct anchors that have similar fixed lengths. For example, if the robot completes a traverse and then heads off on a new path and encounters obstacles, then we cannot determine if new anchors have been previously observed without large-scale loop closure. As such, the batch approach does not allow mergers between anchors that have been removed from the active list.

Batch Implementation:

As previously shown in Figure 3.10, the batch method is initialized with odometry and tether bearing-to-anchor measurements in order to segment the trajectory and initialize anchor estimates using RANSAC. Next, robot poses, anchor positions, and fixed lengths are optimized by gradient descent. We use a robust cost function on loop closures to merge anchors with similar fixed lengths. The optimization iterates until a convergence criterion is met. In practice, we use an average change-in-cost threshold over iterations.

3.4 Experiments

A series of experiments were performed, which range from simulation to experiments with TReX in indoor and outdoor environments, to evaluate proposed solutions to TSLAM. We start by evaluating our online and batch approaches on computer-generated data in order to compare the statistical performance of the algorithms over a number of trials. Next, we evaluate our performance on data generated by TReX in a cluttered, indoor environment with ground truth. Our experiment section concludes with a qualitative example of TSLAM working on data collected during an outdoor field test on a slope with trees serving as anchors.

3.4.1 Simulation

Simulated data was generated to create a number of different ground-truth robot trajectories and potential anchor configurations. For each trajectory and map, ground-truth tether and wheel odometry measurements were recorded and zero-mean, Gaussian noise was added. The position and quantity of anchors are unknown to the estimator and only noisy measurements are used. For statistical analysis, the online and batch algorithms were evaluated on 100 trials using the same input data. The localization accuracy per approach is shown as a percentage error after completing the traverse (i.e., the robot detaches from anchors and returns to the start) in order to evaluate solution drift. Both approaches statistically outperform dead-reckoned odometry as we would expect, with the batch method performing best overall. For comparison, we also show the result of running an alternative batch approach, which makes the approximation that measurements are independent (similar to particle-filter TSLAM). As expected, the original batch method outperforms both the alternative and particle-filter approaches by accommodating for measurement dependence. In other words, both the position and uncertainty for all 'active' anchors can be updated at once using the proposed batch method as compared to just the 'current' anchor for other approaches. For a single example from the trial, Figures 3.15 and 3.16 show how each method's position error and trajectory maps compare (the alternative batch method is not shown). Most importantly, both online and batch approaches successfully reduce accumulated error from odometry as loop closures are made and the robot reobserves its initial anchor. For more details on the result from simulation, Appendix B.2 provides a time-lapse animation of the particle filter and an illustration of batch segmentation and association.

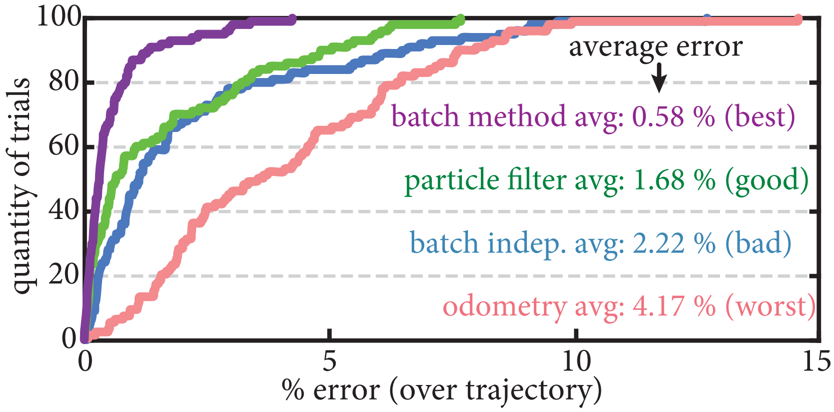


Figure 3.14: *Simulation Statistics*: (100 trials) For all trials, the percentage error over the total trajectory is computed from the RMS error at the end of the robot's traverse. Cumulative distribution functions (CDFs) are used to show the statistical performance of each TSLAM approach with odometry. Each method performs better than odometry as we would expect, with the batch method performing best overall. To visualize the impact of allowing measurements to be dependent, we also include trials from an alternative batch implementation (batch indep.), which makes the approximation that measurements are independent and only allows updates to the 'current' anchor at any given time. As expected, this alternative method performs similar to the particle filter and worse than the proposed batch approach, which properly accounts for uncertainty across all anchors.

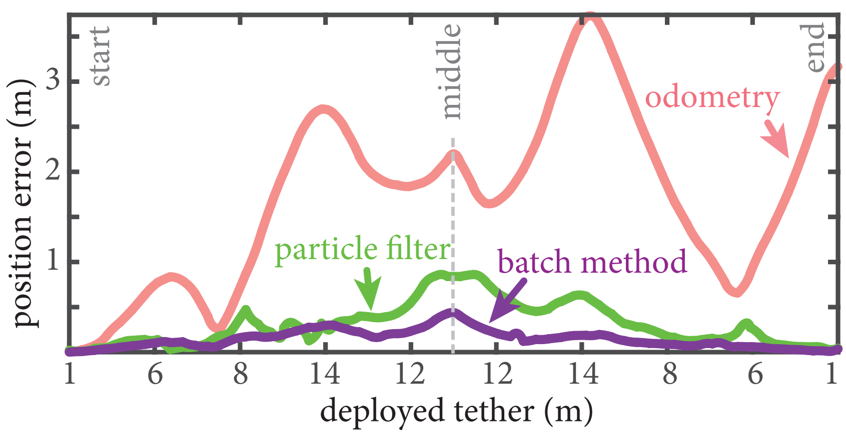


Figure 3.15: *Simulated-Trial Error*: For an example taken from the trial, we compute the RMS position error as a function of deployed tether. Each method successfully reduces dead-reckoned error from odometry as loop closures are made and the initial anchor is reobserved, with the batch method performing best. Most importantly, the error at the end of the traverse approaches zero for each TSLAM estimate. Figure 3.16 illustrates the simulated trajectory and anchor map, which accompany this dataset and plot.

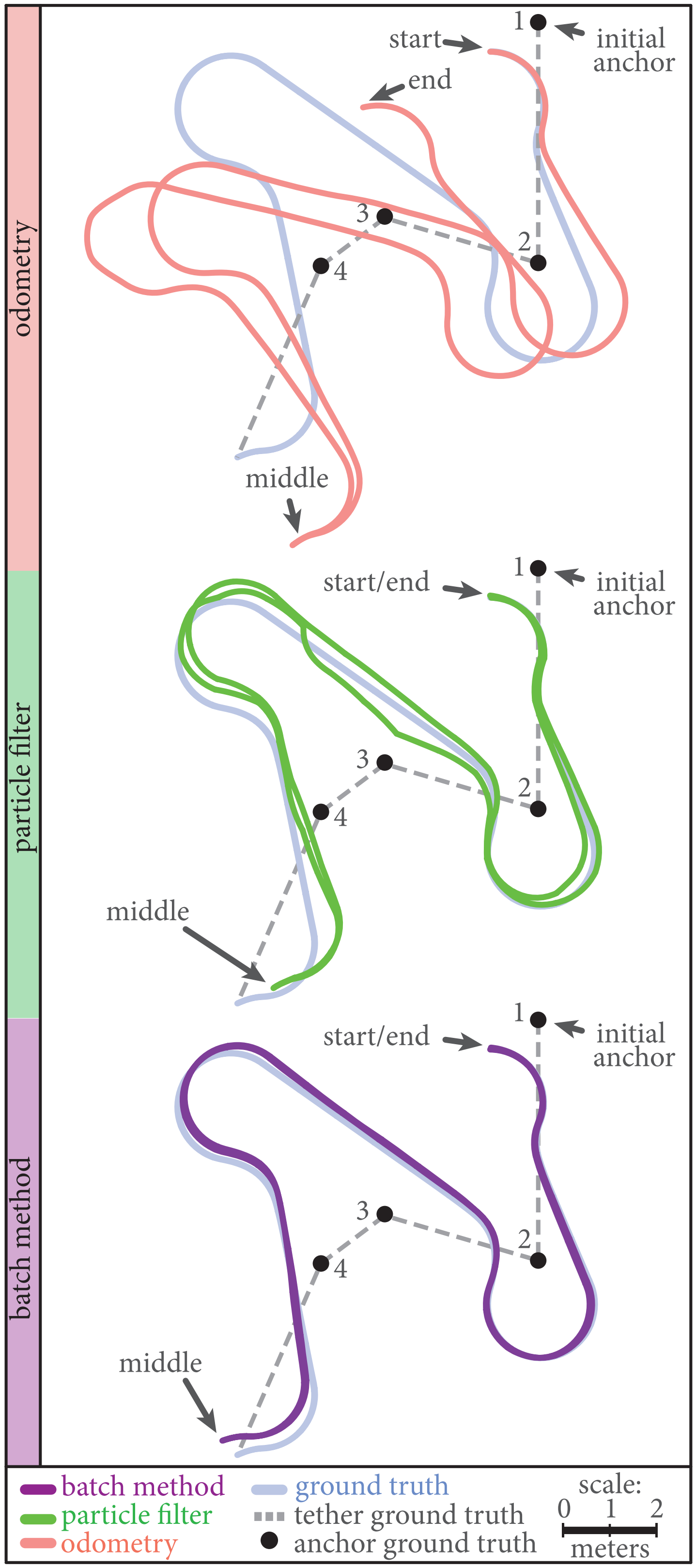


Figure 3.16: *Simulated-Trial Trajectory Maps*: For the example trial from Figure 3.16, the trajectories for each TSLAM method are compared to odometry and ground truth. This visual example shows how dead-reckoned odometry can be drastically improved by incorporating tether measurements. The batch method performs best overall.

3.4.2 Indoor Experiment

Three different experiments were conducted indoors with TReX on a flat surface, where bollards (yellow posts) serve as potential anchors, as shown in Figure 3.17. The robot was manually driven along different trajectories to test a variety of anchor configurations, while the tether was kept taut by an on-board tension controller. The test setup allowed for capturing the ground-truth positions of both the static bollards and a marker attached to the center of the robot using a surveying device (Leica Nova MS50 MultiStation).

3.4.3 Indoor Results

For each experiment, results are provided in the same fashion as in Section 3.4.1. Figure 3.18 plots position error as a function of deployed tether, where each TSLAM approach is compared to both dead-reckoned odometry and ground truth. Figure 3.19 illustrates the position error using a set of trajectory maps, which qualitatively show how each estimate compares to ground truth. Overall, both TSLAM solutions are able to reduce dead-reckoned odometry error once loop closures (i.e., anchor associations) are made. In terms of position error, the batch method performs marginally better than the particle filter, but, as we will see in Section 3.4.4, provides a noticeably better mapping result. One of the issues that makes the flat, indoor environment a challenge is that wheel odometry is made worse due to influence from a taut tether; any 'pulling' on the robot can result in non-uniform wheel slip, which violates the robot's kinematic model used in the TSLAM formulation (sliding is not observable). Fortunately, TSLAM can leverage tether measurements to overcome this issue and repair the trajectory. As before, supplementary results are available in Appendix B.2.

3.4.4 Indoor Mapping

Figure 3.20 provides a qualitative comparison of TSLAM anchor mapping performance to ground truth for each indoor experiment. As opposed to the robot localization results from Section 3.4.3, the batch method demonstrably outperforms the particle-filter anchor map estimate and best approximates ground truth. The particle-filter approach produces a poorer mapping result because (i) the ellipse measurement model is not robust to outlier measurements, (ii) sample impoverishment over successive resample events will cause a lack of map diversity, which means that mapping mistakes cannot be fixed until anchors are removed, and (iii) simply adding more particles to represent the state will be

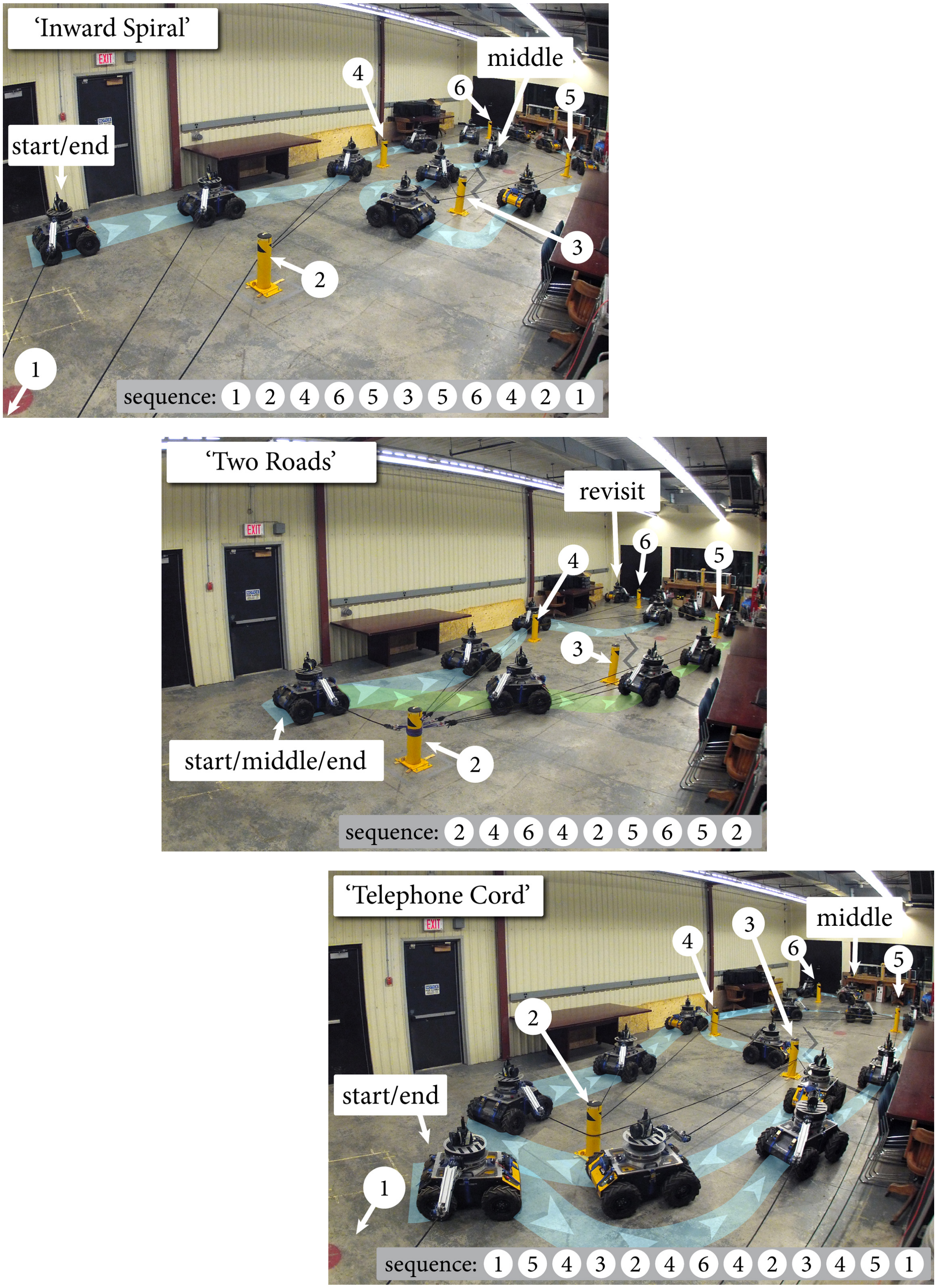


Figure 3.17: *Indoor Setup*: The setup for each indoor experiment is shown using a time-lapse image, which indicates the robot's outgoing trajectory. Anchors are highlighted with markers and their sequence is indicated in the lower right corner.

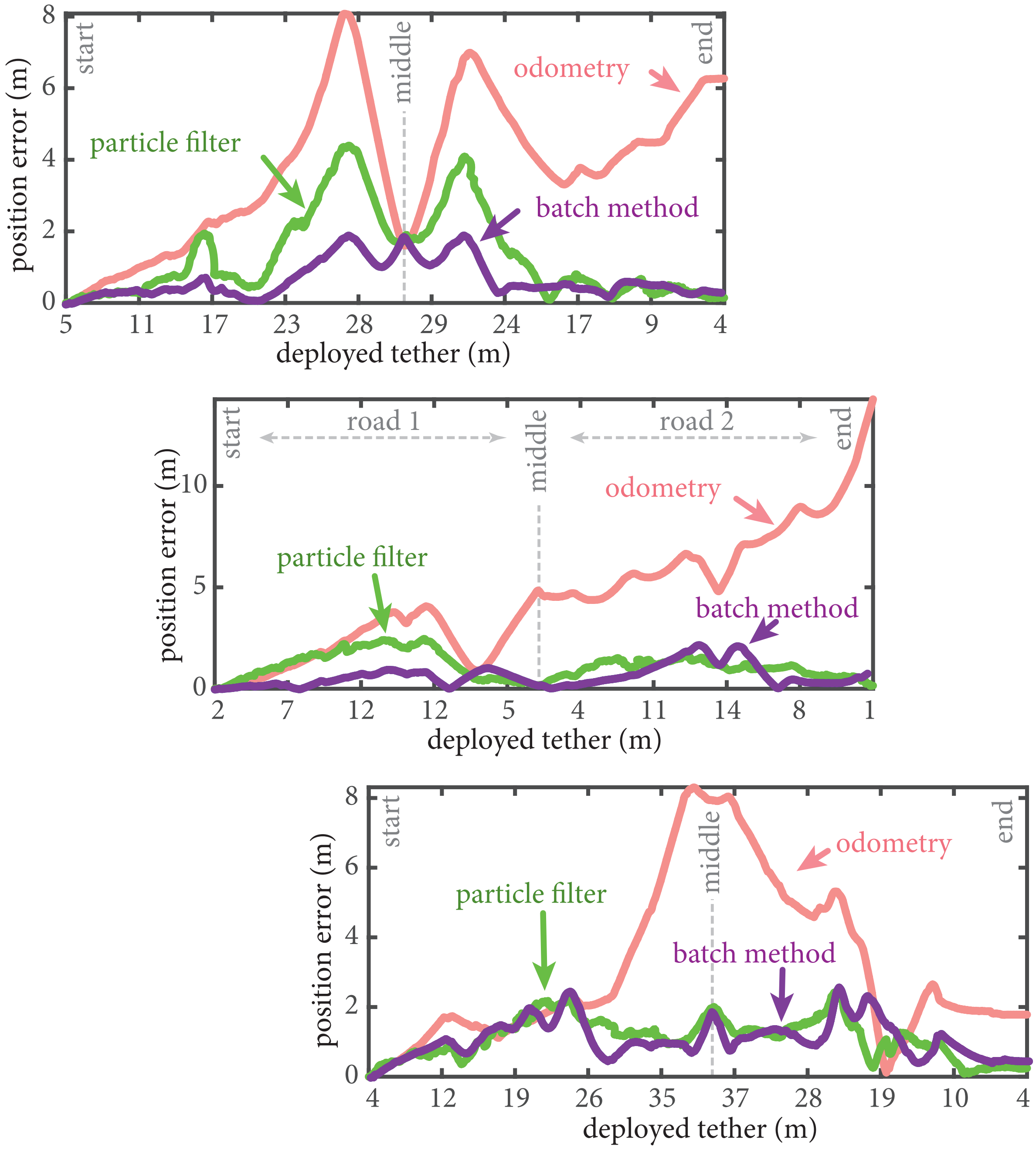


Figure 3.18: *Indoor Position Error*: RMS position errors are shown with respect to deployed tether for three different trajectories. As in simulation, both TSLAM solutions reduce dead-reckoned error due to loop closure (i.e., anchor association). The batch method performs marginally better than the particle filter in the above examples. However, batch clearly performs best in terms of anchor mapping, as shown in Figure 3.20. We note that there are occasions that odometry can serendipitously cross the ground-truth trajectory, which happens in all three tests. However, odometry will generally drift away from the solution given enough integration time.

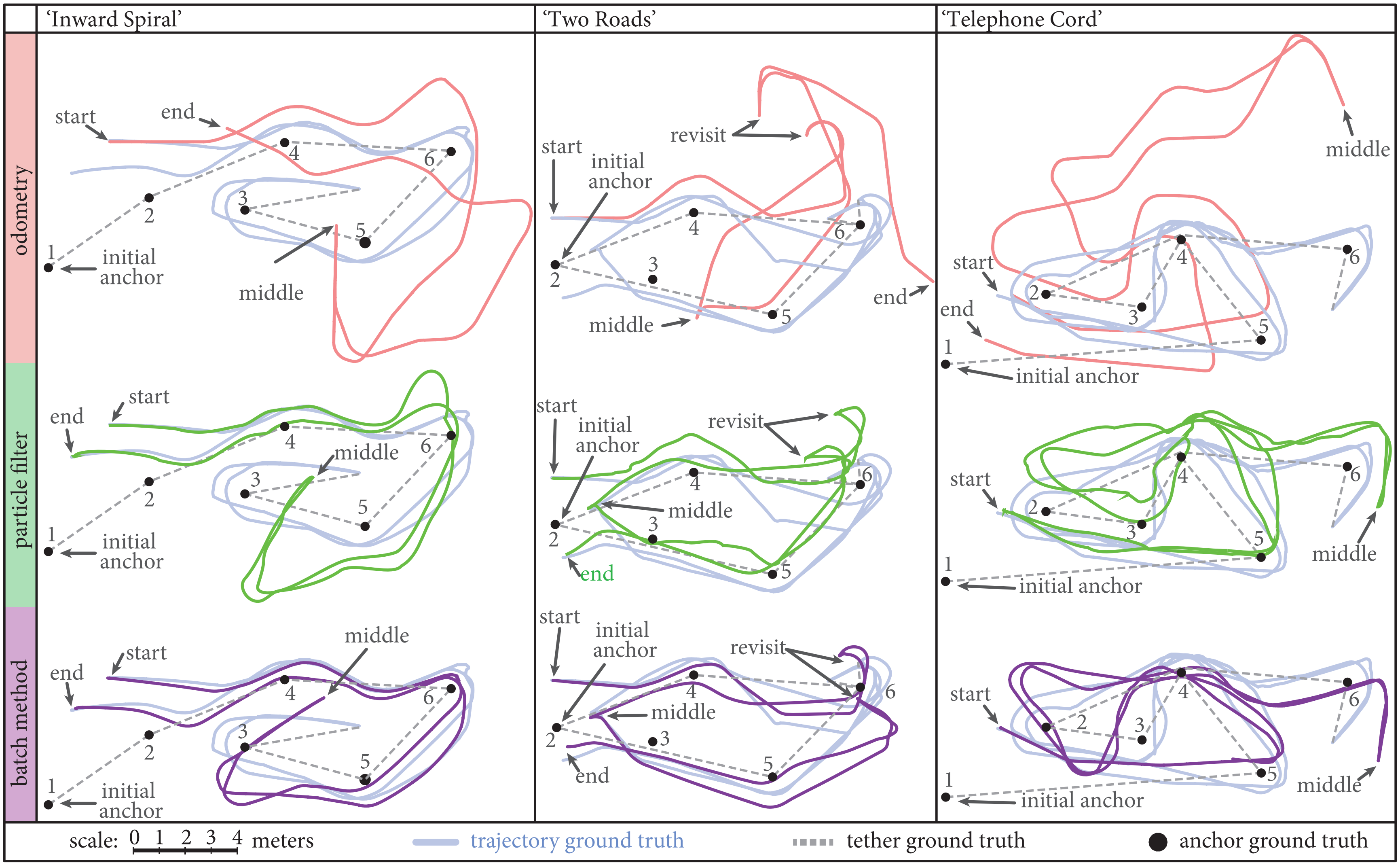


Figure 3.19: *Trajectory Maps*: Localization results from the indoor experiment show that both TSLAM methods outperform odometry. Each method is able to reduce accumulated error from dead-reckoned wheel odometry upon revisiting the initial anchor point due to our use of tether measurements. In general, the batch method's estimated trajectory better matches ground truth than the particle filter.

inefficient, or worse, intractable. Conversely, the batch method benefits from using all the measurements at once and is robust to outliers, which allows for a better approximation of the true posterior. We will now discuss the mapping outcome of each test as follows.

'Inward Spiral'

The robot's close trajectory past anchor 2 results in a negligible offset to the expected tether measurement, which causes a missed detection for each TSLAM approach; missed detections can happen if anchors are positioned in near collinearity, as anchor 1, 2, and 4 are. *Particle-Filter Method*: An early mapping mistake was made in the positioning of anchor 4, which causes the remaining map to misalign and results in additional false detections. *Batch Method*: This map best resembles ground truth with fewer false detections, but suffers from drift in the 'spiral' section of the map.

'Two Roads'

Two different paths or roads are taken to reach anchor 6. In between the paths, the robot unwraps from anchors and reobserves its initial anchor (anchor 2 in this case). *Particle-Filter Method*: A cluster of false detections near the start of the path result in a closer spacing of anchors than is observed in ground truth. *Batch Method*: We notice a similar false detection in the same position as observed in the prior experiment, which occurs in a location that the robot briefly rotates in place; the detection is likely the result of a biasing of the robot's bearing-to-anchor measurement as tension is reduced in the tether due to wheel slip. We note that the map could be improved if we were to merge multiple observations of anchor 6, but (as previously mentioned) we do not allow fixed length associations to anchors that have been removed from the active list.

'Telephone Cord'

While navigating the most complicated indoor trajectory, anchor 4 was observed four times between outgoing and return trajectories. *Particle-Filter Method*: Many false detections early on in the traverse result in biased detections later on and, overall, a poor mapping performance. *Batch Method*: Again, because we do not merge all the observations of anchor 4 during the outgoing traverse, the resulting map is offset from ground truth. However, the accuracy of the nonvisual, batch approach is impressive considering the challenging trajectory, noisy measurements, and number of anchors encountered.

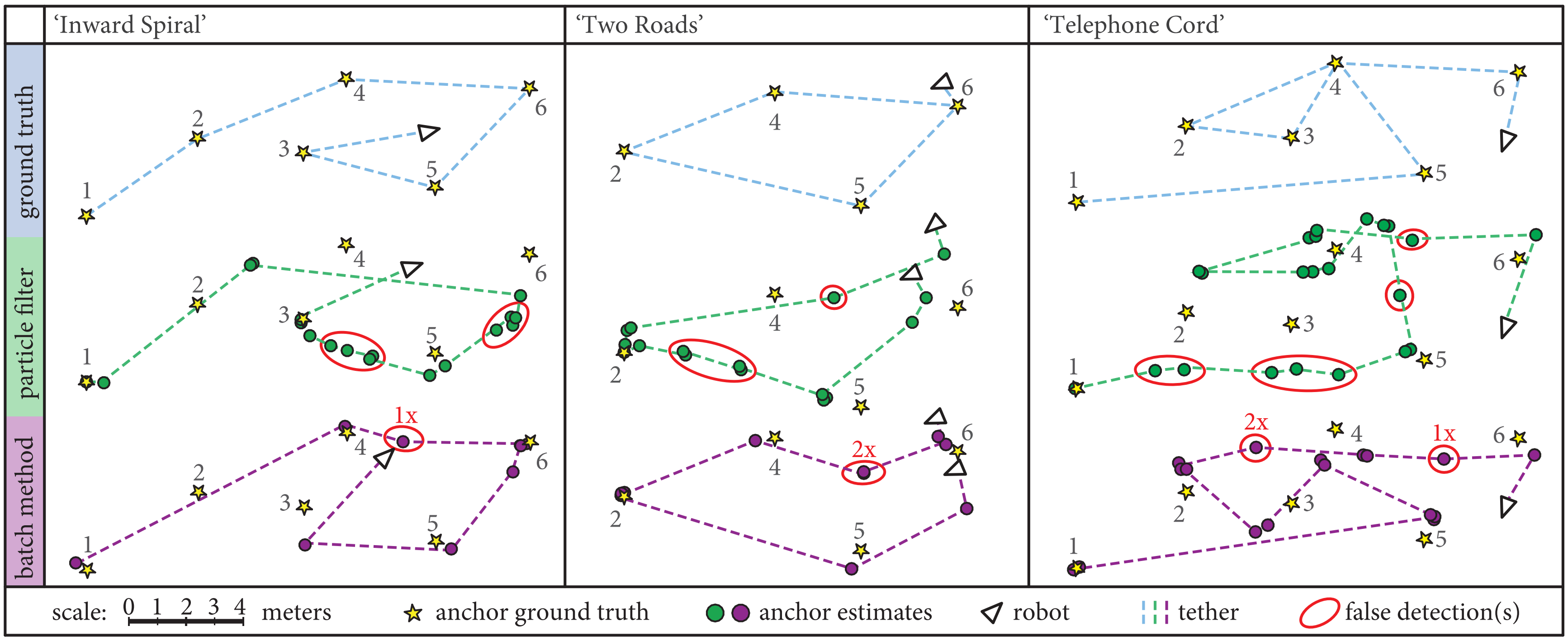


Figure 3.20: *Mapping Comparison*: Here we compare the anchor mapping results for each TSLAM approach to ground truth. The illustrated anchor maps with overlaid tether orientation are intended for a qualitative comparison to ground truth; the comparison is analogous to comparing constellations of points or stars. The main takeaway is that the batch approach best approximates the ground-truth map, having fewer false detections than the particle filter overall. False detections are instances when anchors are placed in locations without a ground-truth anchor and are determined by manual observation.

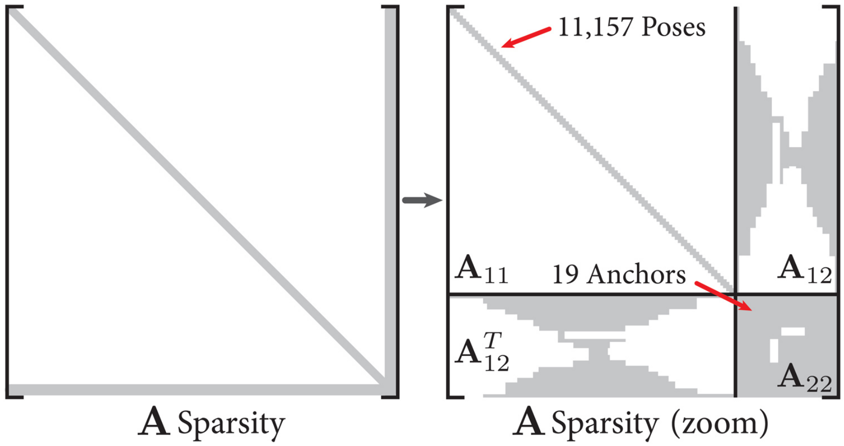


Figure 3.21: *'Telephone Cord' Sparsity*: The *left* plot shows the primary sparsity pattern for the $\mathbf{A}$ matrix used to solve the batch, 'Telephone Cord' estimation. For clarity, the *right* plot provides a non-uniform, zoomed view of the sparsity pattern, which shows that over 11 thousand poses were computed compared to just 19 anchor estimates. A total of 19 anchors were initialized to represent 6 ground-truth anchors. Even though $\mathbf{A}_{22}$ is mostly dense, we can efficiently solve the problem by exploiting the block-tridiagonal nature of $\mathbf{A}_{11}$. We note that it may be possible to further exploit the hourglass pattern of $\mathbf{A}_{12}$ and $\mathbf{A}_{12}^T$, but that is beyond the scope of this work.

Indoor Mapping Accuracy:

Given the success of the batch mapping approach, we provide a quantitative evaluation of mapping error in the following table. Mapping errors per experiment are computed as the average position error for all detections of a ground truth anchor, where associations are made by proximity. The mean error, μ, and standard deviation, σ, are given in meters, while the ratio of true, estimated, and falsely detected anchors is provided; false detections (red circles in Figure 3.20) are not included in the error calculation. The mean mapping error for all experiments is in the range of a meter, which, in spite of the nonvisual approach, would be acceptable for safe navigation.

Table 3.1: Batch method mapping accuracy.

Dataset	μ (m)	σ (m)	true	est.	false
'Inward Spiral'	0.63	0.45	6	11	1
'Two Roads'	0.68	0.41	4	11	2
'Telephone Cord'	0.88	0.44	6	19	3

Computational Cost:

To compare the computational cost between each TSLAM approach, we consider the 'Telephone Cord' experiment, which required 15 minutes (900 seconds) to complete. *Particle-Filter Method*: It took 24 minutes (1,440 seconds) to run the particle filter on this dataset. While the compute time is greater than that experiment duration, we expect that the approach could be run online if the algorithm were implemented in a more efficient, compiled language instead of *MATLAB*. *Batch Method*: The primary sparsity pattern for the **A** matrix, which is used in the 'Telephone Cord' optimization, is illustrated in Figure 3.21. Even though the measurement contribution is dense, the number of anchors relative to poses is small. Thus, we can still solve the problem efficiently by exploiting the block-tridiagonal nature of pose estimates using the sparse Cholesky decomposition from equation (3.23). The batch method iterates to find a solution, where each iteration takes 0.2 seconds. The batch solution converges on a solution to the 'Telephone Cord' trajectory and map in only 40 seconds, which includes the segmentation step. However, the algorithm is also prototyped in *MATLAB*, so it is possible that a solution could be found faster if implemented in another language. It would also be possible, through future adaption of the algorithm, to perform batch TSLAM as a sliding-window, incremental estimation similar to work by Kaess et al. (2008, 2011).

3.4.5 Outdoor Experiment

We evaluate TSLAM in an outdoor experiment to qualitatively demonstrate that our approach can work in a relevant environment. During the field test, TReX was manually piloted on a cluttered, forest slope through trees, which serve as potential anchors. The test setup did not allow for ground truth to be collected, so instead, we overlay estimated trajectories from odometry and TSLAM on an aerial image of the test site. Figure 3.22 shows this trajectory map with anchor estimates from the superior batch approach and provides images collected from a camera on TReX that show different types of anchors being added. The main result is that each TSLAM approach allows the outgoing and return trajectories to be 'pulled' together as loop closures are made. Conversely, wheel odometry exhibits characteristic drift (i.e., the robot start and end positions are offset from one another). Path alignment is used as a qualitative indicator that TSLAM is working, since we expect the robot's outgoing and return trajectories to be similar, so as to avoid entanglement. For clarity, just the batch map of anchors is shown, where anchors

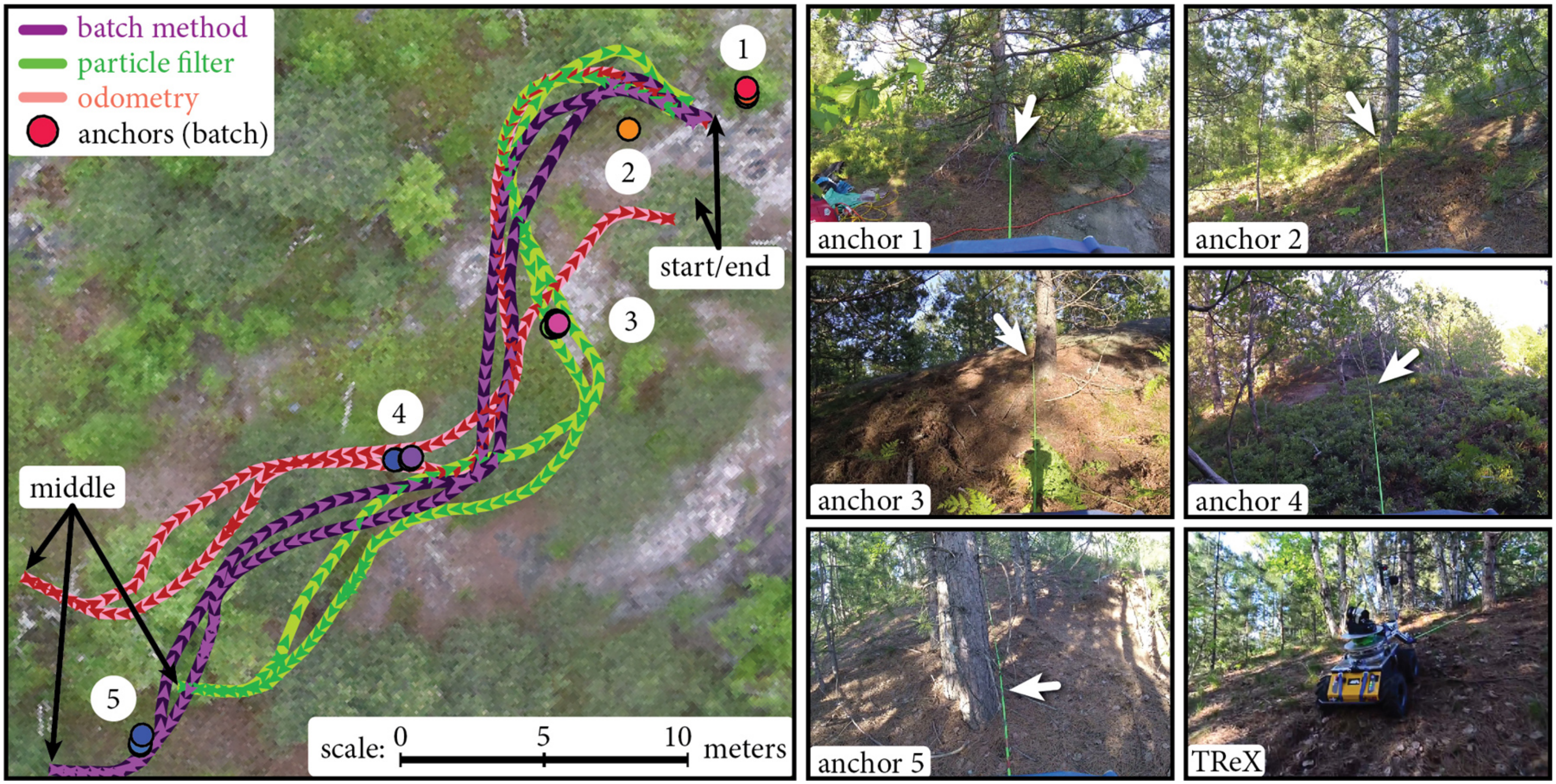


Figure 3.22: *TSLAM Outdoors*: To show that TSLAM can work in outdoor environments, we evaluate each approach with data collected from TReX on a cluttered, forest slope. The *left* image shows an aerial image, onto which, we have superimposed a trajectory map. The anchor sequence (1,2,3,4,5) is numbered according to observation during descent, with 1 being the initial anchor. Although we do not have ground truth, we observe that our estimated trajectories (see direction arrows) are overlapping while odometry has drifted (as we would expect). We show the anchors detected by the batch method for reference. The collection of images on the *right* come from a camera mounted in the direction of the tether's bearing-to-anchor measurement, which shows the types of anchors (i.e., trees) TReX encountered.

have been successfully initialized for each example image collected by a camera on TReX. TSLAM works in this environment because the slope has a gradual inclination ($\sim 20°$), which provides a roughly planar surface for our 2D formulation to work in. For the approach to work in more extreme environments, additional sensors would be necessary to constrain the pose and position of the robot and anchors in 3D. The current approach is the best we can do with the minimal set of nonvisual measurements.

3.5 Lessons Learned

Running our TSLAM algorithms on real data collected by TReX has demonstrated a number of outstanding limitations and problems that are worth discussing as follows.

Assumptions: As formulated, we assume that the tether is perfectly taut and that anchors can be represented as zero-radius points. The tautness assumption ignores the fact that a stretched cable/tether will always exhibit some catenary curve regardless of the tension, which contributes to the length measurement. Dealing with this problem would be difficult because the curve would need to be modeled and the tension between each anchor would need to be estimated. The anchor-point assumption simplifies the representation of large obstacles to a series of points. However, if the tether fully wraps around a small obstacle (e.g., a bollard from our indoor experiment or a tree trunk outdoors), that extra length is not accounted for in our formulation.

Length drift: TReX measures tether length using a non-absolute, rotatory encoder, which can drift during instances of tether slippage at the point of measurement. In order to determine the expected drift of this sensor, we recorded the length after deploying the full tether and found that the noise was $\pm$ 0.05 m for every 45 m of tether deployed (in a single direction). However, drift causes unbounded error accumulation as the robot drives, changes direction, and deploys more tether. Currently, neither TSLAM approach explicitly accounts for drift, which can result in problems with anchor initialization by the particle filter and anchor association during batch loop closure.

Bearing-to-anchor bias: The tether's bearing-to-anchor is measured as the angle between a passively rotating tether arm and the robot body. Since the tether arm is mounted to a mechanical bearing, stiction (i.e., static friction) must be overcome before the arm begins to align in the direction of applied tension. The resulting measurement bias cannot easily be calibrated or measured because it is a function of tension, stiction, wheel slip, and terrain slope. Bias is reduced with increasing slope because stiction is easier to overcome when tether tension is greater. Bias is strongest when the terrain is flat and smooth, as it is in our indoor experiment, because tension must be reduced to prevent unwanted wheel slip. In practice, we assume the measurement noise to be minimal (e.g., within a degree) and allow the batch method to handle bias by using RANSAC to reject outlier measurements, which is why it makes fewer false detections than the particle filter. However, a better understanding of the measurement noise and bias is still needed to adapt TSLAM to more complicated scenarios.

Small-scale loop closure: It is possible that closely spaced anchors with similar fixed lengths will be pulled on top of one another once associated. This action can be problematic when a series of anchors representing a larger obstacle (e.g., a large tree trunk) are pulled together, because we lose useful information about the physical shape of obstacles in the environment.

Large-scale loop closure: Neither TSLAM approach allows for large-scale loop closures to be made for a reobserved anchor that has been previously removed from the active list (i.e., anchors currently in contact with the tether). Since we formulate the problem using nonvisual measurements, place recognition is not easily achieved. Thus, in an example like the indoor 'Two Roads' dataset from Figure 3.19, we do not allow multiple observations of anchor 6 to be associated and merged, even though the result would improve our estimate. A naive solution to the problem, in the context of the batch approach, would be to allow loop closures between any nearby anchors. However, this can result in false associations and could increase mapping and localization errors.

Dynamic anchors: One of the fundamental issues with the proposed TSLAM approach, and in tethered mobility in general, is the assumption that anchors are static, or fixed in the environment. In reality, it possible that an anchor can either shift (i.e., a tree bending under tension) or be removed entirely. In TSLAM anchor change is unobservable as the tether length and bearing-to-anchor are not enough to detect how an upstream anchor has changed. Unfortunately, the dynamic anchor problem is not easily solved and may require other sensor modalities in future formulations of TSLAM.

3.6 Conclusions and Future Work

In spite of the challenges outlined in Section 3.5, we successfully demonstrate that TSLAM is still solvable using a minimal set of nonvisual tether and odometry measurements. We have shown two approaches to the TSLAM problem using both an online particle fil-

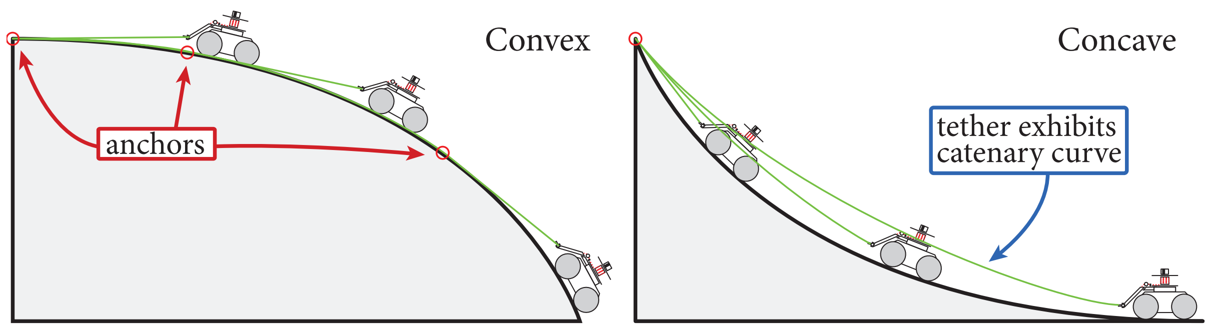


Figure 3.23: *Extending TSLAM to 3D*: These illustrations highlight the difficulty of extending TSLAM to 3D. The *left* example shows TReX navigating on a convex surface, where anchors are added on the ground as the robot drives. In this case, the tether measurements are changing too gradually to detect discrete anchors. The *right* example shows a concave surface, where only a single anchor is present. This is an example of when the perfectly taut tether assumption fails; the mass of the tether causes it to bend, forming a catenary curve. The curved tether's length is a function of the total distance traveled, the topography, the disparity in altitude between the robot and anchor, tether tension, and tether mass, making it difficult to leverage in 3D TSLAM.

ter and an efficient batch formulation. Our indoor and outdoor experiments validate each approach by demonstrating that obstacle-to-tether contacts can be detected for the purpose of tethered navigation in complex, cluttered environments.

Given the performance gains of the batch approach over the particle filter, a future implementation of TSLAM could be posed as a sliding-window batch problem, which would allow for the solution to be updated online as new measurements arrive (Kaess et al., 2008, 2011). It would also be important to extend TSLAM to work in non-planar, 3D environments by incorporating additional information, such as inertial measurements of the robot with respect to gravity and tether pitch, to estimate the 3D pose and position of the robot and anchors. However, 3D environments offer new challenges. Consider the illustration shown in Figure 3.23, which shows how the tether would react in different example terrains. In the convex case, the tether is gradually anchored or laid on the ground, which is difficult to discern given the tether sensors we are using. In the concave case, the tether violates the tautness assumption made previously, which makes it harder to rely on tether length. Adapting TSLAM to 3D would also require an updated representation for anchors, which allows the structure (i.e., radius or shape) of the anchor to be estimated. Furthermore, dynamic anchors will need to be addressed by incorporating additional measurements. Although TSLAM is an interesting problem that can be solved nonvisually, the most robust solution to anchor mapping may involve a fusion of TSLAM with camera or lidar-based SLAM to assist when the environment is dark and dusty.

3.7 Novel Contributions

This chapter has introduced several novel contributions, which are the (i) formulation of the general TSLAM problem, (ii) proposed online and batch solutions to the problem, and (iii) a comparison of each approach on indoor and outdoor datasets.

3.8 Associated Publications

- McGarey et al. (2016). The Line Leading the Blind: Towards Nonvisual Localization and Mapping for Tethered Mobile Robots. In the *2016 IEEE International Conference on Robotics and Automation (ICRA)*.

- McGarey et al. (2017a). TSLAM: Tethered Simultaneous Localization and Mapping for Mobile Robots. In the *International Journal of Robotics Research (IJRR)*.

3.9 Associated Videos

- TSLAM (Particle Filter): https://youtu.be/7ehPxdtYWrA
- TSLAM (Batch Method): https://youtu.be/mzlHJEa3z3Y

Chapter 4

Visual Route Following for TReX

4.1 Motivation

In Chapter 3, we took the approach of mapping tether-to-obstacle anchors using TSLAM as a means to prevent tether entanglement when navigating through cluttered environments. Now, we explore the idea that our robot can unwrap its tether from intermediate anchors by simply retracing its outgoing path. Accordingly, we investigate how a camera-based, visual, route following algorithm can be used on a tethered robot to autonomously repeat a manually-taught path on extreme terrain, detach its tether from obstacles, and return to a safe starting location. Specifically, we must address challenges in tether tension control that prevent route following algorithms from working for tethered robots: the tether shall (i) remain taut regardless of inclination, (ii) not restrict the robot's motion, and (iii) provide motion assistance on steep terrain when wheel traction is reduced.

To accomplish route following, we use the Visual Teach & Repeat (VT&R) algorithm originally from Furgale and Barfoot (2010), which constructs a topometric map of a manually driven path, localizes against that map during an autonomous repeat, and tracks the path by converting perceived localization errors into robot velocity commands. The path tracker relies on a kinematic model of the robot, which, in the out-of-the-box version of VT&R, does not account for tether disturbances and non-uniform wheel slip. In order to avoid changes to the underlying VT&R algorithm, we need a tether controller that allows the robot to drive and track a path as if it were untethered and assists on steep slopes when traction is reduced. To that end, we first formulate a novel tether-control strategy[1], which accounts for commanded robot motions from the path

[1]This work was done in conjunction with Max Polzin from ETH Zurich – see McGarey et al. (2017b).

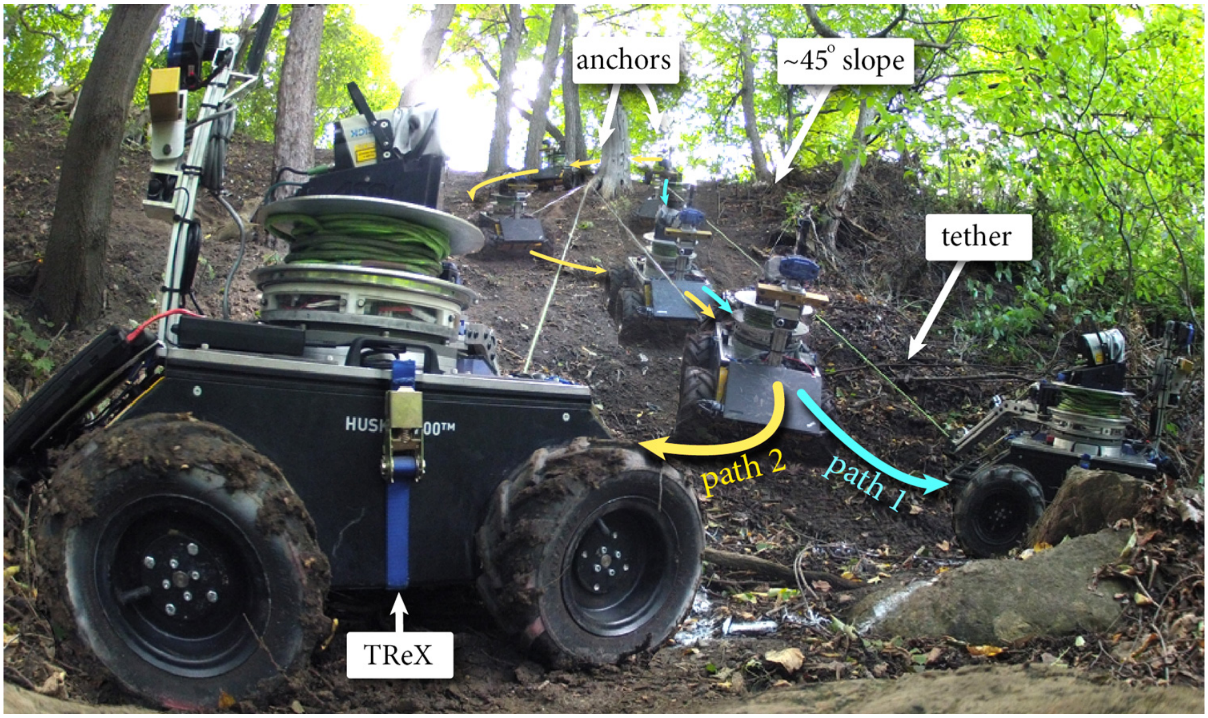


Figure 4.1: *TReX Navigates Steep Terrain*: This time-lapse image highlights visual, route following experiments on steep terrain with TReX. Due to added anchors around trees, the robot must retrace each path to safely return and prevent tether entanglement.

tracker and preserves an appropriate tether tension on any slope. Our experiments with tethered VT&R in cluttered indoor and outdoor environments successfully demonstrate autonomous route following for the TReX robot. Figure 4.1 shows an example time-lapse image from the outdoor experiment on steep terrain.

This chapter proceeds as follows. Section 4.2 examines prior work on visual route following and tether control. Section 4.3 proposes the tether-control strategy. Section 4.4 explains the experiment and provides analysis on path-tracking performance. Section 4.5 offers concluding remarks and future extensions to tethered VT&R, while Section 4.6 states novel contributions.

4.2 Related Work

Camera-based, visual route following is a form of autonomy that enables a mobile robot to safely drive a predetermined path between a start and end goal (Choset, 2005). A common approach to visual navigation will leverage appearance-based, topometric maps, which are built of locally metric, overlapping submaps, and a path-tracking controller, which converts localization error into robot velocity commands (Furgale and Barfoot, 2010; Booij et al., 2007; Zhang and Kleeman, 2009). Bonin-Font et al. (2008) offers a survey of common approaches to visual route following. We are interested in using Visual Teach & Repeat (VT&R), from Furgale and Barfoot (2010), due to its extensive testing in unstructured, outdoor environments. In particular, our experiments leverage multi-experience VT&R from Paton et al. (2016), which utilizes a collection of past experiences to better localize when lighting and scene appearance change over time.

Our problem is complicated by the fact that route following algorithms are not adapted to tethered robots; autonomous driving actions must be paired with appropriate tether actions so that the tether remains taut while also allowing the robot to move as commanded. Tether control theory has been well explored for use in space and underwater environments (Rupp, 1975; Bainum and Kumar, 1980; Yoerger and Slotine, 1985; Nohmi, 2004). However, these works are not applicable to robotic vehicles that require a consistently taut tether for support on steep terrain. Although a variety of tethered robotic systems have been developed in the past (Apostolopoulos and Bares, 1995; Krishna et al., 1997; Huntsberger et al., 2004; Ahn et al., 2006; Iqbal et al., 2008; Abad-Manterola, 2012), none have explicitly proposed robust tether-control strategies. Only one example of visual route following for tethered robots exists in the literature; Tsai et al. (2013) developed a system to allow for short-range ($\sim 4\,m$), autonomous docking of a tethered robot on *flat* ground. This approach relies on camera-to-fiducial tracking and uses a model-predictive, feed-forward controller, which does not account for tether tension (i.e., feedback control). In the following section, we will introduce a tether-control strategy that allows for autonomous route traversal in a variety of environments.

4.3 Methodology

This section will provide a high-level overview of the VT&R algorithm and explain our tether-control strategy, which is used to make VT&R work for a tethered robot. We also look into the forces acting on TReX as a means to determine the robot configurations and environmental conditions that result in excessive wheel slip and cause VT&R to fail.

4.3.1 Visual Teach & Repeat (VT&R)

VT&R is an algorithm that has enabled autonomous route following for a number of mobile robots operating outdoors (Furgale and Barfoot, 2010; McManus et al., 2013; Paton

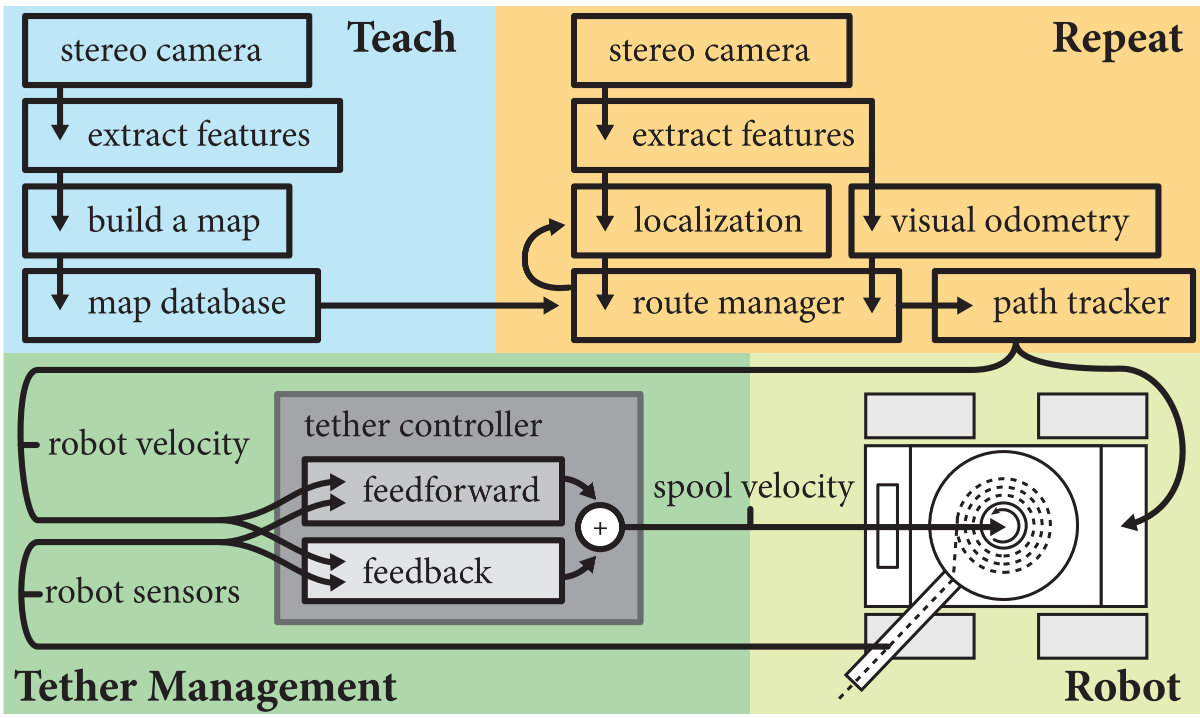


Figure 4.2: *Tethered VT&R*: A tethered mobile robot equipped with a camera is manually driven along a path that is determined to be safe. Visual features are detected and used in building a topometric map, which is localized against during an autonomous repeat of the desired path. Localization errors are converted to commanded wheel velocities by a path tracker. Our tether controller accounts for the desired motion of the robot while managing a taut tether using on-board sensors. The controller allows VT&R to work for tethered robots on extreme terrain, without modification to the VT&R algorithm. In practice, the robot (TReX) is manually piloted down steep terrain and later commanded to autonomously ascend and unwrap from added anchors, which prevents tether entanglement.

et al., 2016). During a teach phase, a mobile robot equipped with a stereo camera (or other visual sensor) is manually driven on a path that a human operator has determined to be safe. The camera outputs a series of paired images, which are used by a feature detector to determine the 3D position of objects in the environment. The 3D features are detected throughout the teach phase and are used to assemble a topometric map, which is structured as a spatio-temporal pose graph. The graph uses vertices to store matched features and edges to store relative transformations. When the robot is commanded to autonomously repeat the path, new features are detected and matched to those stored in the map to localize the robot along the path. Sometimes the localization will fail and visual odometry, which is reasonably accurate over short distances, will be relied on until features are matched again. Localization errors from the robot's pose and trajectory (with respect to the desired path) are converted into velocity commands, which steer the vehicle along the repeat path to minimize error. The path tracker is reliant upon a kinematic model of the robot, which converts velocities into desired vehicle movement. We want to use VT&R without changing this model to account for non-uniform forces from the tether. Therefore, we take the straightforward approach of developing a tether controller that allows the robot to drive as if untethered. Figure 4.2 illustrates how we intend to adapt VT&R for tethered systems.

4.3.2 Forces acting on TReX

The forces acting on our tethered robot, TReX, are illustrated in Figure 4.3. We assume the robot is driving at a constant velocity on a surface with inclination

$$\alpha \in \left[0, \frac{\pi}{2}\right]. \tag{4.1}$$

The force due to gravity, F_g has in-plane and normal-to-the-plane components,

$$F_{g\|} = F_g \sin(\alpha), \quad F_{g\perp} = F_g \cos(\alpha), \tag{4.2}$$

which depend on α. By controlling the robot's tether spool, we can leverage the force due to tension, F_t, from the anchored tether to offset F_g on steep surfaces, which has the beneficial result of reducing the influence of $F_{g\|}$. F_t can also be divided into components,

$$F_{t\|} = F_t \cos(\beta), \quad F_{t\perp} = F_t \sin(\beta), \tag{4.3}$$

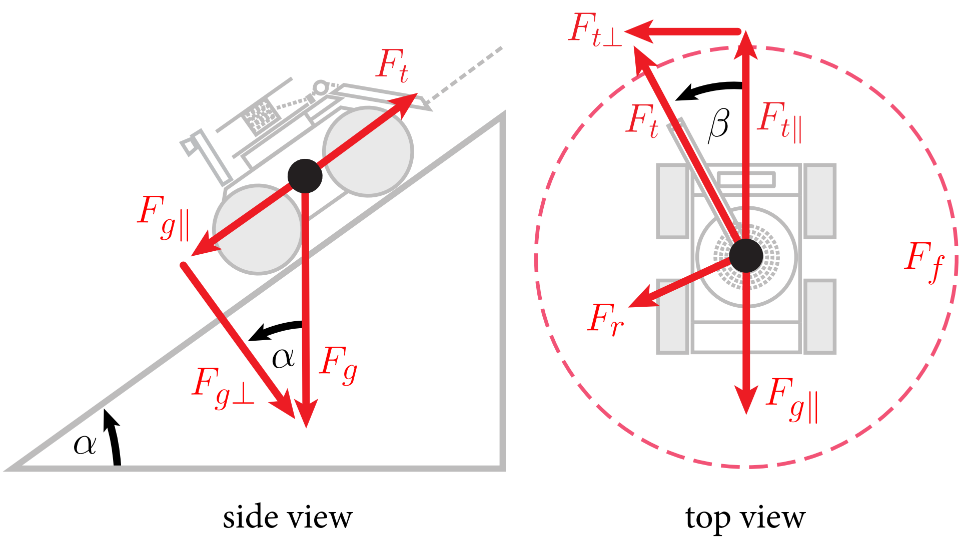


Figure 4.3: *Force Diagram*: Forces on the robot are shown as red arrows and angles as black arrows. The force due to gravity, F_g, has normal and in-plane components $F_{g\perp}$, and $F_{g\|}$, which depend on the inclination, α. A taut tether exerts an in-plane force, F_t, with components $F_{t\|}$ and $F_{t\perp}$. The angle between $F_{t\|}$ and F_t is the arm-to-gravity angle, β. The dashed, red circle illustrates the robot's traction, F_f. Looking to equation (4.5), the net in-plane force, F_r, is used to determine when the robot starts to slip as $F_r > F_f$.

which are the in-plane forces opposite and perpendicular to $F_{g\|}$. The arm-to-gravity angle,

$$\beta \in [-\pi, \pi], \tag{4.4}$$

is an angle we define between $F_{t\|}$ and F_t. When the forces F_t and F_g are out of alignment, a net in-plane force,

$$\begin{aligned} F_r &= \sqrt{\left(F_{g\|} - F_{t\|}\right)^2 + \left(F_{t\perp}\right)^2} \\ &= \sqrt{\left(F_g \sin(\alpha)\right)^2 - 2\sin(\alpha)\cos(\beta)F_g F_t + \left(F_t\right)^2}\,, \end{aligned} \tag{4.5}$$

is applied to the robot, which can cause the robot to slip.

4.3.3 Avoiding Slip

Given that our robot is skid steered (i.e., wheels on each side of the chassis are rotated in opposite directions to turn the robot in place), lateral motion is only possible if TReX slips sideways. Typically, a skid-steered robot can be kinematically represented by a unicycle model, which allows translational and rotational movement in a plane. Accordingly, excessive wheel slip must be avoided to enable the path tracker within VT&R to work in steep environments. With respect to slip, we must also consider the impact that a taut tether imparts while driving on a slope, which produces a non-uniform, external force on the robot that is not represented in the unicycle model. Our solution is to determine a set of robot configurations and slope inclinations where the tether can be controlled in a manner that prevents slip.

Slip is a function of wheel traction, which is the maximum in-plane force that can be applied before slip occurs. Traction can be modeled as Coulomb friction according to

$$F_f = \mu F_{g\perp} = \mu F_g \cos(\alpha)\,, \tag{4.6}$$

with μ being the coefficient of friction, which we experimentally[2] determine for a variety of terrain types. Recall that F_f is illustrated by a dashed circle in Figure 4.3. In practice, we use the lowest measured value for μ to conservatively determine the circle. Slip is avoided when the following holds true,

$$F_r \leq F_f\,. \tag{4.7}$$

Thus, if we control F_t to minimize F_r, slip can be avoided when

$$|\tan(\alpha)\sin(\beta)| \leq \mu\,. \tag{4.8}$$

This simple relationship is found by first taking the derivative of F_r with respect to F_t and setting to zero,

$$\frac{\partial F_r}{\partial F_t} = \frac{F_t - F_g\sin(\alpha)\cos(\beta)}{\sqrt{F_g\sin(\alpha)(F_g\sin(\alpha) - 2F_t\cos(\beta)) + F_t^2}} = 0\,. \tag{4.9}$$

Next, we solve for F_t,

$$F_t = F_g\sin(\alpha)\cos(\beta). \tag{4.10}$$

Finally, F_t is used to simplify equation (4.7) in order to arrive at equation (4.8). Our approach only requires a rough estimate of μ and measured values for α and β, and can

[2]For simplicity, we assume the terrain type remains static during a single experiment.

be used to determine the set of conditions that should not result in slip as illustrated in Figure 4.4. The figure also shows that, for an example configuration of α and β, the robot is not likely to slip on hard-packed soil and should slip on smooth concrete. With respect to VT&R, avoiding slip will increase the chance that a path is successfully repeated. During a manual teach, a warning indicator alerts the driver before a slip condition is reached, which ensures that the return path is safe to repeat.

4.3.4 Selecting a Reference Tension

Up until now, we have covered how tether tension, F_t, can be selected to minimize the net in-plane force, F_r, and reduce the chance of slip. However, if the robot is in a configuration on the slope that is very unlikely to slip (i.e., sufficiently below the curve from Figure 4.4), then there will be margin in the values of F_t, which satisfy the equality from equation (4.7).

We will now cover ways to exploit this margin to assist the robot when climbing steep terrain. We start by defining a reference tension

$$\begin{aligned} F_t' =& F_g \cos(\beta)\sin(\alpha) \\ &+ \lambda_F F_g \sqrt{\cos(\alpha)^2\mu^2 - \sin(\alpha)^2\sin(\beta)^2}\,, \end{aligned} \tag{4.11}$$

The first term of equation (4.11) is exactly the F_t that minimizes F_r from before. The second term decides how margin is exploited assuming the equality from equation (4.7) holds. A scaling term, λ_F, determines how much margin is exploited according to

$$\lambda_F = \cos(\phi)\frac{v_x}{v_{\max}}\,, \tag{4.12}$$

where the measured bearing-to-anchor angle,

$$\phi \in [-\pi, \pi]\,, \tag{4.13}$$

is the tether-arm orientation with respect to the robot body as before. λ_F also depends on the robot's linear velocity,

$$v_x \in [-v_{\max}, v_{\max}]\,, \tag{4.14}$$

where $v_{\max}$ is the maximum linear speed. These measurements are illustrated by the model in Figure 4.5. In order for the margin in F_t' to be fully exploited (i.e., $\lambda_F = \pm 1$),

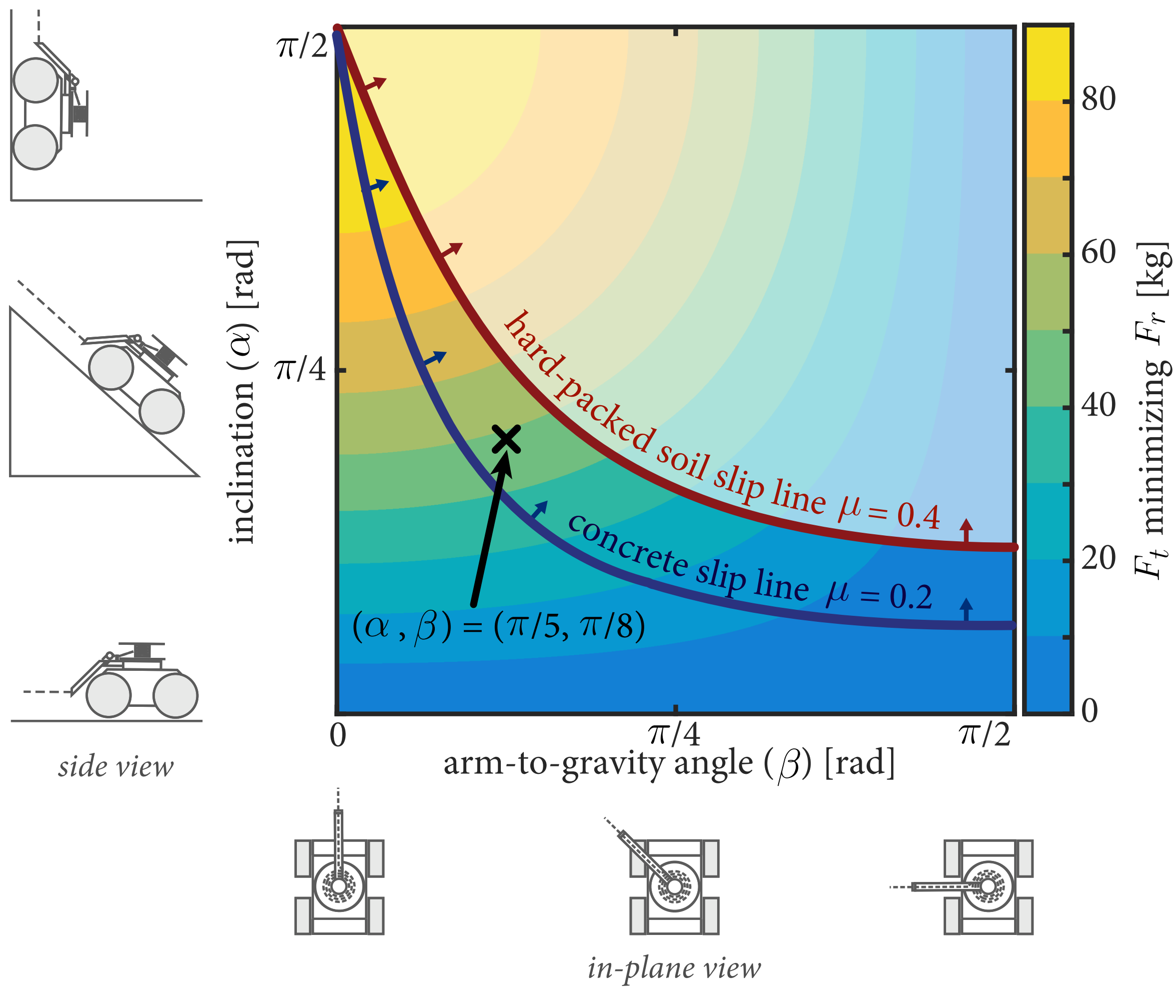


Figure 4.4: *Slip Condition*: For a range of inclinations, α, and arm-to-gravity angles, β, the tension, F_t, that minimize the net in-plane force, F_r, from equation (4.5) is illustrated as a color gradient. Cartoons of TReX on a slope are provided to visualize different configurations of α and β. If possible, slip is avoided by minimizing F_r with respect to F_t. Two curves, which correspond to different coefficients of friction, μ, show configurations that are not prone to slip on different terrain types. The areas above each curve represent potential slip configurations. For the example from Figure 4.3, an '`x`' indicates that the robot is likely to slip on smooth concrete but not on hard-packed soil.

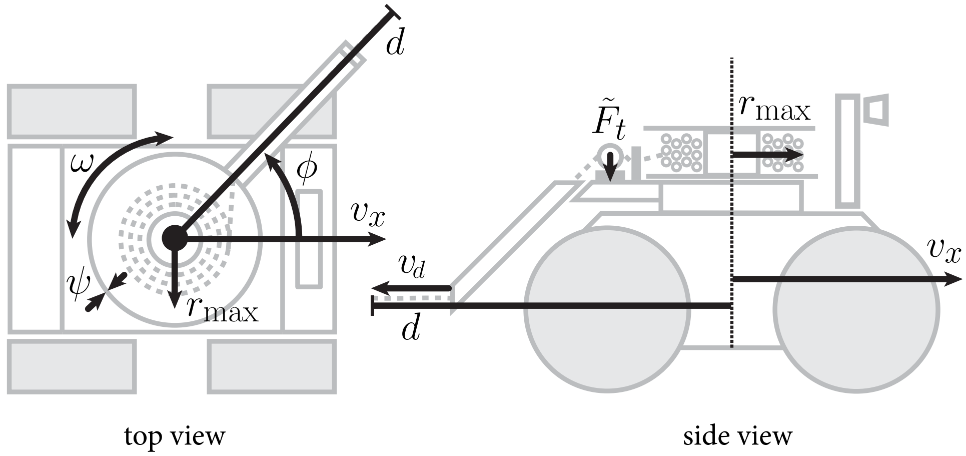


Figure 4.5: *Measurement model*: The following measurements are used to compute feedback and feedfoward control inputs. Tether length and bearing-to-anchor measurements are d and ϕ. The robot's linear velocity is v_x. the velocity of tether deployment is v_d. The spool's angular velocity is ω. The number of spool turns is ψ. The maximum spool diameter is $r_{\max}$. The measured force from tension is $\tilde{F}_t$.

the robot must drive at full speed away from or towards its current anchor. However, if the robot is already slipping, then there will not be margin available to exploit (i.e., the second term goes to zero). Therefore, the best the robot can do is choose an F_t' to minimize F_r. For the example problem from Figures 4.3 and 4.4, we illustrate how margin is selected on both hard-packed soil and concrete in Figure 4.6.

4.3.5 Feedback Control

The goal of feedback tether control is to reduce the error between a true and reference tension, F_t and F_t', which allows the robot to be supported on steep terrain and ensures the tether is safely taut regardless of the inclination. To accomplish this task, we define a tension error,

$$e_{F_t} = F_t' - \tilde{F}_t \,, \tag{4.15}$$

where $\tilde{F}_t$ is the tension measured by a piezoelectric force sensor on the robot as shown in Figure 4.5. Spool motor commands are computed by proportional feedback control:

$$u_{\text{fb}} = K_{\text{in}}\; e_{F_t} \quad \text{and} \quad u_{\text{fb}} = K_{\text{out}}\; e_{F_t} \,. \tag{4.16}$$

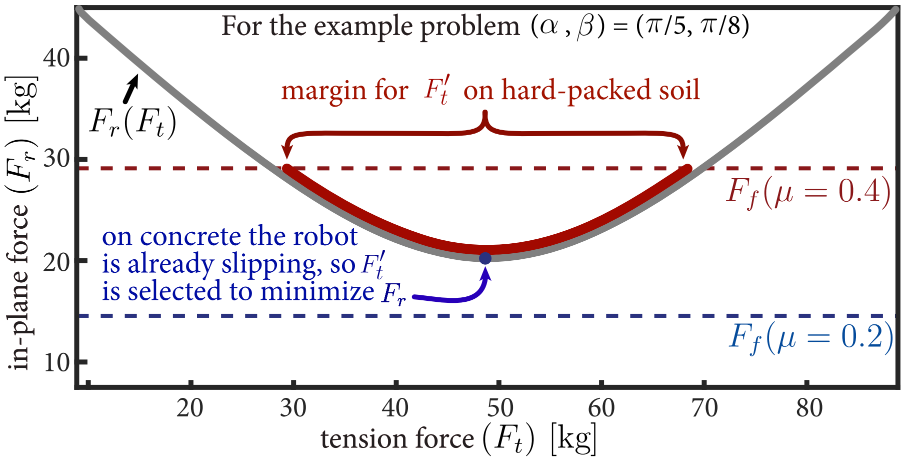


Figure 4.6: *Reference Tension*: With reference to the example problem from Figures 4.3 and 4.4, we show how margin is exploited in the potential values of F_t' that minimize F_r. F_f is a function of the coefficient of friction, μ, which is shown by dashed lines for concrete (blue) and hard-packed soil (red). The best we can do on concrete is select F_t' to minimize F_r since the robot is already in a slip condition. However, the robot driving on hard-packed soil has margin in the values of F_t' that can be selected, which is represented by a solid red line on the curve. The robot chooses F_t' according to equation (4.11) and exploits the margin for assistance on steep slopes if needed.

Gains, K_{in} and K_{out}, are used to account for sensor output discrepancies between loading (reeling in) and unloading (reeling out) conditions, which is analogous to measurement hysteresis. To handle this and also account for system dynamics, the gains decay as the slope increases and vary with changing bearing-to-anchor measurements. The parameters that determine the rate-of-decay are determined through calibration and testing.

4.3.6 Feedforward Control

We select gains conservatively for feedback control to accommodate for sensor measurement noise and prevent overshoot while tracking the reference tension, F_t'. Consequently, the controller is slow to adapt to fluctuating values of F_t'. In response, a feedforward controller was developed to account for the commanded motion of the robot, which is accomplished through a mapping of linear velocity input, v_x, to a spool motor command, u_{ff}. The following mapping allows the robot's motion and tether spool to be synchronized:

$$\cos(\phi)v_x = v_d = \omega r\,, \tag{4.17}$$

where ω is the angular speed of the tether spool, ϕ is the bearing-to-anchor measurement from before, v_d is the velocity of the tether (i.e., rate of deployment), and r is the spool radius. Since the radius of the tether decreases as tether is deployed, r is modeled as an Archimedean spiral according to

$$r(\psi) = r_{\max} - \delta r\ \psi\,, \tag{4.18}$$

which is a function of the current number of spool turns (i.e., rotations),

$$\psi \in [0, \psi_{\max}]\,. \tag{4.19}$$

After $\psi_{\max}$ turns, the tether is considered fully deployed. Both $\psi_{\max}$ and the rate of spool radius change, δr, are determined in calibration. With these terms defined, we can compute a feedforward spool command as,

$$u_{\text{ff}} = \frac{u_{\max}}{\omega_{\max}}\omega = \frac{u_{\max}}{\omega_{\max}}\frac{\cos(\phi)}{r}v_x\,, \tag{4.20}$$

where $u_{\max}$ and $\omega_{\max}$ are the maximum spool motor command and angular velocity.

4.3.7 Tethered Visual Teach & Repeat (VT&R)

With both feedback and feedforward terms defined, we arrive at the combined tether spool input,

$$u = u_{\text{fb}} + u_{\text{ff}}\,, \tag{4.21}$$

which enables tethered VT&R by (i) ensuring that the tether is safely taut at any inclination and (ii) providing motion assistance for the robot to drive as if untethered on steep slopes (assuming the slip condition is not violated). As before, the full tethered VT&R system is illustrated by Figure 4.2 above.

4.4 Experiment

Experiments are conducted in both cluttered-indoor and steep-outdoor environments to evaluate tethered VT&R and the proposed controller. For the indoor test, TReX is manually piloted around a set of bollards (i.e., posts) serving as anchors until its tether is fully deployed. For the outdoor test, the robot is piloted down a steep ($\sim 45°$), forest

Figure 4.7: *TReX Configuration*: The configuration of TReX during tethered VT&R experiments is shown. Major components are annotated. We note that the configuration has changed from experiments presented in previous chapters. In order to make VT&R work for TReX, we made the following changes: (i) a rigid mast and stereo camera are fixed to the robot body, (ii) an externally-attached, GPU-enabled computer runs VT&R on board and computes robust visual features (Bay et al., 2008), and (iii) an inertial sensor (on the mast) is used to calculate the robot's inclination with respect to gravity.

slope with trees serving as intermediate anchors. In either test, the robot is tasked to autonomously repeat the path, detach from added anchors, and return to its starting position. The experiments are intended to demonstrate that our tether-control strategy provides motion assistance and ensures a safely taut tether at a variety of inclinations and robot configurations (i.e., arm-to-gravity angles). With respect to VT&R, the taught path is repeated in both directions (outgoing and incoming) three times each to evaluate average path tracking error. Figure 4.7 shows the configuration of TReX during the experiment, which is different than configurations that have been previously described; some changes were necessary to make VT&R work for TReX.

4.4.1 Indoor Tests

Figure 4.8 provides a time-lapse image from the indoor test. The experiment involved teaching (i.e., manually piloting) a complicated, outbound path through yellow bollards that serve as intermediate anchors. As shown by the image, the tether (at the end of

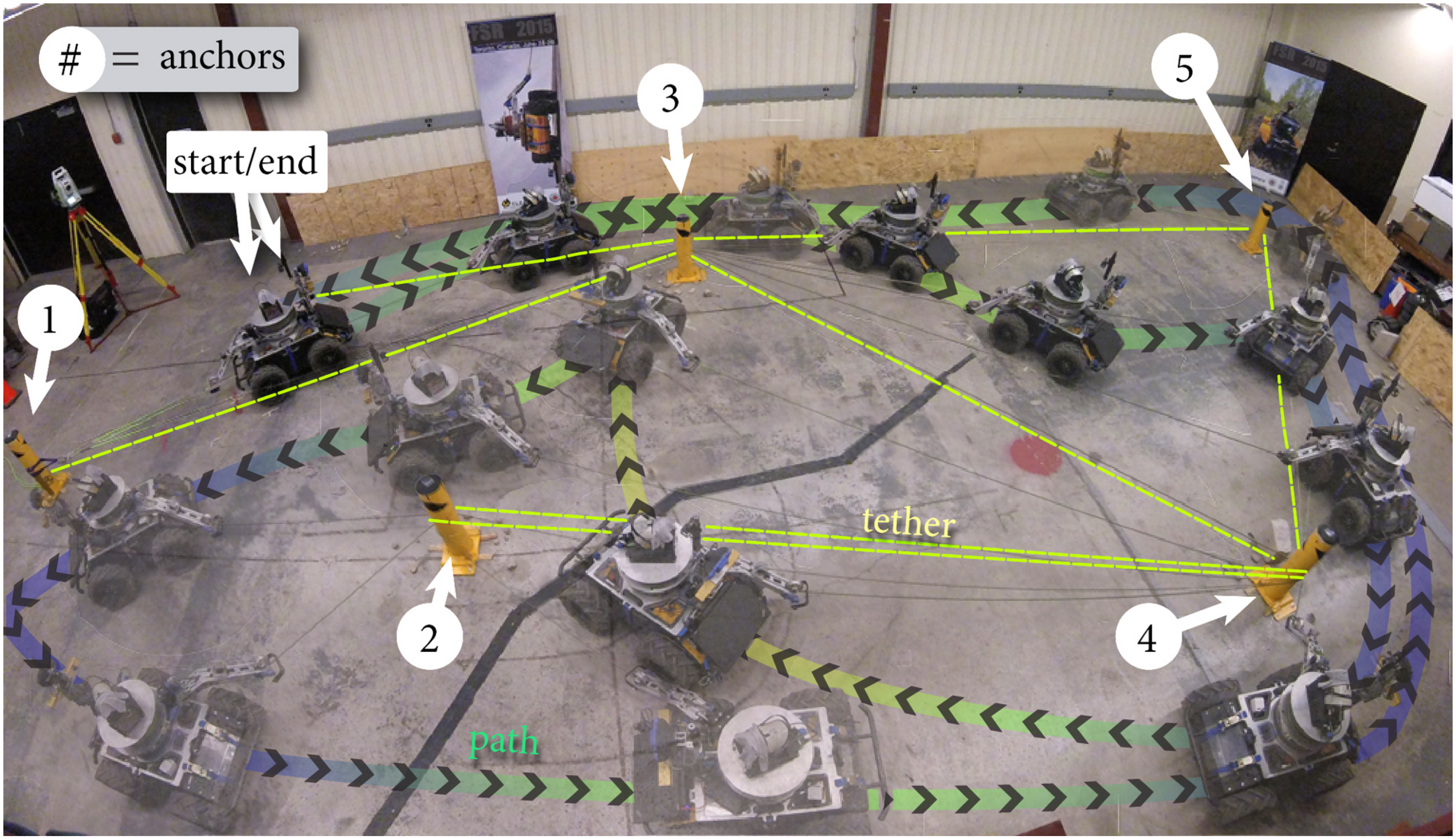
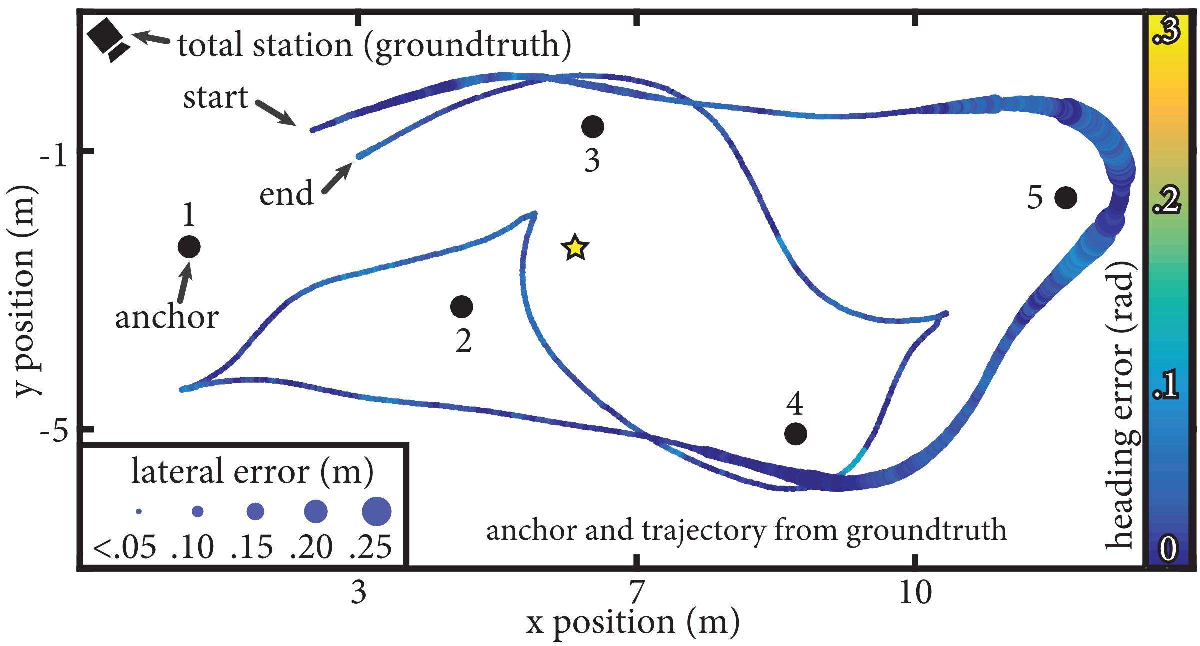


Figure 4.8: *Indoor-Experiment Setup*: TReX is shown driving in a cluttered, indoor environment in this time-lapse image. Yellow bollards or posts serve as intermediate anchors, which are denoted 1:5. The robot was manually piloted around the bollards, which are encountered in the sequence, 1 (initial), 3, 4, 2, 4, 5, 3. The trajectory of the robot's path is illustrated by a color overlay and direction arrows, which show the outbound path. To show the problem of tether entanglement, a dashed green line highlights the fully-deployed tether configuration. The robot must autonomously repeat the taught path in order to detach from added anchors and return to the start position, which is coincidently (but not exactly) close to the end position (i.e., the location at which the tether is fully deployed and the robot begins the repeat).

Incoming Trajectory



Outgoing Trajectory

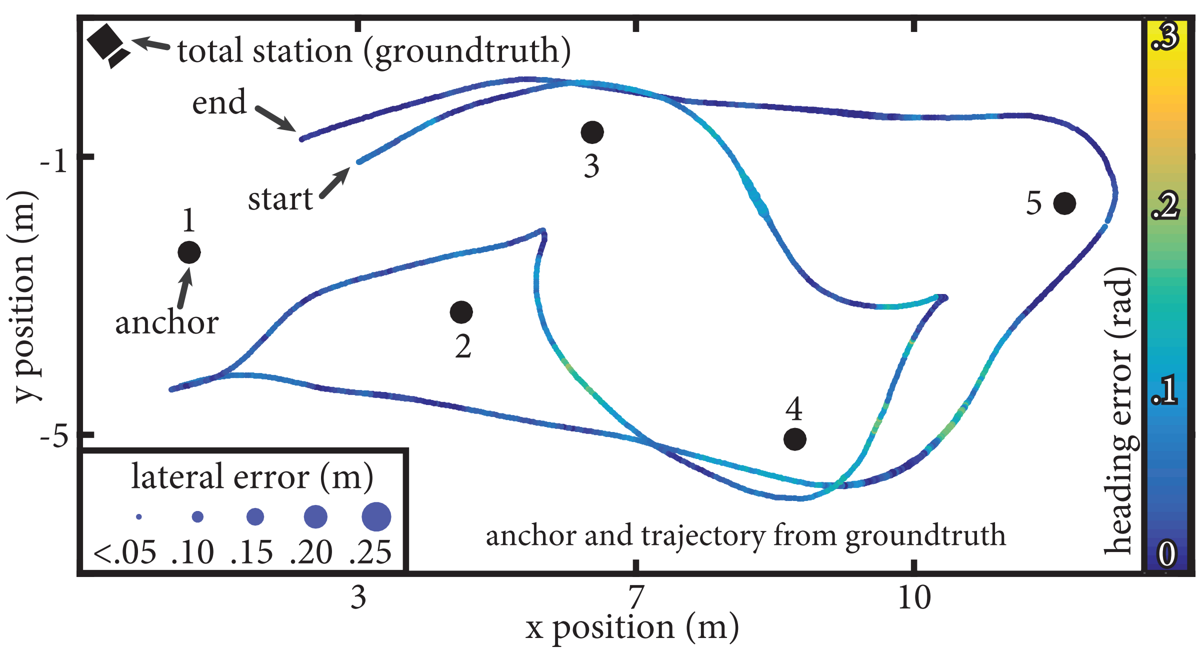


Figure 4.9: *Indoor-Trajectory Maps*: After teaching a path through obstacles and adding anchors, the robot autonomously drives along an incoming repeat path to avoid tether entanglement. We repeat this experiment three times per path direction (i.e. incoming and outgoing), while ground-truth position is collected by a survey station. The robot's average trajectory is plotted using overlapping circles, whose location is determined by the mean of ground truth for all runs. Furthermore, each circle is sized and colored according to the mean lateral and heading errors from the path tracker. Overall, these maps qualitatively show that tethered VT&R is possible in cluttered environments.

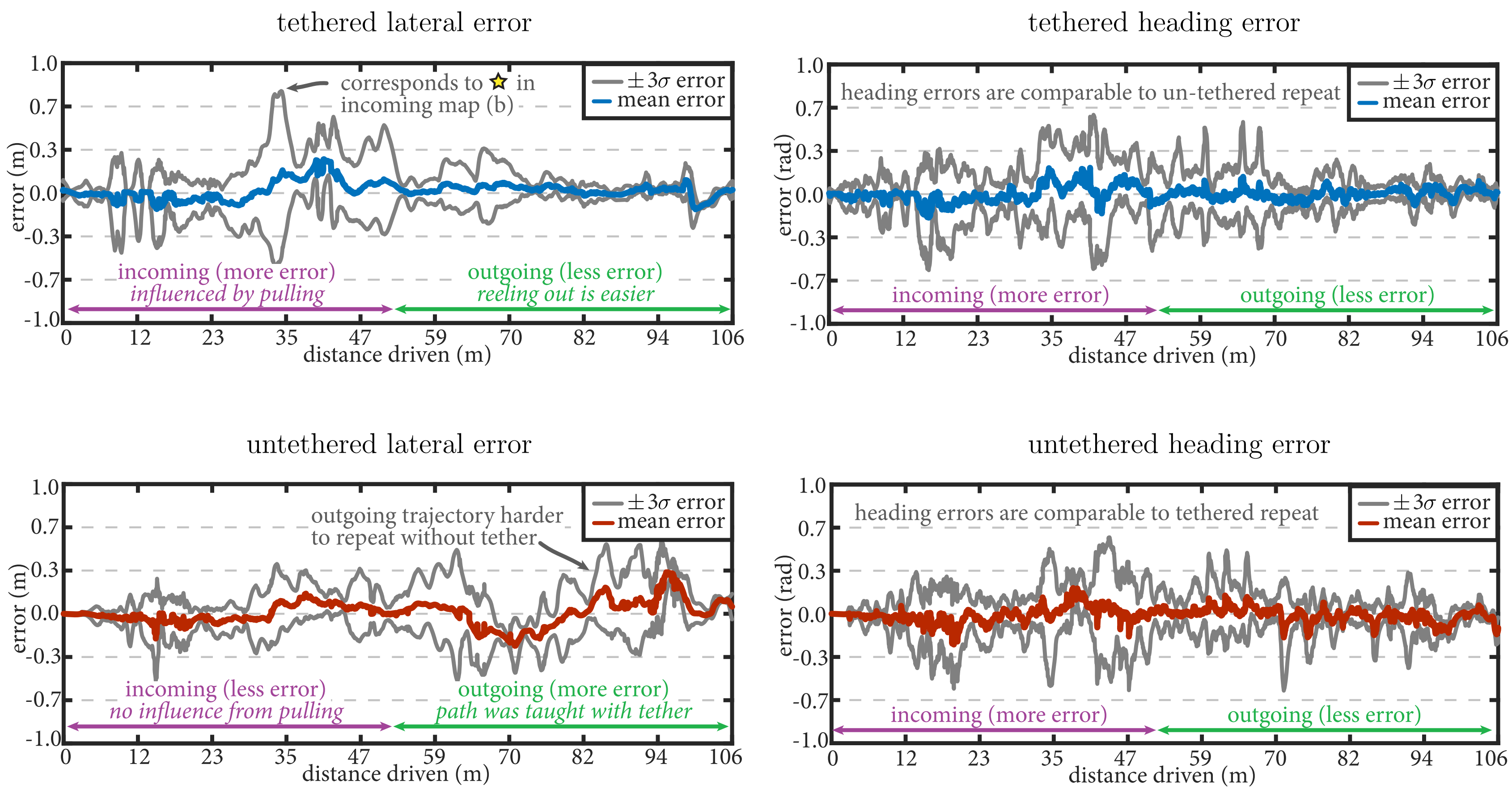


Figure 4.10: *Indoor Path-Tracking Errors*: With reference to the maps shown in Figure 4.9, the first row of plots provides the mean and $\pm 3\sigma$ lateral and heading errors over the distance driven. Both incoming and outgoing trajectories are shown together. We note that the repeat starts with the incoming trajectory. For comparison, the second row plots mean errors for a set of three untethered repeats. The takeaway is that untethered and tethered repeats have comparable tracking errors, which serves as verification that our tether controller allows the robot to drive as if untethered, while still maintaining a safe, taut tension.

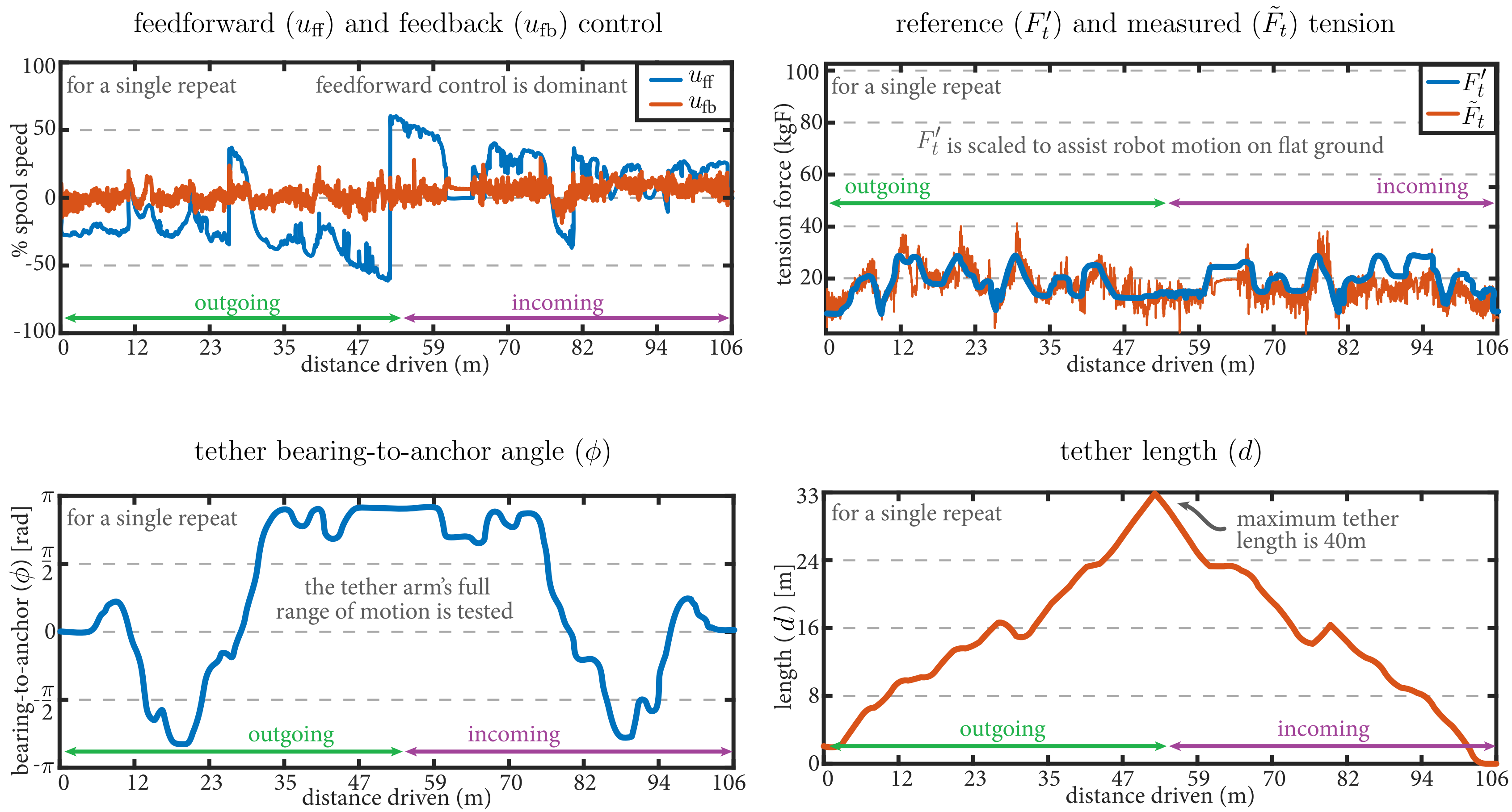


Figure 4.11: *Indoor Tether Control*: These plots concern the tether controller and tether measurements for a single repeat from the indoor experiment. The top-left plot illustrates the contribution of feedforward (u_{ff}) and feedback (u_{fb}) components to the tether spool. As shown, u_{ff} is the predominant control input, while u_{fb} ensures only that the tether is sufficiently taut. The top-right plot compares the calculated reference and measured tensions, F_t' and $\tilde{F}_t$. The bottom row shows how the tether measurements for bearing-to-anchor, ϕ, and length, d, change over time.

the teach phase) has wrapped around the obstacles and risks becoming entangled. To prevent entanglement, TReX autonomously repeats the path to sequentially detach from anchors. For the purpose of evaluation, the robot was tasked to repeat the path in both directions, three times, while ground truth was collected by a survey station[3]. For comparison, we repeat the path without an attached tether as well. Route-following performance is evaluated by looking at lateral and heading errors, which are output by the path tracker during autonomous operation.

Looking to the trajectory maps in Figure 4.9, we illustrate path tracking errors as colored circles drawn along the path. The circle positions are determined by the average position from ground truth over three repeats. The circle radius and color correspond to the mean lateral and heading errors output by the path tracker for all repeats. Since the path is repeated in both directions (outgoing and incoming), we divide the maps into two for visual comparison. The reason that the incoming return repeat path exhibits more error than the outgoing repeat is due to increased wheel slip caused by the taut tether; due to sensor saturation at low tension, the gains have been increased to preserve a taut tether while reeling in, which can cause wheel slip on smooth surfaces (e.g., concrete). For a quantitative comparison, Figure 4.10 compares path tracking errors between tethered and untethered autonomous drives, which shows comparable path-tracking performance overall. Normally, we would expect the untethered performance to be noticeably better because the external tether forces are no longer influencing motion. However, due to the fact the tether was attached during the initial teach pass, the tether actually improves the ability to track the path. We can also look at the performance of the controller during a single repeat of an outbound and incoming return path (i.e., the full trajectory) in more detail. Figure 4.11 provides plots of controller contribution (i.e., feedforward versus feedback), tether tension (i.e., measured versus desired tension), and tether-measurement change (i.e., length and bearing-to-anchor).

Overall, the key takeaway is that the proposed tether-control strategy enables VT&R for tethered robots in highly cluttered environments.

4.4.2 Outdoor Tests

A time-lapse image describing the outdoor test setup is provided in Figure 4.12. Two paths were autonomously repeated on steep terrain during the test. The straight path

[3]Leica Nova MS50 MultiStation

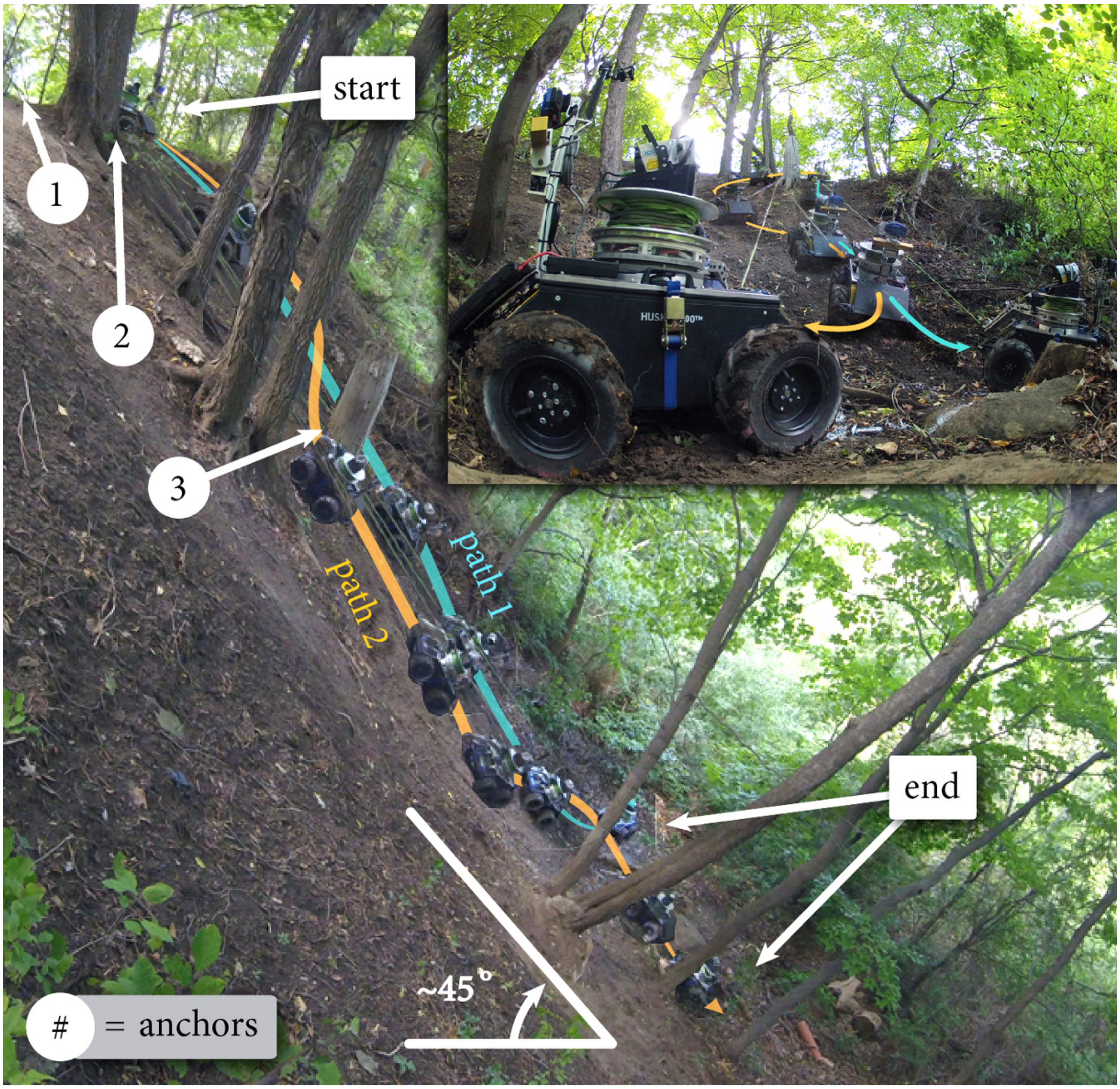


Figure 4.12: *Outdoor-Experiment Setup*: In this time-lapse image, TReX is shown while autonomously repeating a series of challenging paths on steep, outdoor terrain. As the robot descends, its tether wraps around tree trunks serving as intermediate anchors. Added anchors must be sequentially detached during the ascent to prevent entanglement. In our experiment, the first path is a straight descent without additional anchors, while the second is a challenging, slalom path through trees. Since path 1 involves no added anchors, the results from this section are given for path 2 alone.

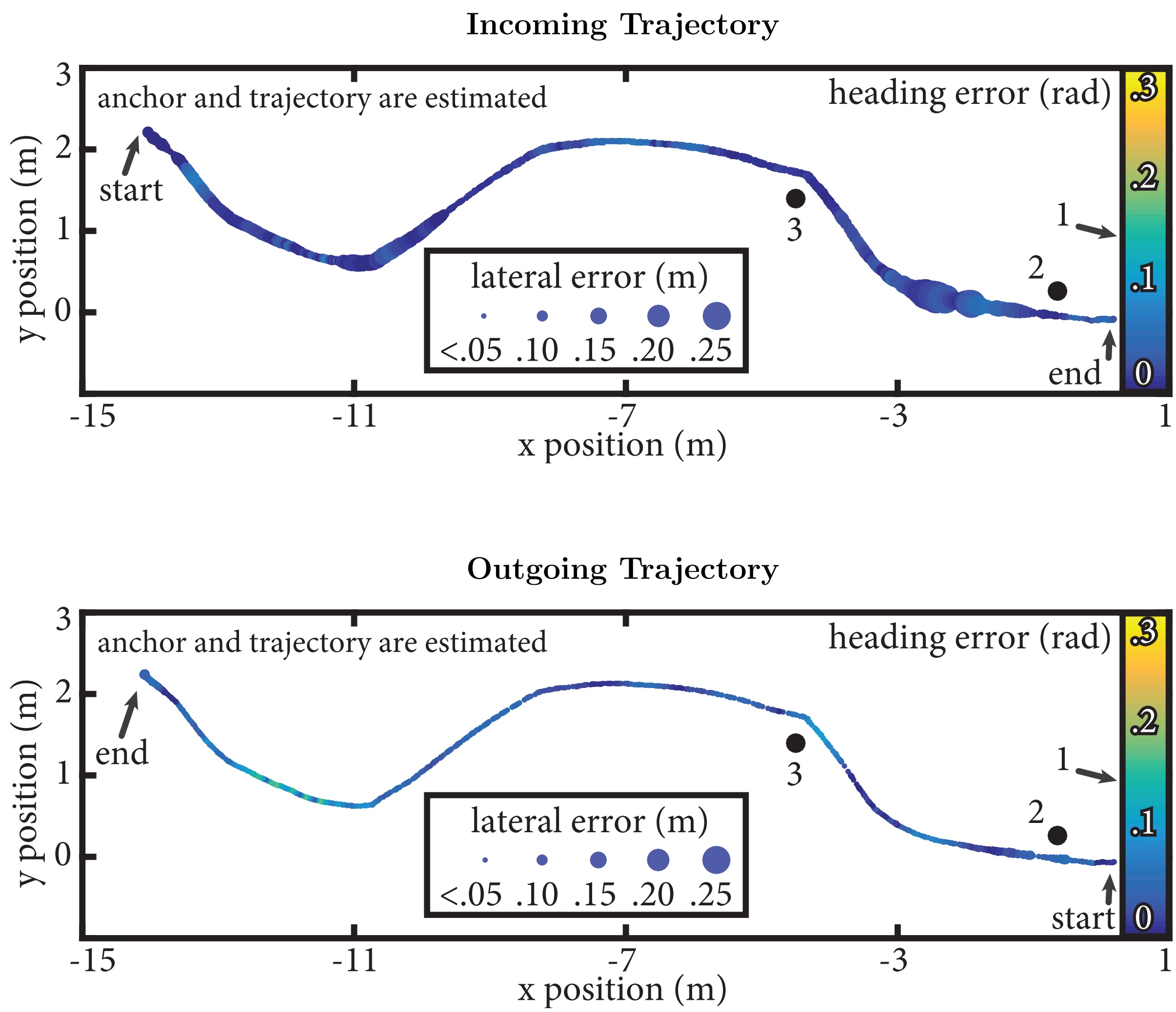


Figure 4.13: *Outdoor-Trajectory Maps*: In this experiment, a path was taught down a steep slope through trees serving as intermediate anchors. As before, the path is repeated three times per path direction (i.e. incoming and outgoing). The robot's average trajectory is plotted as it was for the indoor experiment. However, ground-truth position was not collected. Instead, we use the trajectory output from VT&R to position the circles along the path. The maps are plotted from a birds-eye-view (i.e., height information is omitted). Again, the result qualitatively shows that tethered VT&R is possible in steep, cluttered environments. Due to increased wheel traction and relative path complexity, the outdoor result shows lower tracking errors than the indoor maps from Figure 4.9.

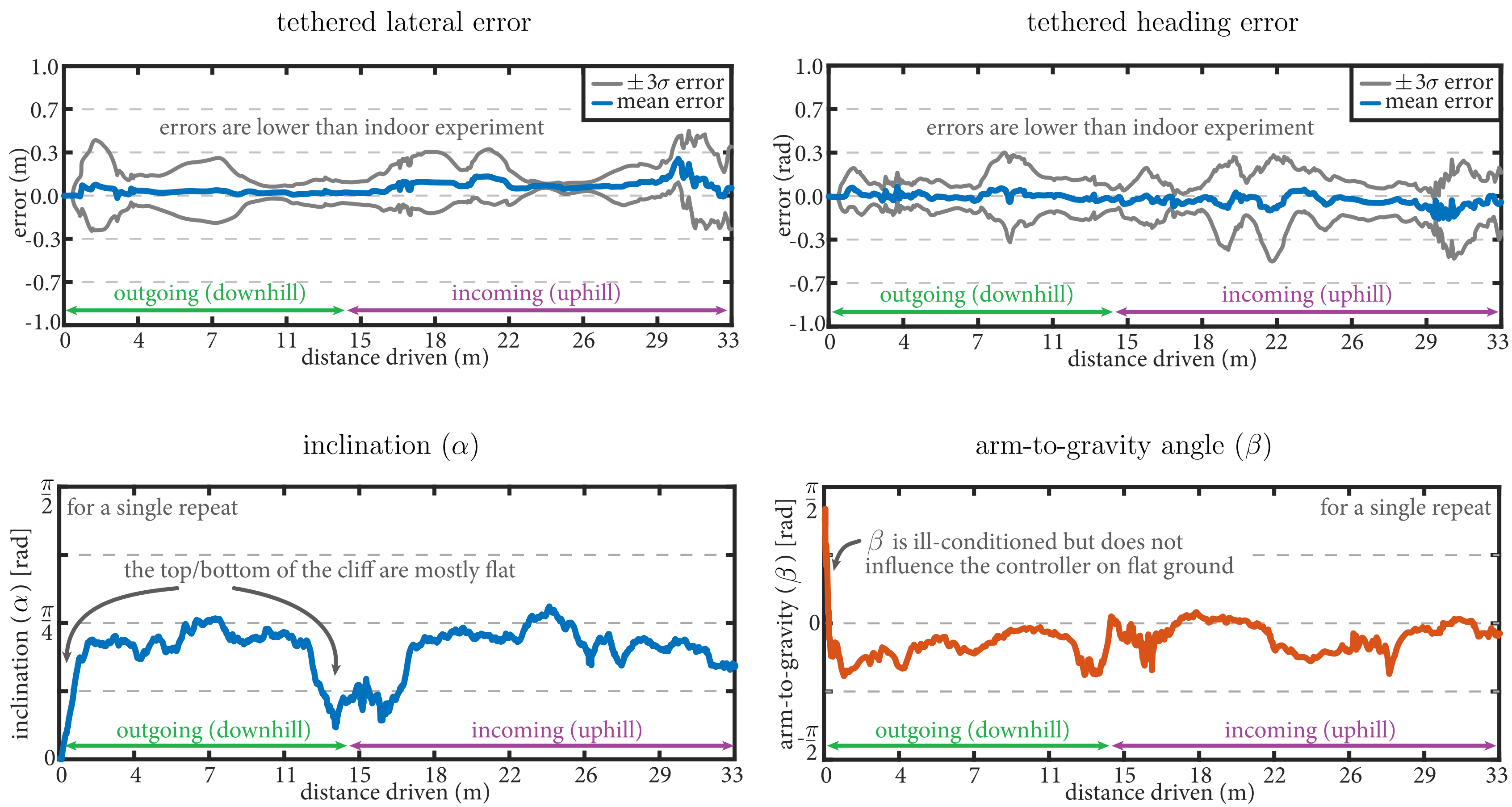


Figure 4.14: *Outdoor Path-Tracking Errors*: Extending Figure 4.13, the first row of plots provides the mean and $\pm 3\sigma$ lateral and heading errors over the distance driven for the outdoor experiment. Both incoming and outgoing trajectories are shown together. We note that the repeat starts with the outgoing trajectory. For reference, the second row plots the slope inclination (α) and arm-to-gravity angle (β) for just one of the repeats. Overall, The key idea is that VT&R is made possible on extremely steep terrain through the use of the proposed tether controller, which assists the robot to climb when wheel traction is reduced.

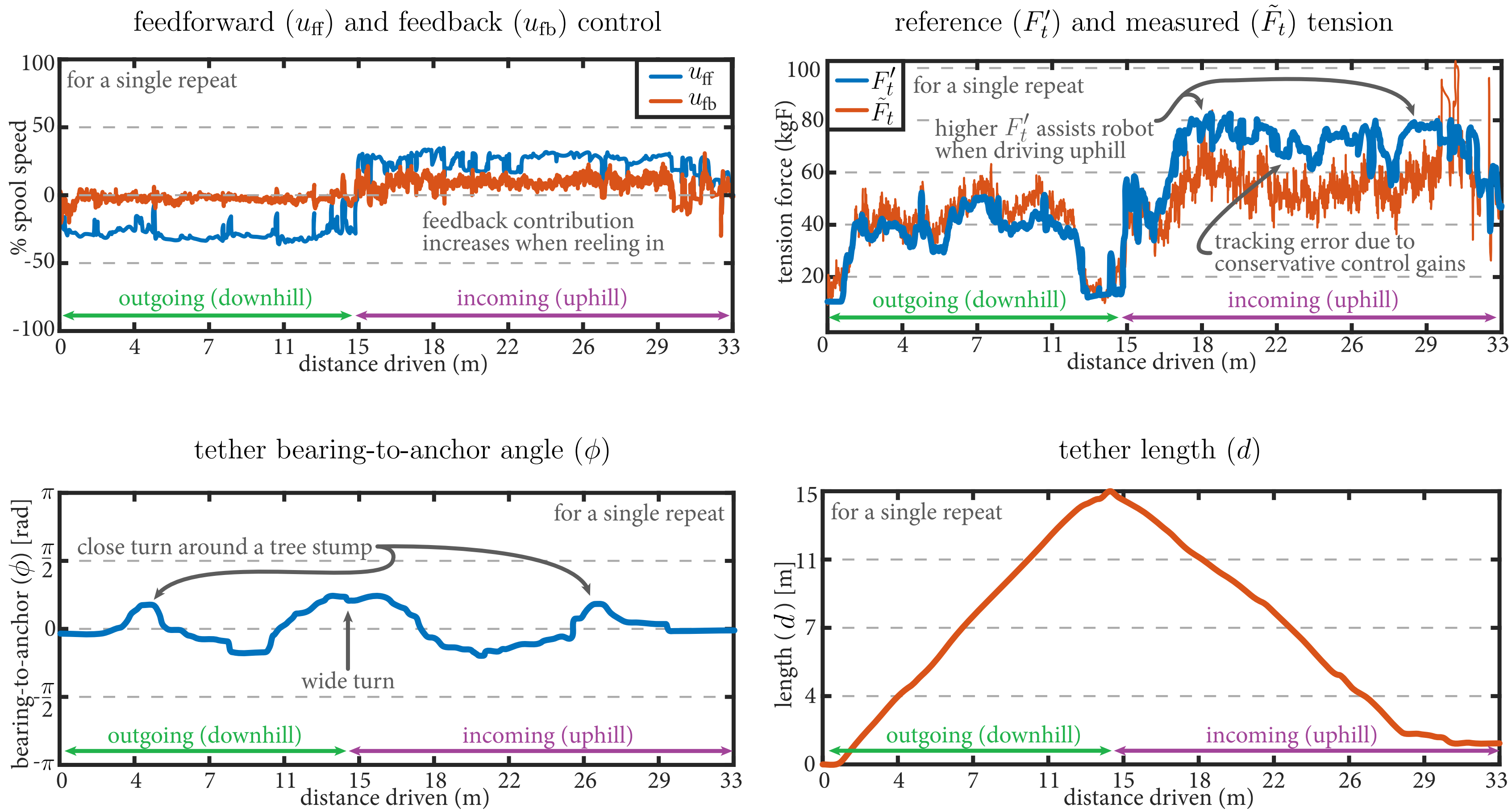


Figure 4.15: *Outdoor Tether Control*: Similar to the indoor result, these plots show the tether-control inputs and tether measurements for a single repeat from the outdoor experiment. The top-left plot illustrates the contribution of feedforward (u_{ff}) and feedback (u_{fb}) components to the tether spool. In this case, u_{fb} is stronger during the incoming repeat to assist the robot in ascending the steep slope. The top-right plot compares the calculated reference and measured tensions, F_t' and $\tilde{F}_t$. This example illustrates how margin, explained in Figure 4.6, can be leveraged for assistance. The bottom row shows how the tether measurements for bearing-to-anchor, ϕ, and length, d, change over time.

served to verify that the VT&R algorithm could operate on a steep path, where the slope inclination is roughly 45°. However, the results that follow concern a more challenging, slalom path through trees serving as intermediate anchors.

Figures 4.13, 4.14, and 4.15 provide results that follow the same format as used to present the indoor results with exception that we do not perform any untethered (i.e., unsupported) repeats. In order to compare indoor and outdoor VT&R performance, the trajectory maps use the same circle color and size metric from before. We note that outdoor repeats show less lateral/heading tracking error than indoors, which is due to increased wheel traction on the hard-packed soil and differences in overall path length and complexity. Since this test takes place on steep terrain, we also provide plots of the inclination angle, α, and arm-to-gravity angle, β, in Figure 4.13.

Again, the key takeaway from this experiment is that the tether controller allows VT&R to work for tethered robots on extremely steep, cluttered terrain.

4.5 Conclusions and Future Work

This chapter has demonstrated that visual, route following algorithms can be adapted to tethered, climbing robots navigating steep, cluttered terrain. Autonomous route following is critical to safety because, as a tethered robot navigates in the presence of obstacles, intermediate anchors (i.e., tether-to-obstacle contacts) can be added. If the robot can successfully return along a path similar to its outgoing trajectory, then tether entanglement can be avoided. Our experiments have demonstrated that, through the development of a tether controller that assists on steep terrain and allows the robot to drive as if untethered, VT&R can work for our tethered robot, TReX, without modification.

In the future, it would be beneficial to more rigorously calibrate the controller gains in order to ease the path-tracking effort. Unfortunately, this process would be difficult because our motion model does not account for non-uniform disturbances from the tether. Furthermore, the dynamics of our tethered system change with inclination, arm-to-gravity angle, tether length, and even the terrain's composition. In practice, our experimental calibration of the control gains was sufficient for tethered VT&R. It would also be beneficial to investigate methods to infer the terrain's coefficient of friction. Currently, our naive approach requires a preliminary test and assumes the terrain is homogeneous. A more robust approach could adaptively learn the coefficient using machine learning techniques through classification of the terrain by experience.

4.6 Novel Contributions

The main contribution of this chapter is to demonstrate, for the first time, autonomous route following for a tethered robot on steep, cluttered terrain, which is made possible through the development of a tether controller that manages tension on any slope and assists the robot when wheel traction is reduced.

4.7 Associated Publication

- McGarey et al. (2017b). Falling in line: Visual Route Following on Extreme Terrain for a Tethered Mobile Robot. In the *2017 IEEE International Conference on Robotics and Automation (ICRA)*.

4.8 Associated Video

- Tethered VT&R: `https://youtu.be/qqIkfSabtZs`

Chapter 5

Environment Mapping

5.1 Motivation

So far, we have covered the design, implementation, and experimental autonomy developed for the TReX platform. Now, we evaluate the robot during a field deployment to map extreme terrain in an open-pit gravel mine. We are motivated to access areas that are difficult to observe remotely (e.g., by aerial vehicle or ground-based survey) and dangerous for human exploration for applications such as geologic survey, infrastructure inspection, and reconnaissance. During the deployment shown in Figure 5.1, TReX was manually teleoperated with the assistance of an interface showing live camera and mapping views to a remote pilot. We controlled the robot to navigate steep, forested terrain, while an on-board lidar was used to scan exposed bedrock. The 2D lidar is mounted to the robot's tether spool and only rotates to produce 3D scans while TReX is in motion. Given this unique configuration, we must account for scan distortion in order to produce globally metric maps of the environment. With data collected from the deployment, we evaluate two existing approaches to estimate the robot's trajectory to account for motion distortion using (i) a continuous-time approach that handles asynchronous lidar measurements using a physically motivated motion prior, and (ii) a technique that leverages visual odometry (VO) during scan generation. Once a scan is collected and rectified, it is matched to the map using an efficient variant of the Iterative Closest Point (ICP) algorithm.

Our field results include a comparison of mapping performances and a discussion of lessons learned from the deployment. Estimated maps from each approach are compared to a ground-truth point cloud from an unobstructed (i.e., not occluded by vegetation)

Figure 5.1: *Field Deployment:* TReX navigates on steep, forested terrain during a field deployment to map exposed bedrock in an outdoor mine located in Northern Canada. In this scenario, the robot is anchored to the top of the cliff and teleoperated down a series of paths to cover a targeted area. 3D maps produced from deployments like these are useful for geologic exploration, mine inspection, and reconnaissance.

portion of the site, which is mapped using a remote survey station. The results show that VO, which is known to be accurate over short distances, best captures the robot's complex motion on extreme terrain and helps to produce the most accurate map with respect to ground truth. We highlight the key advantage of using tethered robots over other systems by providing a qualitative example, where bedrock, which is hidden from remote/aerial view by dense vegetation, is mapped by TReX in situ. Finally, we discuss lessons learned from the mapping deployment and look further into the performance of each motion-estimation technique.

This chapter is structured as follows: Section 5.2 covers prior work on lidar-based mapping. Section 5.3 details our mapping methodology. Section 5.4 describes the deployment and mapping results. Section 5.5 discusses lessons learned. Section 5.6 offers concluding remarks and future extensions. Section 5.7 states novel contributions.

5.2 Related Work

Our primary interest in developing TReX is to aid in the geologic mapping of steep cliffs, caves, and crevices on Earth and other planets. Lidar-based mapping is not only more efficient and accurate than manual survey, but it also allows for investigating the underlying rock structure and composition through lidar intensity returns (Osinski et al., 2010). With TReX, we want to produce a 3D map from a 2D lidar mounted to a rotating tether spool. This mapping problem, which involves a scanning sensor mounted to a moving vehicle, can be formulated in the context of a simultaneous localization and mapping (SLAM) problem (Smith et al., 1990). A common approach to SLAM leverages landmarks (Durrant-Whyte and Bailey, 2006; Cole and Newman, 2006), where the robot's state, which includes its trajectory and map, is estimated using exteroceptive measurements of the environment (i.e., range and bearing to landmarks).

For TReX, mapping is accomplished while moving and scanning, which requires us to consider the sensor's trajectory in 3D space during the collection of a scan. Solutions to this problem exist in the literature. Bosse and Zlot (2009) solve for the pose transformation that represents the sensor's trajectory between two scans without the need for odometric measurements using continuous-time interpolation. Zhang and Singh (2014) expand on continuous-time SLAM with the development of lidar odometery and mapping (LOAM), a real-time formulation that uses high-rate, low-resolution lidar odometry to provide a motion estimate for low-rate, fine-resolution scan registration. LOAM would not work for TReX without modification due to the fact that (i) tether spool rotation and 3D scanning are coupled to the motion of the robot (i.e., lidar odometry is not possible when the robot stops), and (ii) our intended operating environment is highly unstructured (i.e., few edges and planar surfaces). Alternatively, VLOAM, which is a camera-aided adaptation of LOAM, would work for TReX because it utilizes VO to estimate the robot's motion while scanning (Zhang and Singh, 2015).

In the experiments that follow, we test two approaches to estimate the trajectory of the robot while scanning. First, we evaluate a continuous-time approach from Anderson and Barfoot (2015) that uses a constant-velocity motion prior for the robot and allows for any-time trajectory sampling; as measurements arrive, the trajectory is queried to get an estimate of the robot's pose, which is used to rectify an individual scan before alignment. This approach differs from LOAM because it uses a physically motivated motion prior and can accommodate for periodic stops while scanning. The second approach to scan rectification uses VO as a motion prior during scan collection (Kubelka et al., 2015), which is similar to the motion estimation step in VLOAM. The continuous-time approach is ideal for its simplicity, in that it only requires lidar measurements and tuning of the motion prior. However, given the operating environment, the VO-aided approach is best for capturing the motion of the robot over complex, unstructured, outdoor terrain. After rectification is handled by one of the previous methods, scans are matched to the global map using an efficient ICP algorithm from Pomerleau et al. (2013).

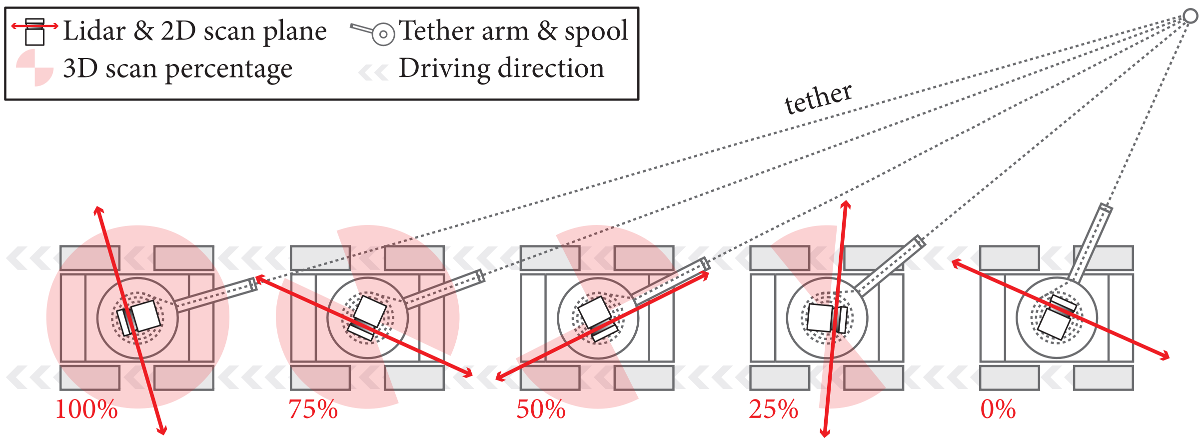


Figure 5.2: *Generating a 3D Scan*: This illustration shows how a 3D scan is generated from a 2D lidar as the robot drives and the tether spool rotates. The motion of the sensor while scanning results in distortion analogous to the rolling shutter effect on a passive camera. The typical driving distance to complete a single scan is 0.5 meters.

5.3 Mapping Methodology

The process required to generate a 3D scan from a 2D lidar attached the tether spool on TReX is illustrated in Figure 5.2. As shown, the robot must be in motion to generate a 3D scan, which results in scan distortion comparable to the rolling-shutter effect for a passive camera. Given the configuration of the lidar on the robot, each scan requires the tether spool to rotate by 180°, which translates to 0.5 m of distortion along the direction of travel at full speed. For comparison, a car with an attached Velodyne lidar scanning at 10 Hz would need to travel at 5 m/s (18 km/h) to produce the same distortion. For TReX, the problem is complicated by the fact that (i) the lidar's rotational speed is not constant given the coupling between the robot's motion and tether spool (i.e., scan distortion is not uniform over time), and (ii) our robot drives on rough, 3D terrain, which amplifies scan distortion.

In order to account for scan distortion, the robot's trajectory must be estimated during scan generation. Given the rough terrain and slow speed of the robot, wheel odometry and inertial measurements alone cannot be relied on. One solution is to assume a constant-velocity model for the robot during the collection of individual scans to accommodate for asynchronous measurements from the moving lidar. This approach uses Gaussian-Process (GP) regression to permit any-time trajectory querying using GP interpolation and, more importantly, allows the robot's pose to be queried at measurement time. Another benefit to the continuous-time approach is that it only requires one visual sensor (lidar in our

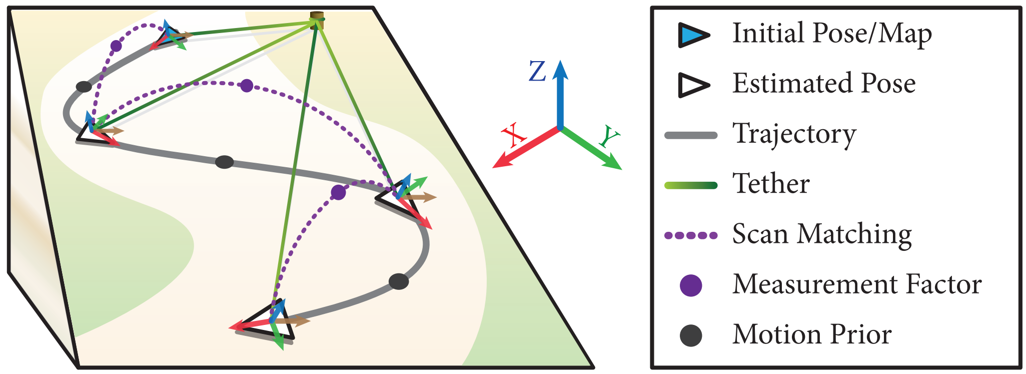


Figure 5.3: *Building a Map*: The pose graph illustrates the process of scan matching used to generate a global map. Triangles represent the robot's pose along a trajectory. A single scan is collected between each pose in the graph. In order to account for motion distortion, the trajectory of the robot while scanning is estimated using either a constant-velocity or VO-informed motion prior. Scans can be matched to a global map incrementally or in a sliding-window batch optimization; matching is handled by ICP.

case) and uses a physically motivated motion prior (i.e., white noise on acceleration). For more on this continuous-time approach, including detail on the constant-velocity model and implementation, see Anderson and Barfoot (2015). Unfortunately, this model may fail to capture the robot's complex motion over especially challenging terrain. For this reason, we evaluate a second, camera-aided approach that leverages VO as a motion prior. The benefit to this approach is that VO is known to be reasonably accurate over short distances and is better equipped to capture complex motion provided that the field-of-view and lighting conditions are generally stable. To compute VO, we use an open-source package[1], which provides pose estimates at the frame rate of the camera (10 Hz) and relies on a linear-time interpolation function available in Robot Operating System (ROS) for associating incoming lidar data to poses. After scans are rectified using one of the aforementioned methods, scan matching[2] is performed to align new scans into the global map. Figure 5.3 illustrates the scan-matching process as a pose graph.

5.4 Field Deployment & Mapping Results

Field Deployment: TReX was deployed to an outdoor, open-pit mine located in Northern Ontario, Canada[3]. The robot was driven to investigate and map an area where bedrock

[1]Fast Odometry from Vision (25), package available: https://github.com/srv/fovis.
[2]Libpointmatcher (48), package available: https://github.com/ethz-asl/libpointmatcher.
[3]Sudbury Ontario, Canada: 46°24′33.5″N, 80°50′27.3″W

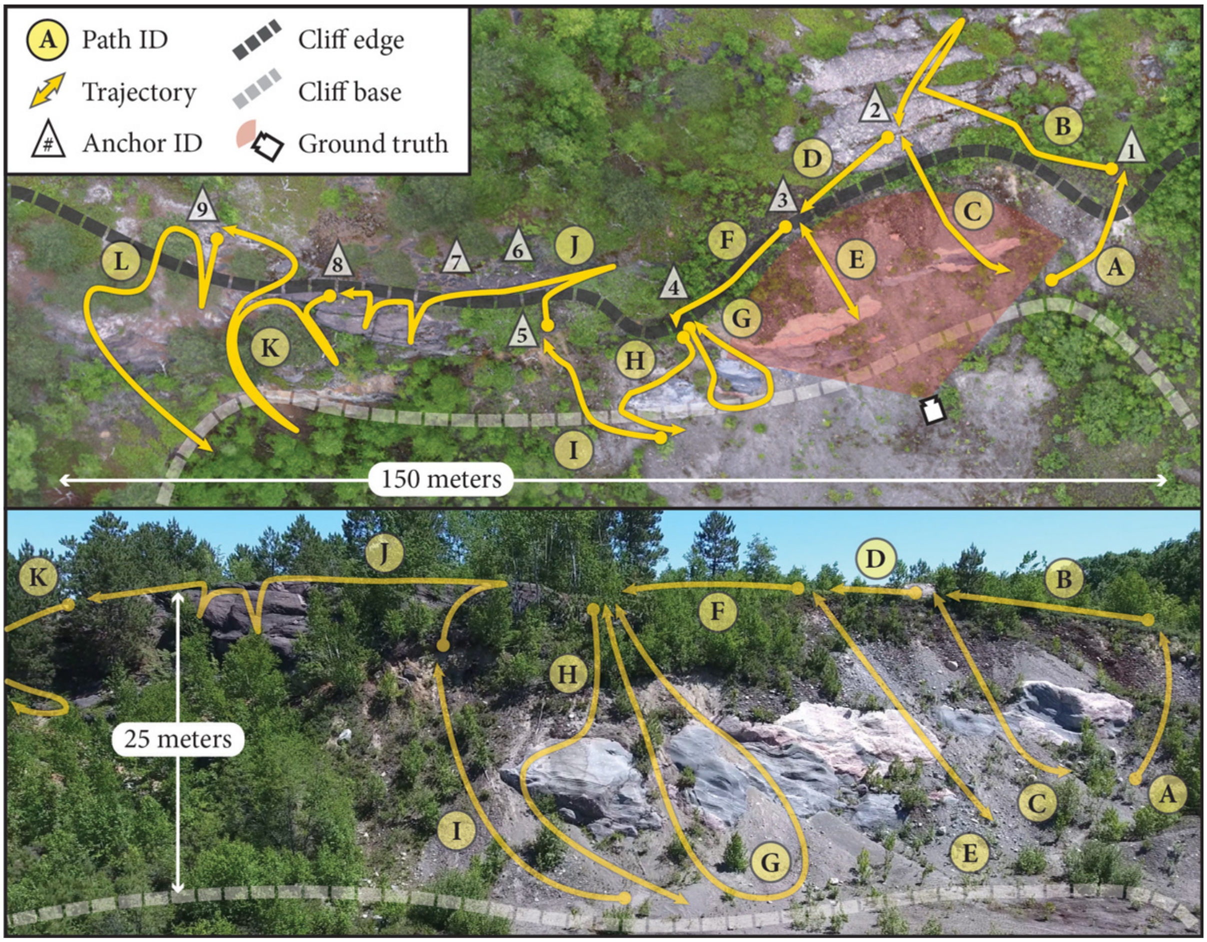


Figure 5.4: *Experiment Site*: These annotated images show the experiment site from different vantage points. The paths driven to map the exposed bedrock are illustrated. Different anchor points were leveraged throughout to gain access to new areas. A ground-truth scan was collected in the area indicated for mapping comparison. TReX is suited to navigate and map in areas that are heavily forested; the robot can navigate below the tree canopy to observe terrain that is not visible remotely.

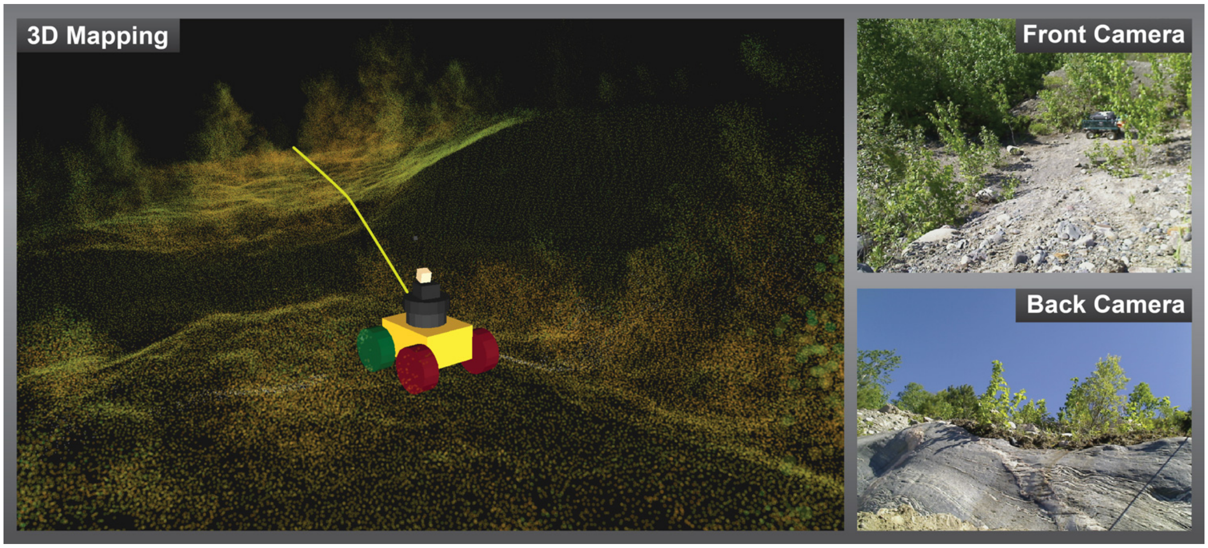


Figure 5.5: *User Interface*: This image shows the user interface used for teleoperating TReX and monitoring 3D mapping. Measurements and images from the on-board lidar and live cameras are safely transmitted by tether to a remote base station for off-board processing and storage. Driving commands to the robot can also be sent over the tether or via a wireless controller. For reference, the tether has been illustrated on the map.

is exposed on a steep, forested cliff. The field test took place over two days and involved over a kilometer of tethered driving. Images of the experiment site are shown in Figure 5.4, where robot paths (i.e., trajectories) have been superimposed to show the scope of mapping. Due to vegetation and tree cover, a large section of the rock is not visible remotely. Given the extreme terrain, the robot was teleoperated during the experiment by a remote operator. Section 5.5 provides further intuition as to why teleoperation was necessary in the field, and later, Chapter 6 summarizes the key challenges to realizing fully autonomous, tethered driving. To aid in teleoperation, the operator was able to view the user interface shown in Figure 5.5 to control the robot and monitor mapping. The interface highlights the advantage of using a hard-wired, tethered connection between the robot and base station; the tether allows data to be reliably streamed for off-board processing and safe storage.

Mapping Results: For paths C and E of the field test, the position of a marker attached to TReX was measured by a Leica Total Station[4]. The same instrument was also used to generate a dense, ground-truth point cloud on a selected area of the cliff containing both paths. Figure 5.6 shows the point cloud with trajectory estimates from continuous-time and odometry-aided ICP pipelines as compared to ground truth. We note that only the

[4]Model: Leica Nova MS50 MultiStation

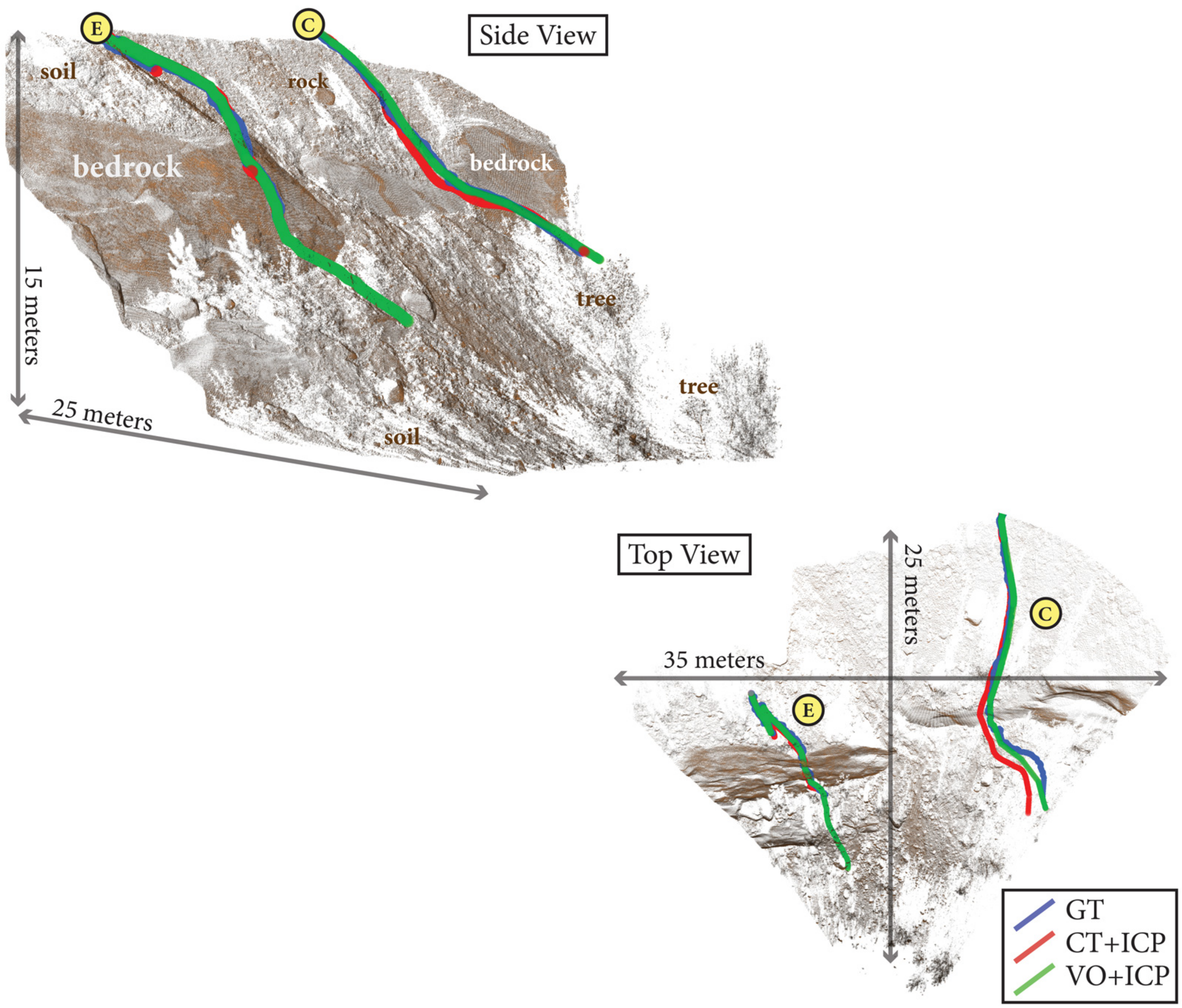


Figure 5.6: *Trajectory Comparison*: These images allow for comparing outgoing trajectory estimates for continuous-time, CT+ICP, and odometry-aided, VO+ICP, approaches to ground truth (GT). The comparison only involves paths C and E. The trajectories are overlaid on a ground-truth point cloud, which is colored by intensity. Lighter points represent more intense returns (white areas have no points). The GT trajectory is given by a position marker attached to the robot that is recorded by a ground-based survey station, which also generates the point cloud. The 'side view' shows the slope of the terrain, which ranges from 30-60°. We note that GT and CT+ICP are not complete for path E, which is due to target loss and estimation failure resulting from the robot's complex motion. Both methods are comparable to ground truth but VO+ICP is best.

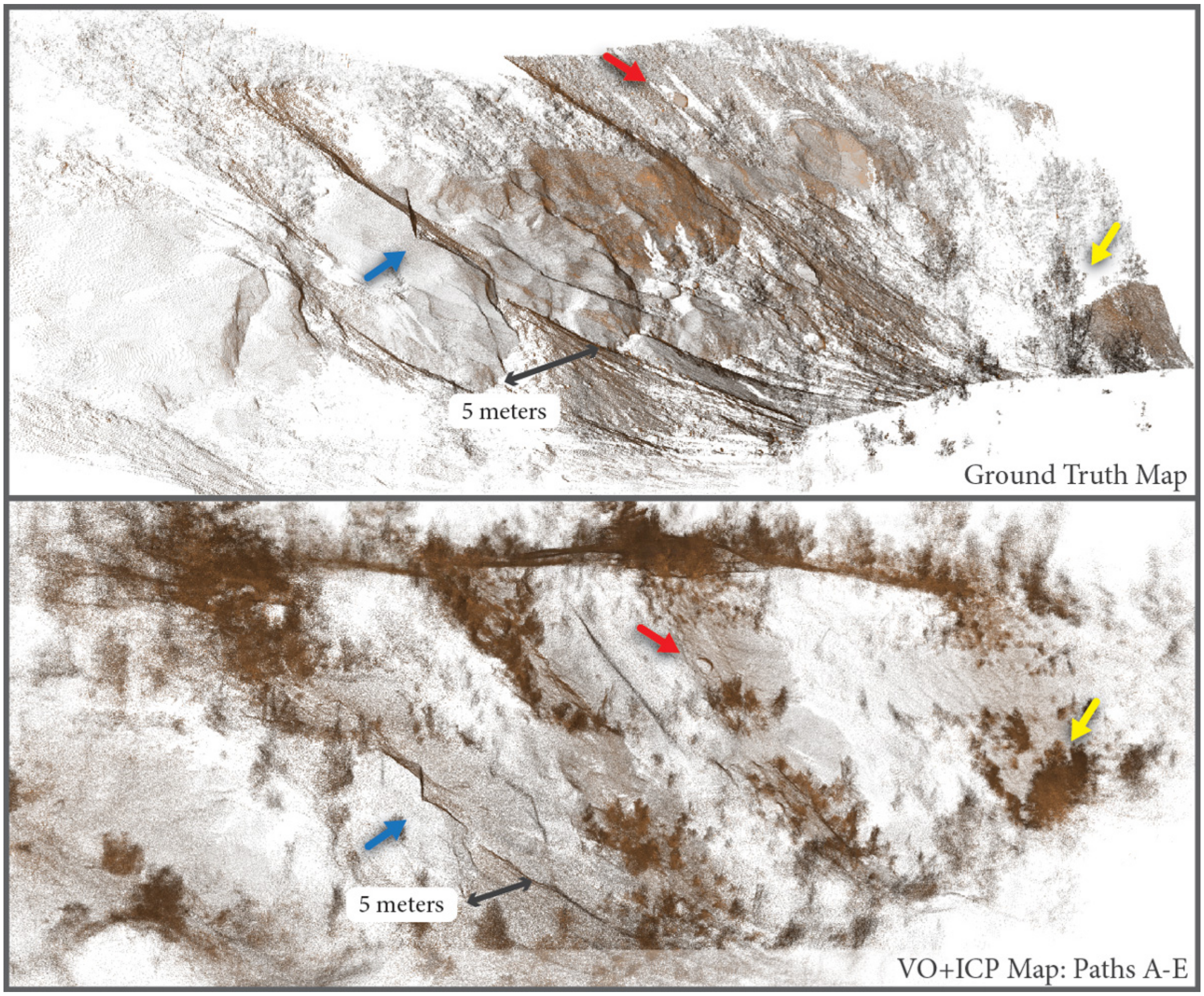


Figure 5.7: *Global-Map Comparison*: Point clouds for ground truth and the VO+ICP estimate are compared. The ground-truth map was generated for a small subset of the terrain visited by TReX. Accordingly, we show a view of the VO+ICP map, which is comprised of point clouds generated from paths A-E. These point clouds were combined manually. Common features are indicated by colored arrows. Overall, the ground-truth map is more dense, but the VO+ICP map covers more area.

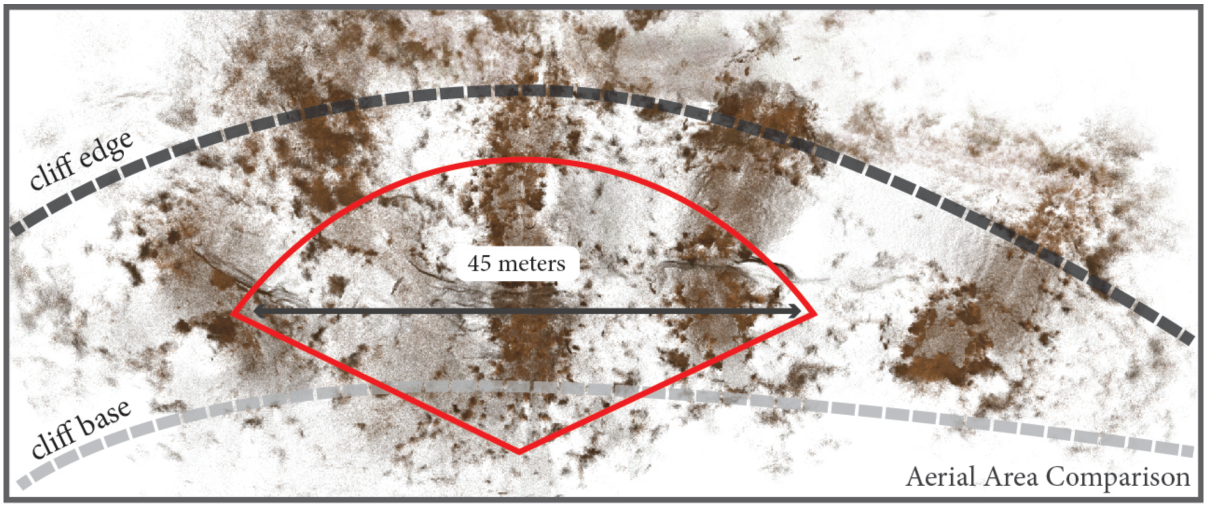


Figure 5.8: *Coverage-Area Comparison*: This top-down aerial view compares the area mapped by ground truth (highlighted in red) to area mapped by TReX. The map comes from a combination of point clouds produced on paths A-E using the VO+ICP approach (paths I-L are not pictured). The main point is that TReX can access and map areas that are not visible remotely (e.g., the top of the cliff or below the tree canopy).

outbound (downhill) trajectory is shown for clarity. While each approach generally aligns with ground truth, the VO+ICP method is able to estimate the trajectory in cases when CT+ICP fails (e.g., CT+ICP only works for the a portion of the trajectory on path E). The performance difference, as previously stated, is related to the idea that VO best captures the robot's complex motion on uneven terrain. The continuous-time approach works best when the robot's motion is relatively smooth and the environment is sparsely vegetated (see Section 5.5 for an explanation).

In Figure 5.7, we compare a combined, large-scale point cloud from the VO-aided approach to ground truth. To create the combined point cloud, maps generated from paths A through E are manually aligned. Although the resulting map is less dense than ground truth, the robot is able to cover more area than is visible by remote survey as shown in Figure 5.8. We take a closer look at path C in Figure 5.9 to compare how the continuous-time and odometry-aided approaches perform with respect to the ground-truth point cloud. Again, VO+ICP outperforms CT+ICP. However, each approach fails to capture the exposed rock face in detail because (i) individual scans collected by TReX are sparse, and (ii) an insufficient number of scans were collected of the target area.

Figure 5.10 highlights the advantage of using TReX to map in cluttered environments. As shown, the robot navigates below the tree line to visit hard-to-reach areas that are occluded from remote observation by dense vegetation.

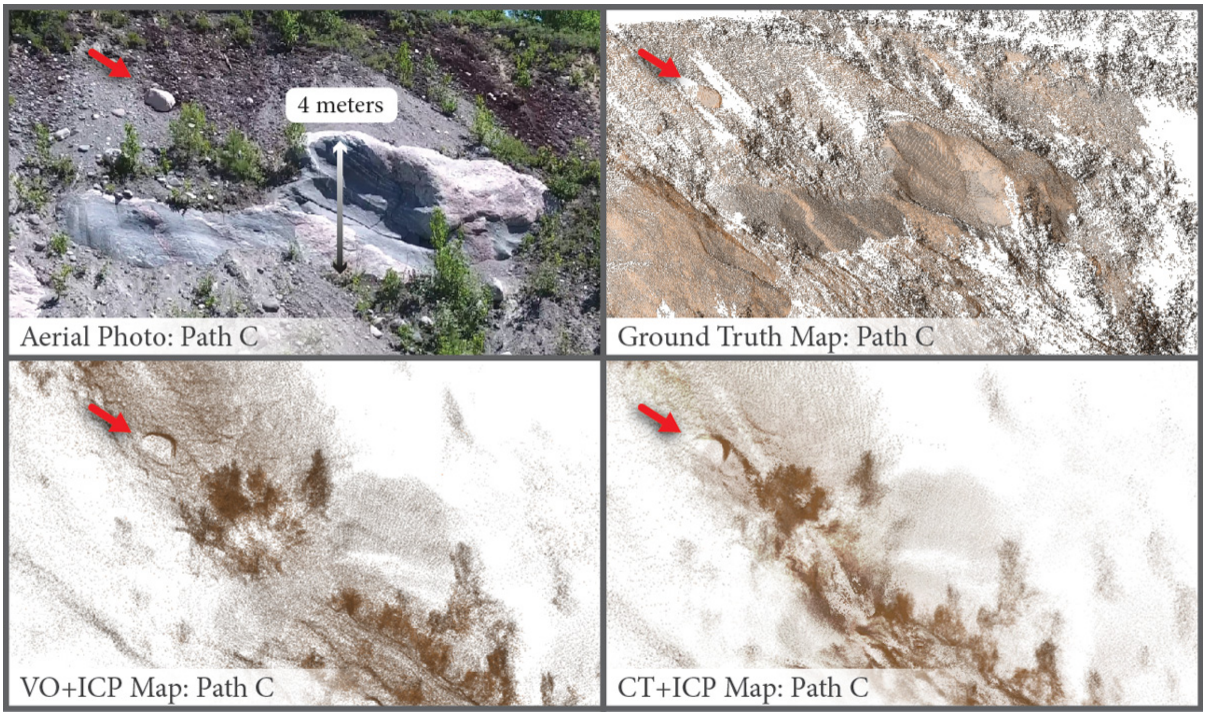


Figure 5.9: *Local-Map Comparison*: Taking a closer look, we compare virtual images from the ground-truth, VO+ICP, and CT+ICP maps colored by point intensity. If possible, it is always preferable to do targeted scanning of an area while the robot is not moving, which is why the ground-truth map is most dense and clearly shows rock features. However, the experimental configuration of our platform, which moves while mapping, still allows for reasonable accuracy at the cost of density, where the VO+ICP approach outperforms CT+ICP overall. The main cause of increased sparsity involves a lack of observation time in the target area (e.g., not enough scans were generated). We note that this is one of the few examples where CT+ICP provides a reasonable estimate; the robot's motion is too complex for CT+ICP to work for more challenging traverses. Section 5.5 looks into the performance of CT+ICP in more detail.

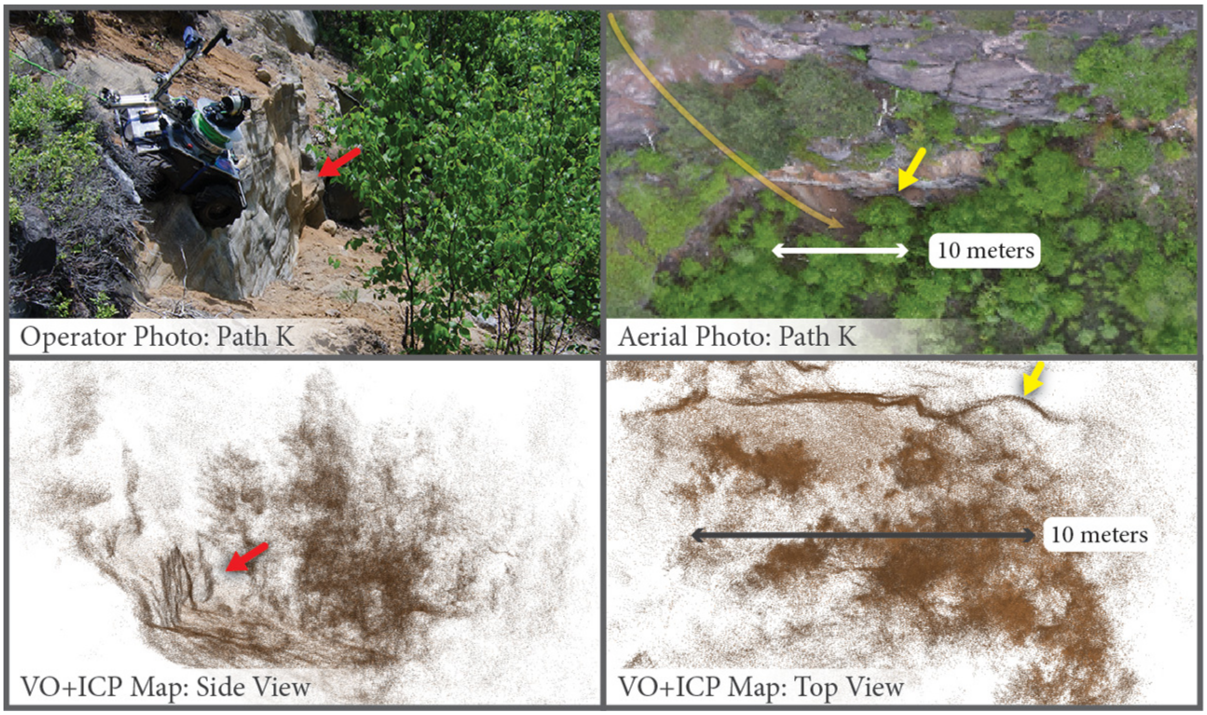


Figure 5.10: *Mapping Cluttered Environments*: The key benefit of the TReX platform is that it can navigate in cluttered environments to explore and map areas that cannot be viewed remotely. In this example, TReX climbs into a heavily forested, hard-to-reach area on path K where bedrock is exposed. Our estimated map, which comes from VO+ICP, is compared to several images collected by the operator or through aerial imaging. The point cloud provides detail on the 'hidden' vertical section of the cliff.

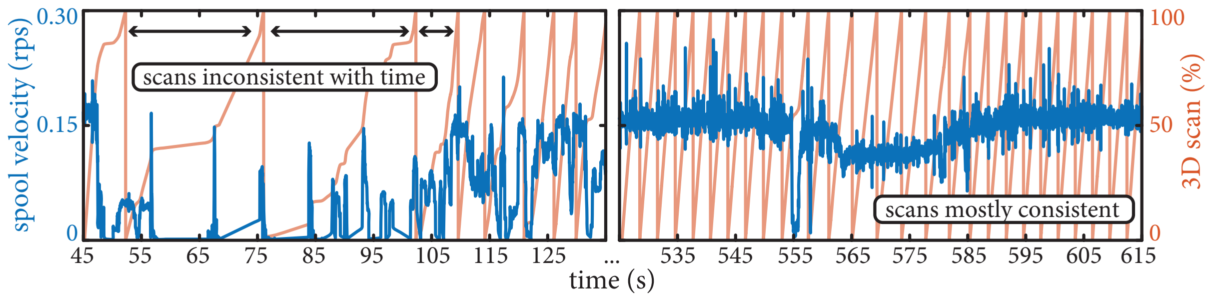


Figure 5.11: *Scan Completion Over Time*: The relationship between spool velocity and percentage scan completion is shown for two intervals of time. Given the configuration of TReX, spool velocity is measured in half rotations-per-second (rps). The important point is that scan completion varies strongly with time because of coupling with vehicle motion, which causes the constant-velocity assumption to be violated.

5.5 Lessons Learned

This section covers lessons learned during the field deployment. Specifically, we address mapping challenges, teleoperation, platform mobility, tether management, and perceived design limitations, which affect our ability to explore extreme environments.

Mapping Challenges: With reference to Section 5.4, a number of factors result in the poor performance of the continuous-time method with respect to the odometry-aided approach. Looking to Figure 5.11, one factor is the correlation of spool velocity to scan completion over time. As shown, the time to complete a scan varies strongly given the coupling of the robot's motion to spool rotation. Accordingly, the constant-velocity model assumed in continuous-time estimation does not adequately represent the robot's motion while scanning. One solution to the problem would be to add more keyframes (i.e., pose estimates along the trajectory) to represent the trajectory, which could improve the result. However, inserting additional keyframes would also slow down the pipeline and make the estimation intractable for complex trajectories. In practice, we drop keyframes upon the completion of each scan, which allows the pipeline to run in real time. As an alternative to adding more keyframes, we set the velocity prior to zero when gaps in measurements occur, which only happens when the robot is not in motion. However, these practical fixes do not address the underlying problem with the continuous-time approach, which is that the robot's true motion is not adequately captured by its motion prior. Figure 5.12 shows an extreme example of this problem at the point just as the robot drives over the edge of a cliff. The close-up view in Figure 5.12 shows how the estimated trajectory compares between constant-velocity and VO motion priors. Characteristically,

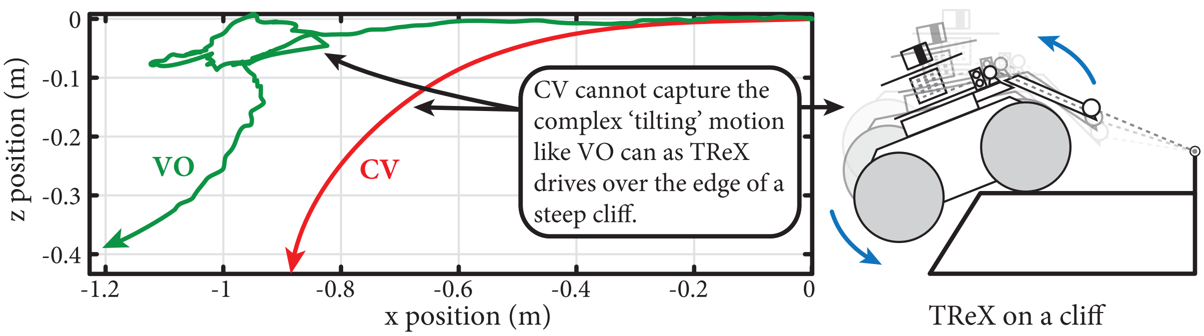


Figure 5.12: *Motion-Prior Comparison*: In order to compare motion priors, we show a zoomed-in view of the trajectory just after the robot descends down the cliff as illustrated in the cartoon. Note that a vertical profile is shown. The example best highlights the problem with the constant-velocity motion prior (CV), which represents the complex trajectory as a smooth arc. Alternatively, VO best captures the motion of the robot as it 'tilts' back and forth before descending.

the constant-velocity prior generates a smooth, arcing trajectory, while VO best captures the complex 'tilting' motion as the robot descends down the cliff. Nonetheless, VO can also fail if the environment/lighting conditions are highly dynamic, which occurs when navigating through dense vegetation (i.e., strong shadows and obstructed vision from hanging leaves).

Another concern for mapping in cluttered environments is related to scan obstruction, as illustrated in Figure 5.13. In particular, dynamic objects in the environment, like trees and vegetation, can impact scan density and quality. The result is that the average depth and number of points per scan can vary as illustrated in the plot. The scans generated by TReX are sparse in comparison to conventional 3D lidars, which produce an order of magnitude more points. For example, a Velodyne lidar can generate between 100-200 k points per scan with less distortion. Thus, it would be better to use a 3D lidar to produce maps with higher density and accuracy in place of a slow-spinning, 2D lidar attached to the tether spool. Decoupling the lidar's motion from tether spooling would allow for increased accuracy in mapping and better autonomy.

Teleoperation: The challenge of navigating TReX in the field required teleoperation. The terrain we encountered was highly varied with transitions between dense vegetation, loose sand and scree, and eroded rock material with large variations in inclination, which made the user interface difficult to use for exclusively piloting the robot. In fact, the first traverse of the robot was piloted using the interface, which resulted in the robot rolling over on a steep, convex rock (see Figure 5.14). The reason for this failure was related

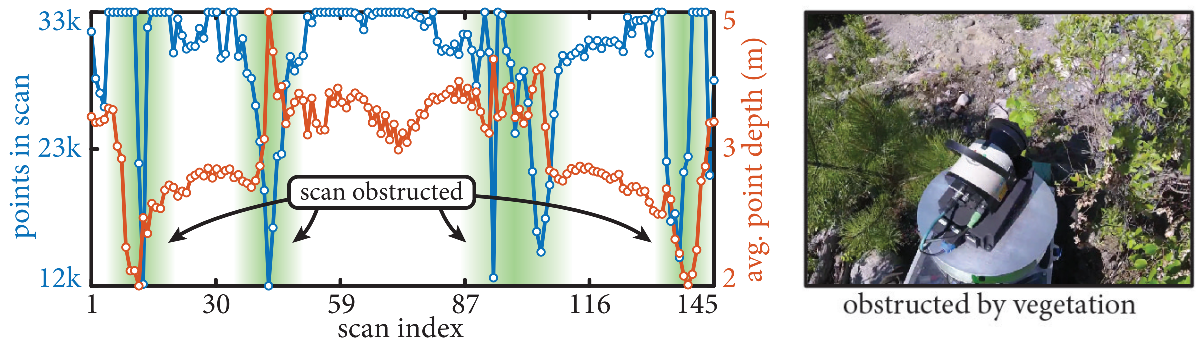


Figure 5.13: *Scan Quality*: For a sample traverse (including descent and ascent) the quantity and average depth of 3D points per scan are shown. Shaded regions highlight degenerate scanning conditions that occur when passing nearby vegetation (see the example image). Overall, scans are generally sparse and shallow in depth, which can adversely impact the scan matching process.

to a lack of knowledge about the robot's configuration on the rock, as both cameras and lidar scans were occluded by dense vegetation. Thus, the pilot would often defer to line-of-sight control of the robot during the field test. At the time of this experiment, the autonomy functions discussed in Chapters 3 and 4 were not ready to deploy into the field. In Chapter 6 we discuss about the advantages and limitations of these technologies learned through other experiments and speculate how they may apply in the field.

Platform Mobility: In Chapter 2, a basic mobility test was performed to verify that TReX could rotate on steep terrain while under tension. Now, with added field experience, we can examine the mobility of TReX in a more relevant, outdoor setting. The selection of images in Figure 5.14 were collected during the field test and demonstrate both the advantages and current limitations of the TReX platform. For the most part, TReX was able to successfully navigate a range of inclinations over mixed terrain (e.g., rock, sand, or grass) as shown in images (1-3). Images (5-6) provide further examples of the robot's ability to turn in place and drive laterally on steep terrain. However, there were some limitations of the platform that should be noted. Image (4) shows an example where TReX drives on an over-vertical surface, which causes the front tires to lift and results in decreased platform mobility. Image (7) shows TReX after it has tipped over while driving on an angled surface, which slopes tangent to its driving direction and tether arm configuration. This problem indicates that (i) the center-of-mass is too high, and (ii) the robot would benefit from a passive differential suspension (e.g., rocker-bogie system) to accommodate for uneven terrain. Tipping is a serious problem in the field that requires manual intervention. As such, future iterations of the design should consider a configuration that is self righting.

Tether Damage: Caused by acute bend angles, excessive twisting, and abrasion from rough surfaces, tether damage is a constant concern when navigating outdoors. Damage can result in power and data connection loss, or worse, a dangerous fall. In practice, it is extremely challenging to navigate outdoors, let alone autonomously, while also considering how each tether-to-obstacle contact may impact tether health. Image (8) from Figure 5.14 shows an example where the robot's tether is wrapped around a sharp rock face in the field. Since tether damage cannot be completely avoided, we replace the tether after each deployment. Unfortunately, this task results in additional cost and time but is necessary for the safety of the robot and operator. A robust approach to the prevention of tether damage would involve (i) ensuring that the bend radius throughout the tether management system is within tolerance, (ii) implementing a coiling or level-wind system to avoid internal twisting, and although difficult in practice, (iii) detecting obstacles that may damage the tether and avoiding them as potential anchors.

Size Limitations: The robot's large form factor makes it difficult to deploy in remote locations. Currently, TReX has a mass of 100 kg, encompasses 1 cubic meter, and requires at least two people for deployment. If the robot starts at the bottom of a cliff, as it did in our experiment, then the operator must hike up with a rope, generator, and base station. Once on top, the rope can be thrown down to a second person, who attaches it to the tether. The tether is then pulled to the top and attached to a sturdy anchor. After the initial ascent, the robot is re-anchored at different locations to access the desired terrain. In order to reduce the complexity and number of people required to deploy TReX, future designs should consider a platform that is scaled down to be deployed by a single operator with ease. In other words, the system (i.e., the robot, generator, and base station) should be made backpackable. As an additional benefit, the size reduction would allow a thinner tether to be used, which may increase the overall range of the vehicle.

5.6 Conclusions and Future Work

This chapter has explored the field deployment of TReX to an outdoor mine located in Northern Ontario, Canada. The purpose of the deployment was to evaluate the robot's ability to map exposed bedrock on extremely steep terrain and further test the system. We discuss and compare several approaches to deal with motion-distorted scans while mapping and produce a set of point clouds, which are compared to ground truth. We show

Figure 5.14: *Platform Evaluation*: These images provide a visual evaluation of TReX's advanced mobility during the field deployment. Images (1-3) demonstrate that TReX can navigate on a variety of steep surfaces. Image (4) provides an example where the robot's tires lift off the surface due to the over-vertical slope and direction of tether tension, which results in reduced mobility for the robot. Images (5-6) highlight the robot's ability to turn under tension and drive laterally on steep terrain. Image (7) shows a failure where TReX has tipped onto its side and requires human intervention. The fall occurred when the robot's heading and tether arm were aligned tangent to a highly angled surface. Image (8) shows a taut tether in contact with a sharp rock, which can result in abrasion, over bending, or worse, breakage.

that VO provides the best motion prior when navigating through complex, unstructured environments. Additionally, a combined map of the cliff is produced, which covers areas that are occluded from remote view by dense vegetation. Finally, we provide lessons learned from the field and discuss challenges to mapping. Overall, TReX was successful in investigating extreme terrain and shows promise for use in future mapping missions.

Future missions with TReX should consider deployment to additional harsh environments like caves, mine shafts, and dams, where we would implement autonomy in mapping by leveraging route-following capabilities presented in Chapter 4. In terms of mapping output, we could incorporate additional sensors to use in motion estimation, including inertial, inclinometer, and tether measurements. Although we have a performed a qualitative study in platform mobility, which concerns how TReX moves in translation on rough terrain, future work should include a look into platform stability (e.g., how the robot pitches and rolls and what impact does the tether have on this motion). Finally, with lessons learned from the development of the TReX prototype, future designs should consider a form factor reduction that would make the robot more transportable and economical for use in research, inspection, and reconnaissance.

5.7 Novel Contributions

The main contribution of this chapter is to field test TReX in a relevant environment and generate a 3D map of the terrain from highly distorted lidar measurements.

5.8 Associated Publication

- McGarey et al. (2017c). Field Deployment of the Tethered Robotic eXplorer to Map Extremely Steep Terrain. In the *2017 International Conference on Field and Service Robotics (FSR).*

5.9 Associated Video

- TReX in the Field: `https://youtu.be/VakpChosVNE`
- Mapping Extreme Terrain: `https://youtu.be/9r10kC7GTmc`

Chapter 6

Conclusion

6.1 Thesis Summary

We have evaluated how tethered mobile robots can be used to effectively explore and map extreme environments for geologic survey and structural inspection. When a mobile robot is outfitted with a supportive electromechanical tether, it benefits from continuous power and hard-wired communication to a remote base station. However, as the robot navigates in cluttered environments, its tether is susceptible to entanglement. Tethered Robotic Explorer (TReX) was designed to map steep, cluttered environments and enhance tethered mobility. We explore solutions to the tether entanglement problem, which are to (i) map the location of tether-to-obstacle contacts, and (ii) retrace the robot's outgoing path in order to sequentially detach its tether from obstacles. Our work shows that (i) can be accomplished by tethered simultaneous localization and mapping (TSLAM), where tether measurements and wheel odometry are used to nonvisually estimate the robot's pose and location of tether-to-obstacle contacts. We show that (ii) can be approached by performing visual route following with TReX on steep, cluttered terrain using the Visual Teach & Repeat (VT&R) algorithm. In order for VT&R to work with TReX, a tether control strategy was developed that ensures tether tautness, allows the robot to drive as if untethered, and aids the robot to climb when wheel traction is reduced. Finally, a large-scale, field deployment was conducted at a mine site to demonstrate TReX's ability to navigate and map extremely steep terrain in a relevant environment.

6.2 Novel Contributions

The following thesis contributors have been achieved in this thesis.

- The first tethered robot design capable of continuous rotation under tension.
- The first formulation of the TSLAM problem with incremental and batch solutions.
- The first demonstration of autonomous route following on steep, cluttered terrain.
- The deployment of TReX to map steep terrain in an outdoor environment.

6.3 Lessons Learned

This section summarizes engineering lessons learned from the development and field deployment of TReX. These notes are provided for those interested in developing future tethered robotic systems and are organized by category.

Systems Design:

- *Mobility*: Several considerations should be made related to mobility; (i) the mass distribution of the robot should be close to the ground to prevent tipping, and (ii) turning-in-place with a skid-steered tethered robot is a high-torque action that requires sufficient motor power. While TReX is a successful example of a conventional, ground robot outfitted to access extreme terrain, the base platform is underpowered (considering the terrain) and not ideal for traversing steep, rocky environments. Instead, the design should consider a differential suspension (e.g., rocker-bogie system) for navigation on uneven terrain, more powerful drive motors, and a reduction in size for increased stability.
- *Size*: The current size and mass of TReX ($1\,\mathrm{m}^3$ and $100\,\mathrm{kg}$) are limiting factors to its adaptation and usefulness in the field. Currently, at least two operators are required for deployment. When considering scale, both the robot and base station (computer, anchor equipment, and generator) should be sized such that they can be stored in a large backpack and carried by a single operator. The size reduction would also allow for a thinner tether to be used, which, depending on the spool capacity, would increase the operational range of the robot.

- *Tension Measurement*: The TReX design measures tension using a force-plate assembly on the tether arm, where small fluctuations in the sensor's structure are converted to a calibrated voltage output. The drawback of this approach is that the rigid configuration allows for minimal shock absorption or flex, which makes the output of the sensor noisy (especially at low tensions). Currently, shock is absorbed directly into the tether arm, which is not ideal for structural safety, as the strength of metal components will degrade with continued use. An alternative approach would be to account for shock absorption using a spring-mass damper system, which could ensure consistent sensor readings and better tether control.

- *Tether Damage*: Tether can be damaged in two ways, (i) by the environment, or (ii) by the tether management system. Handling (i) is less straightforward and would require anchor planning, which is beyond the scope of this work. Handling (ii) is accomplished by limiting tight bends, twisting, and abrasion within the system. Our tether management system uses a set of pulleys to passively guide the tether on/off the spool as the robot moves. In some instances, the recommended bend radius for the tether is exceeded due to the size of the pulley, which results in degradation of the internal wires over time. This degradation is made worse by tether twist, which is caused by continued spooling. In practice, we simply track the damage and replace the tether after a number of deployments. We also do not incorporate a level-wind system to prevent non-uniform tether buildup on the spool, which in some cases results in tether 'knifing' (i.e., the outer layer of tether slips between the inner layer and becomes caught).

Tethered Autonomy:

- *Entanglement Prevention*: Visual route following is the most straightforward approach to preventing tether entanglement. However, the approach is limited by our understanding of tethered system dynamics and intrinsic properties of the terrain. In practice, we assume both a simple model for the vehicle and a static coefficient of friction for the terrain. A more robust solution would be to implement an iterative-learning approach to reason about safe regions for exploration and better dynamic system modeling.

- *Anchor Detection*: While TSLAM is an interesting problem in 2D, it is not practical for extreme environments without adaptation. Adapting the problem to 3D environments would require the use of additional sensors to constrain the pose of the robot. However, we would still have to deal with the fact anchors are dynamic and can change position or be removed entirely. The solution is to detect/sense anchor change; it may be possible to do frequency analysis on a tether with an integrated fiber-optic cable to determine where bends occur or have been removed, but that is beyond the focus of this work.

Environment Mapping:

- *Lidar Configuration*: The lidar should rotate independently of vehicle movement to enable targeted, dense mapping. Due to its increased availability and reduced form factor, 3D lidars would be beneficial for mapping because they limit scan distortion. Another concern is that falling rocks or debris can strike and damage the sensor if not shielded. Accordingly, the lidar should be mounted so that rocks falling from above cannot hit the sensor's lens or sensitive electronics.

- *Intensity Mapping*: Given our interest in leveraging intensity returns from the onboard lidar to infer characteristics about the environment, the problem of in-situ mapping, where the most intense returns come from nearby points, must be addressed through range correction of the lidar. Currently, we assume a factory-calibrated range correction, which makes it difficult to detect intensity differences in rocks or nearby features from scans collected by TReX.

- *Sensors for Mapping*: Robustly estimating the trajectory of the robot for mapping would benefit from additional sensors beyond a camera, lidar, and wheel odometry. From experience, there are degenerate conditions that cause the camera and lidar to fail (e.g., navigating through very dense vegetation), which could be aided by incorporating inertial and tether measurements. Inertial measurements are useful for tethered robots because they operate on steep, inclined terrain, which means that an IMU or inclinometer can be used to extract global information about the robot's orientation with respect to gravity. Also, tether length is useful for constraining the robot's translational distance away from a known starting point.

6.4 Technology Readiness

Considering the various threads of research proposed in the thesis (e.g., design, autonomy, mapping), this section highlights progress made thus far and attempts to evaluate technology readiness. The following table highlights each research topic with respect to the listed categories. Readiness is listed as low (1), moderate (2), and high (3).

Topic	Assumptions	Recommendations	Limitations	Readiness
Design	Starts at the top, powered from the base station, initial anchor is provided.	Works well on varying terrain types (rock, grass, sand, metal, concrete) and slopes, use armored tether.	Size, rigid chassis, high mass distribution (tipping), wheel torque is too low to skid steer under high tension.	2
TSLAM	2D planar terrain, anchors are static, anchors are zero-radius points, wheel odometry is calibrated, outgoing paths are repeated.	Extend to 3D environments, account for dynamic and removed anchors, integrate with visual SLAM methods.	Only works in contrived environments, runs offline, not used for online planning, not integrated with tethered VT&R	1
VT&R	Robot will not slip, coefficient of friction is known, taught path is exactly repeatable.	Allow multiple paths on ascent, modify path tracker to use vehicle model with attached tether.	Fails if taught path is not exactly repeatable, requires external laptop computer to process features.	2
Mapping	The robot and spool are in motion, motion prior is accurate, environment is mostly static, lidar intensity is calibrated.	Decouple spool motion from lidar scanning, use a 3D lidar, incorporate IMU in cases when vision fails due to occlusion.	Maps are too sparse to be useful for geologists, lidar intensity is not calibrated for TReX, mapping is implemented offline.	2

6.5 Future Work

Looking forward for tethered robots, it will be important to integrate new autonomy to allow for unsupervised exploration of extreme environments. Consider the example scenario shown in Figure 6.1, where an operator hikes to an interesting area, sets up the base station, and deploys the robot to explore and map. In order to do this operation autonomously, the robot will start by using on-board sensors to plan a safe, tangle-free path down the terrain, where Teshnizi and Shell (2014) have proposed tethered,

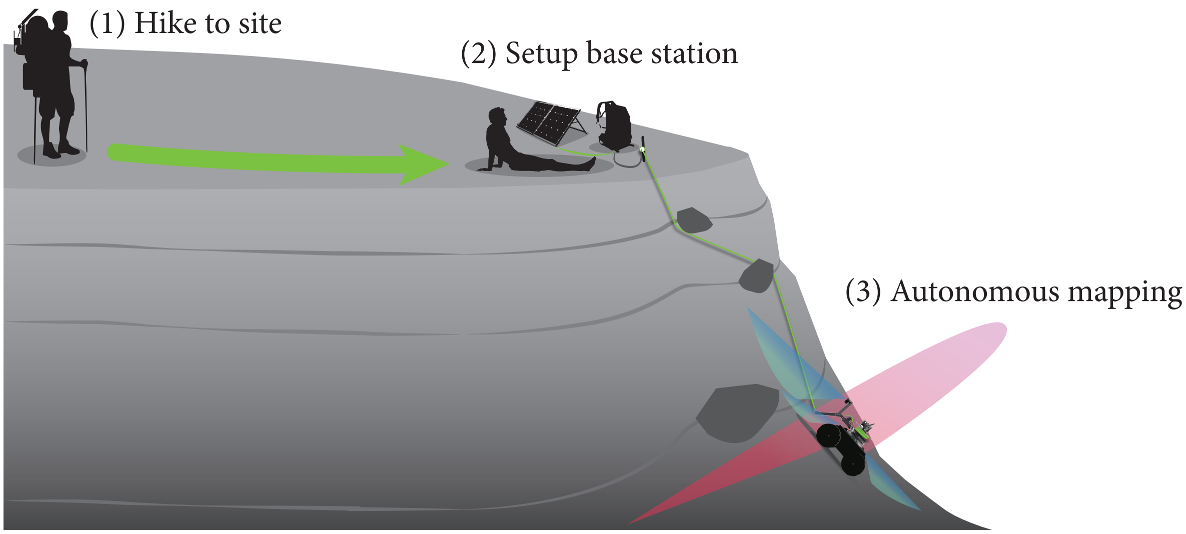


Figure 6.1: *Future Concept*: We illustrate a future concept for the autonomous mapping of cliffs with a tethered robot. In this scenario, the operator hikes to an interesting site, assembles the base station, and deploys a tethered robot, which can plan paths down the slope to expand the mapping frontier and return safely along a previously driven path.

path-planning algorithms in prior work. However, to extend path planning theory to work for real, tethered robots, we would first need to determine where the obstacles are and how the tether interacts with them. The first step could use aerial images to generate a coarse, *a priori* map, from which obstacles are determined (Abad-Manterola, 2012). Once on site, the robot can generate a dense scan of the environment, match that to the coarse map, and plan a descent path that covers as much of the terrain as possible. Next, the robot can either repeat back along the same path to unwind from added anchors (McGarey et al., 2016) or continue to explore safe paths that expand the frontier of the estimated map. It would also be beneficial to integrate autonomous path following with anchor detection, which could potentially allow for extending VT&R to work within safe, boundary regions defined by detected anchors. Most importantly, the operator can monitor mapping progress and relax while the robot autonomously explores all the accessible terrain from a given anchor location. In order to map more terrain, the operator can move the initial anchor and deploy the robot again. By leveraging off-board power, the robot can repeat this task until the user is satisfied that the targeted area has been mapped. Looking further ahead, the autonomous exploration of cliffs and caves on other planets will require additional consideration for how the robot is transported to, and anchored from, the cliff. Matthews and Nesnas (2012) show that an attached, companion robot can be used as an anchor, power supply, and communication intermediary for data transmission. This approach has the benefit that the ‘anchor’ can be moved. Alternatively, a targeted, stationary lander with an attached tethered robot could be sent to the edge of a cliff. Unfortunately, this approach leaves little margin for landing error and would only allow for a single site to be explored.

6.6 Closing Statement

I sincerely hope that the ideas and work presented in this thesis have increased the awareness, capability, and autonomy of tethered robots for exploring extreme terrain. As the paradigm of exploration shifts to more challenging environments on Earth and other planets, tethered robots are poised to offer advantages over other systems due to their ability to access hard-to-reach places and harness off-board resources.

Appendix A

Design Extras

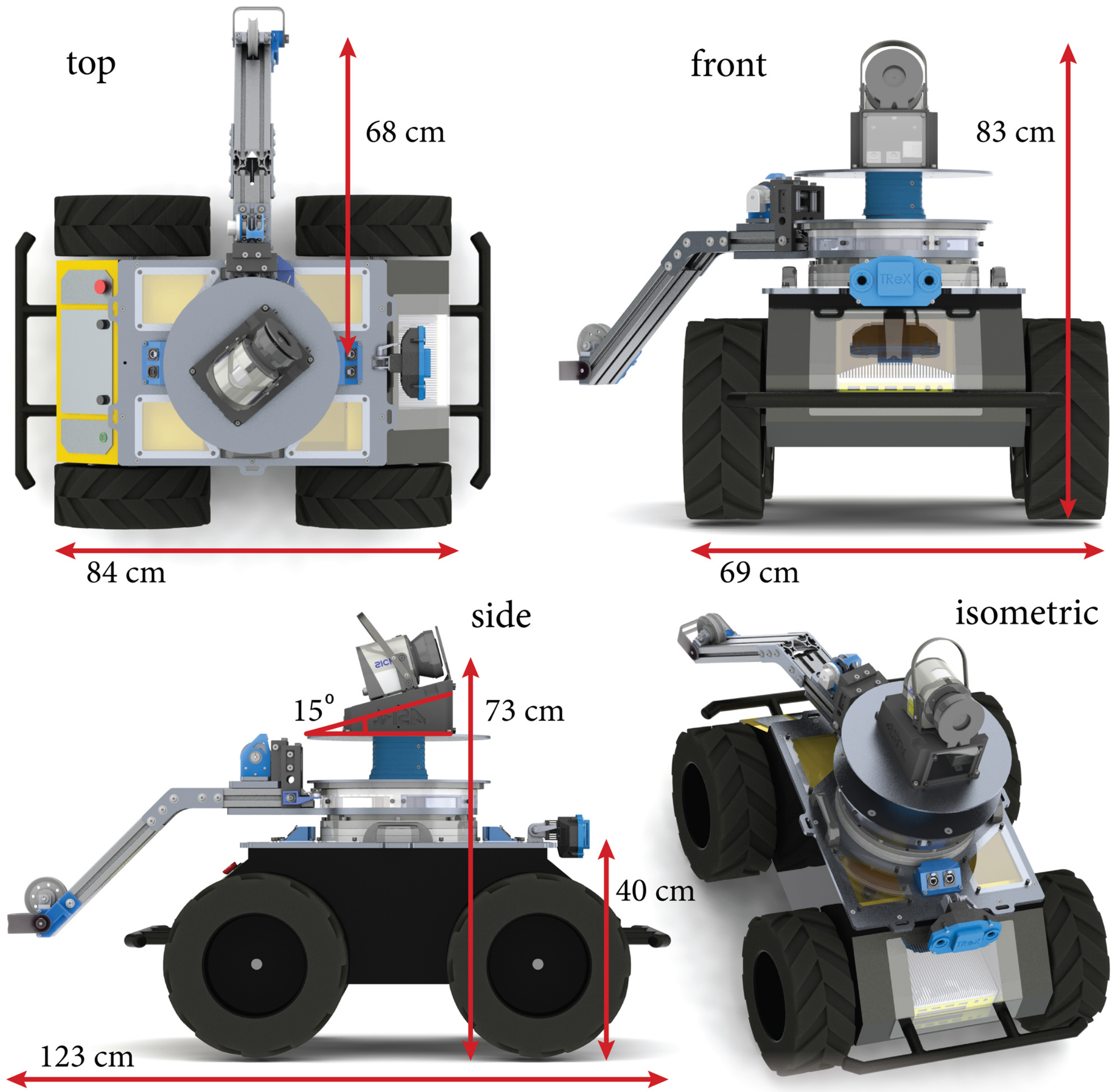


Figure A.1: *System Dimensions.*

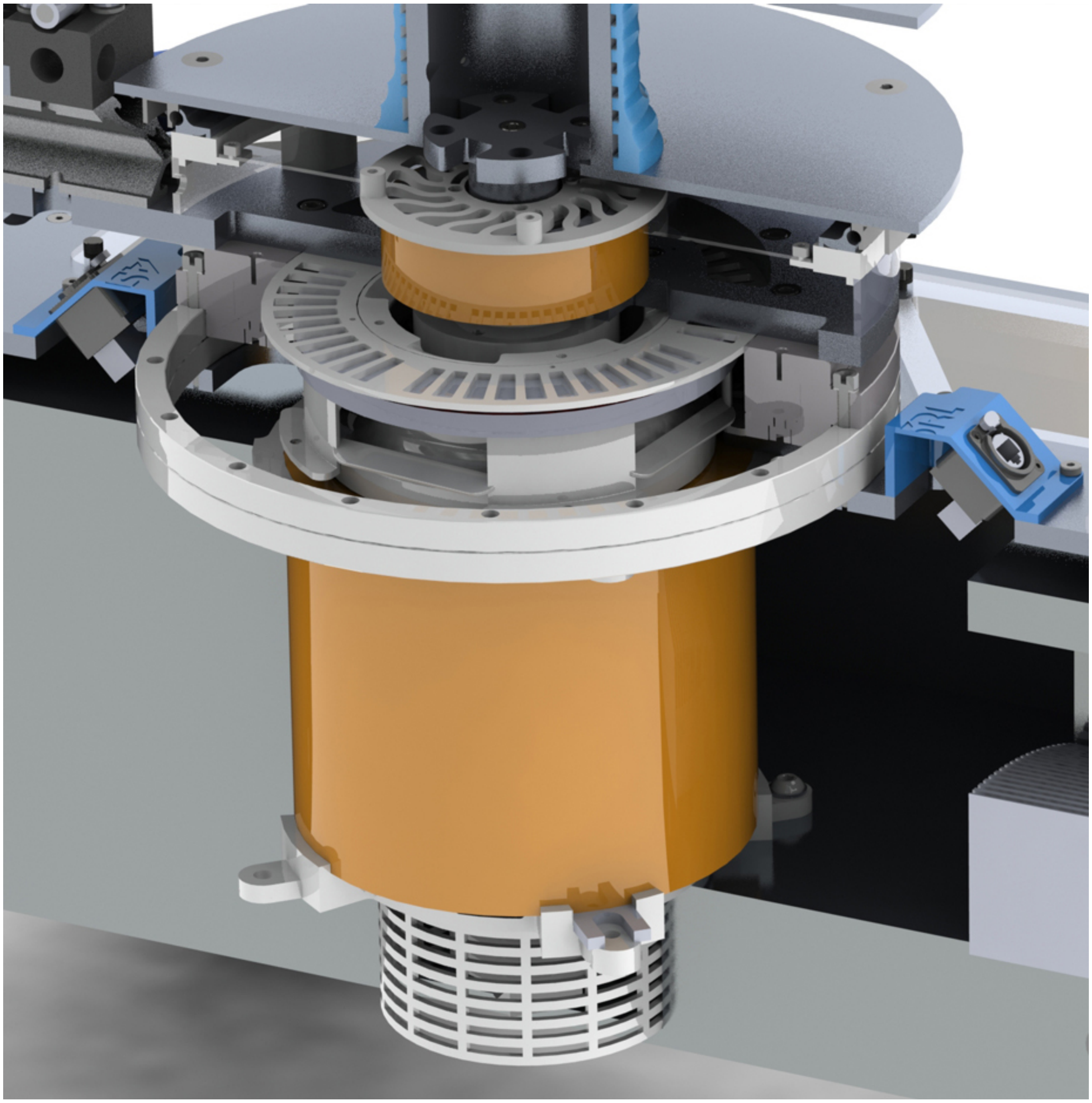


Figure A.2: *Rotational Elements:* This cross-section of the TReX design highlights the complexity of the rotational architecture, which allows the tether arm to passively rotate while also actuating a top-mounted spool/lidar. Slip rings (for electrical transmission between rotating elements) are shown in orange, 3D-printed, plastic parts are shown in blue and white, and metal parts are grey.

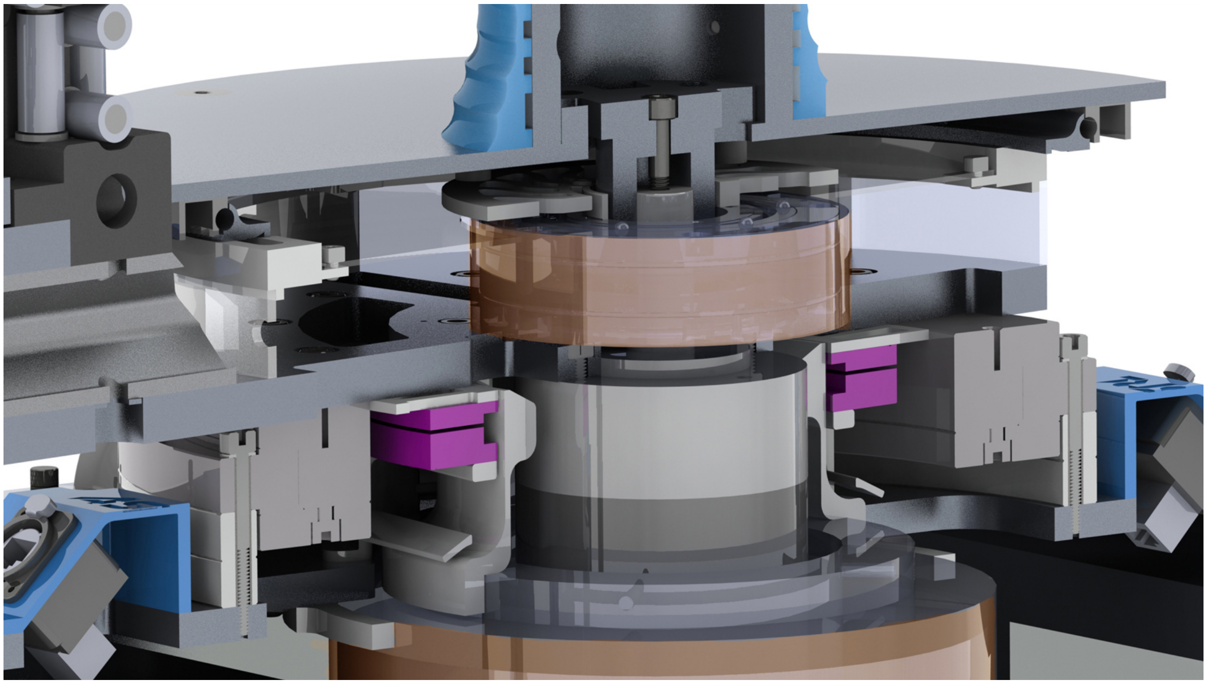


Figure A.3: *Rotational elements (close-up view).*

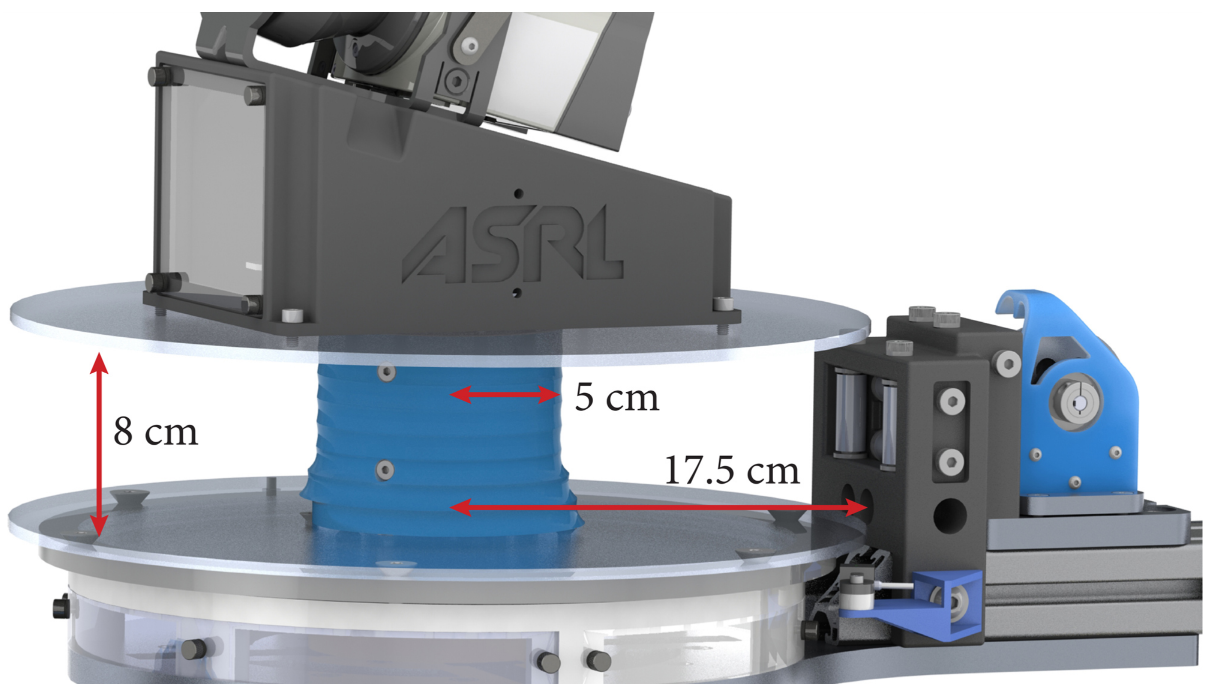


Figure A.4: *Spooling system:* The capacity is ∼45 m with a tether diameter of 9.4 mm.

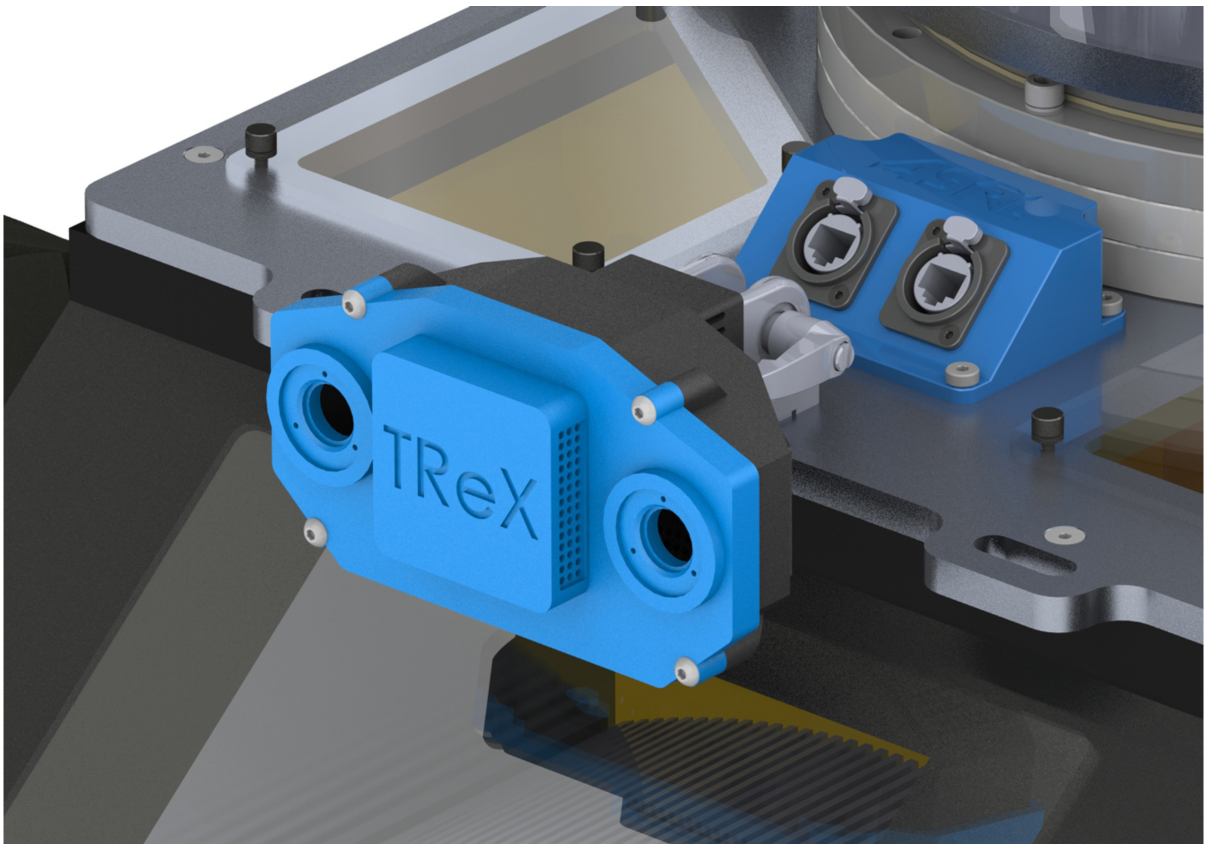


Figure A.5: *Stereo Camera Mount:* This 3D-printed camera case houses a Skybotix VI-Sensor. Weatherized cables from the camera connect to the access panel shown.

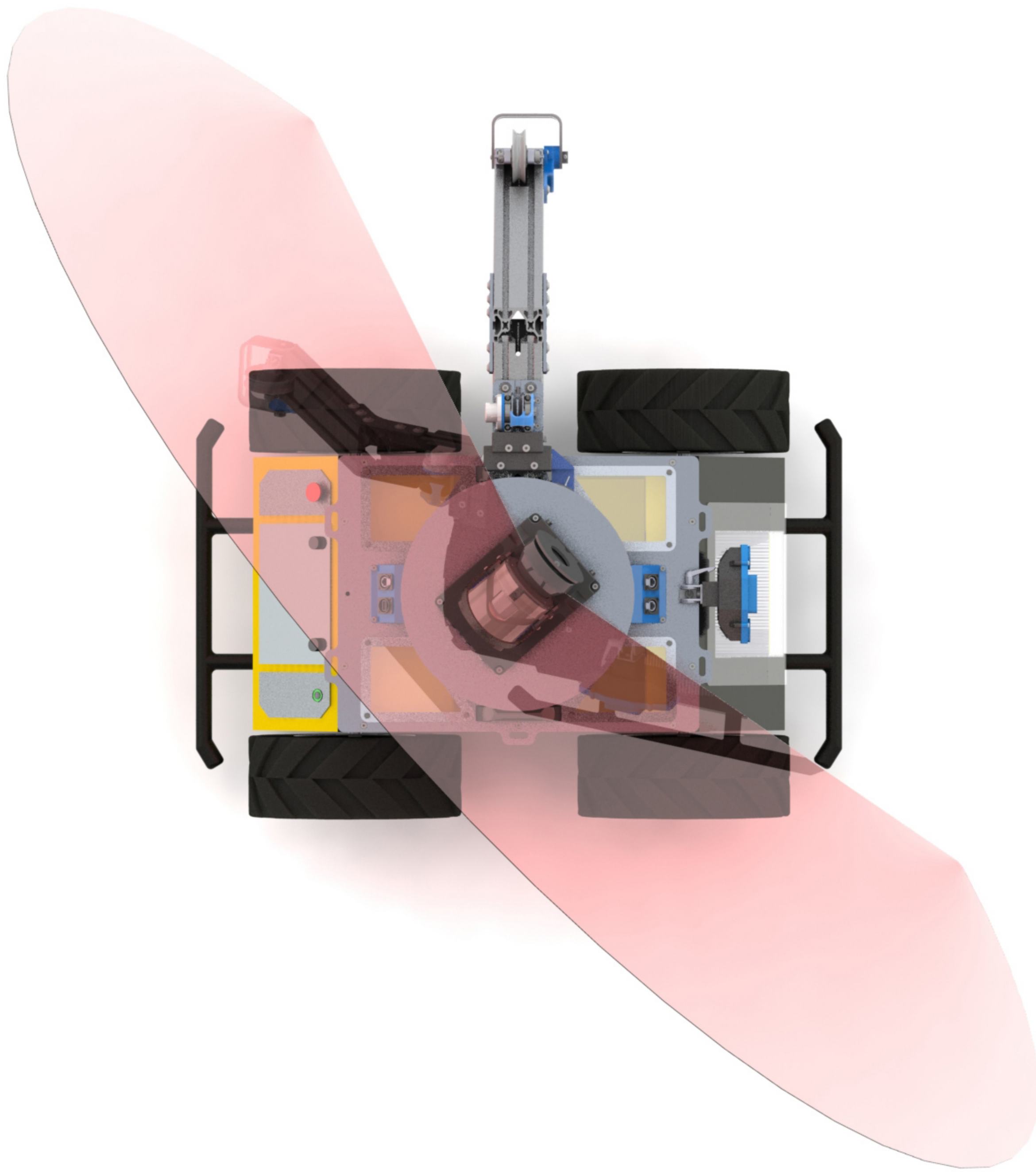


Figure A.6: *Lidar Orientation:* A top view of the scan plane for the 2D lidar is shown. The lidar is angled by 15° to increase scan overlap; the angled plane will produce a cross-hatching effect for a 3D scan collected during a single rotation.

Appendix B

TSLAM Extras

This appendix contains additional math formations for the general TSLAM problem and proposed particle filter and batch solutions presented in Chapter 3. We also provide supplemental results to illustrate the particle filter as a time-lapse animation and show a visual example of the batch segmentation and association processes.

B.1 Expanded Math

General TSLAM

Robot trajectory:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \tag{B.1}$$

Body-centric velocities:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} \upsilon_{x,k} \\ \upsilon_{y,k} \\ \omega_k \end{bmatrix}, \tag{B.2}$$

Motion model:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k), \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_0), \quad \mathbf{Q}_0 = \begin{bmatrix} \sigma_{\upsilon_x}^2 & & \\ & \sigma_{\upsilon_y}^2 & \\ & & \sigma_{\omega}^2 \end{bmatrix} \tag{B.3}$$

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + T \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} v_{x,k} \\ 0 \\ \omega_k \end{bmatrix} + \begin{bmatrix} w_{v_{x,k}} \\ w_{v_{y,k}} \\ w_{\omega_k} \end{bmatrix} \right), \tag{B.4}$$

where T is a sample rate (e.g., 10 Hz)

Tether measurements:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} d_k \\ \phi_k \end{bmatrix}, \tag{B.5}$$

Anchor List (two different representations are used):

$$\boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_1 \\ \vdots \\ \boldsymbol{\ell}_N \end{bmatrix}, \quad \boldsymbol{\ell}_n \rightarrow \underbrace{\mathcal{N}\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}, \begin{bmatrix} \sigma_d^2 & \\ & \sigma_\phi^2 \end{bmatrix} \right)}_{\text{particle filter (see next section)}}, \quad \boldsymbol{\ell}_n = \underbrace{\begin{bmatrix} x_n \\ y_n \\ d_{\text{fix},\boldsymbol{\ell}_n} \end{bmatrix}}_{\text{batch method}}, \tag{B.6}$$

where $d_{\text{fix},\boldsymbol{\ell}_n}$ is the fixed tether length for current anchor, $\boldsymbol{\ell}_n$.

Measurement model:

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n) + \mathbf{n}_k, \quad \mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_0), \quad \mathbf{R}_0 = \begin{bmatrix} \sigma_d^2 & \\ & \sigma_\phi^2 \end{bmatrix} \tag{B.7}$$

$$\begin{bmatrix} d_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{\ell}_1 - \boldsymbol{\ell}_2\| + \ldots + \|\boldsymbol{\ell}_{n-1} - \boldsymbol{\ell}_n\| + \left\|\boldsymbol{\ell}_n - \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right]\right\| \\ \operatorname{atan2}(\ell_{y,n} - y_k, \ell_{x,n} - x_k) - \theta_k \end{bmatrix} + \begin{bmatrix} n_{d,k} \\ n_{\phi,k} \end{bmatrix} \tag{B.8}$$

Pose errors:

$$\mathbf{e}_{\mathbf{x},k}(\mathbf{x}) = \begin{cases} \mathbf{0} & k = 0 \\ \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}) - \mathbf{x}_k & k = 1 \ldots K \end{cases} \tag{B.9}$$

Measurement errors:

$$\mathbf{e}_{\mathbf{y},k}(\mathbf{x}, \boldsymbol{\ell}) = \mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}), \quad k = 1 \ldots K \tag{B.10}$$

Particle Filter TSLAM

Inverse measurement model (initializing the mean, $\boldsymbol{\mu}_{n+1}$, of Gaussian, $\boldsymbol{\ell}_{n+1}$):

$$\mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n) = \boldsymbol{\mu}_{n+1} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - d_{\text{free},k+1} \begin{bmatrix} \cos(\phi_k + \theta_k) \\ \sin(\phi_k + \theta_k) \end{bmatrix} \tag{B.11}$$

Free length to a newly detected anchor:

$$d_{\text{free},k+1} = \frac{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - (d_{\text{free},k})^2}{2(d_{\text{free},k}) - \left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos\psi_k}, \tag{B.12}$$

where ψ_k is the bearing-to-anchor measurement error.

Linearized inverse measurement model w.r.t. tether measurements:

$$\mathbf{G}_k^{-1} = \frac{\partial \mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \mathbf{y}_k} = \begin{bmatrix} \frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} & \frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} \\ \frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} & \frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} \end{bmatrix} \tag{B.13}$$

$$\frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} = \frac{2(\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - d_{\text{free},k+1}^2)\cos(\phi_k + \theta_k)}{(2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k))^2} + \frac{2d_{\text{free},k+1}\cos(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k)}$$

$$\frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} = \frac{2(\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - d_{\text{free},k+1}^2)\sin(\phi_k + \theta_k)}{(2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k))^2} + \frac{2d_{\text{free},k+1}\sin(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k)}$$

$$\frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} = \frac{(\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - d_{\text{free},k+1}^2)\sin(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k)} - \frac{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| (\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - d_{\text{free},k+1}^2)\cos(\phi_k + \theta_k)\sin(\psi_k)}{(2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k))^2}$$

$$\frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} = -\frac{(\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - d_{\text{free},k+1}^2)\cos(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k)} - \frac{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| (\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2 - d_{\text{free},k+1}^2)\sin(\phi_k + \theta_k)\sin(\psi_k)}{(2d_{\text{free},k+1} - 2\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\| \cos(\psi_k))^2}$$

Linearized measurement model w.r.t. anchors:

$$\mathbf{G}_k = \frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n)}{\partial \boldsymbol{\mu}_n} = \begin{bmatrix} \frac{\partial g_1}{\partial \mu_{x,n}} & \frac{\partial g_1}{\partial \mu_{y,n}} \\ \frac{\partial g_2}{\partial \mu_{x,n}} & \frac{\partial g_2}{\partial \mu_{y,n}} \end{bmatrix} = \begin{bmatrix} \frac{\mu_{x,n} - x_k}{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|} & \frac{\mu_{y,n} - y_k}{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|} \\ \frac{-\mu_{y,n} + y_k}{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2} & \frac{\mu_{x,n} - x_k}{\left\| \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right] - \boldsymbol{\mu}_n \right\|^2} \end{bmatrix} \tag{B.14}$$

Initializing an anchor:

1) $\boldsymbol{\mu}_{n+1} = \mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)$ $\Leftarrow$ intitialze the mean of Gaussian $\boldsymbol{\ell}_{n+1}$

2) $\mathbf{G}_k^{-1} = \dfrac{\partial \mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \mathbf{y}_k}$ $\Leftarrow$ Jacobian of inverse measurement model

3) $\boldsymbol{\Sigma}_{n+1} = \mathbf{G}_k^{-1}\mathbf{R}_0(\mathbf{G}_k^{-1})^T$ $\Leftarrow$ initialize measurement covariance

4) $w_k = w_0$ $\Leftarrow$ initialize weight

5) $\boldsymbol{\ell}_{n+1} \to \mathcal{N}(\boldsymbol{\mu}_{n+1}, \boldsymbol{\Sigma}_{n+1})$ $\Leftarrow$ Gaussian with anchor position and covariance

Update an anchor:

1) $\hat{\mathbf{y}}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n)$ $\Leftarrow$ estimated measurement

2) $\mathbf{G}_k = \dfrac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n)}{\partial \boldsymbol{\mu}_n}$ $\Leftarrow$ Jacobian of measurement model

3) $\mathbf{R}_k = \mathbf{G}_k \boldsymbol{\Sigma}_n \mathbf{G}_k^T + \mathbf{R}_0$ $\Leftarrow$ update measurement uncertainty

4) $w_k = |2\pi \mathbf{R}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{y}_k - \hat{\mathbf{y}}_k)^T \mathbf{R}_k^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_k)\right\}$ $\Leftarrow$ update weight

5) $\mathbf{K}_k = \boldsymbol{\Sigma}_n \mathbf{G}_k^T \mathbf{R}_k^{-1}$ $\Leftarrow$ Kalman gain

6) $\boldsymbol{\mu}_n = \boldsymbol{\mu}_n + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$ $\Leftarrow$ update anchor position

7) $\boldsymbol{\Sigma}_n = (\mathbf{1} - \mathbf{K}_k \mathbf{G}_k)\boldsymbol{\Sigma}_n$ $\Leftarrow$ update covariance

8) $\boldsymbol{\ell}_n \to \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ $\Leftarrow$ update Gaussian

Removing an anchor: *Remove from the list and then perform an update (as above) on the next anchor in the list.*

Batch TSLAM

Linear System to Solve:

$$\underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \delta \mathbf{x}^\star \\ \delta \boldsymbol{\ell}^\star \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}}_{\mathbf{b}}. \tag{B.15}$$

Components of $\mathbf{A}$ (Hessian):

$$\mathbf{A} = \mathbf{H}^T \mathbf{W}^{-1} \mathbf{H} = \begin{bmatrix} \mathbf{F}^{-T}\mathbf{Q}^{-1}\mathbf{F}^{-} + \mathbf{G}_1^{-T}\mathbf{R}^{-1}\mathbf{G}_1 & \mathbf{G}_1^{-T}\mathbf{R}^{-1}\mathbf{G}_2 \\ \mathbf{G}_2^{-T}\mathbf{R}^{-1}\mathbf{G}_1 & \mathbf{G}_2^{-T}\mathbf{R}^{-1}\mathbf{G}_2 \end{bmatrix} \tag{B.16}$$

Components of $\mathbf{b}$ (Jacobian):

$$\mathbf{b} = \mathbf{H}^T \mathbf{W}^{-1} \mathbf{e}(\mathbf{x}^\star, \boldsymbol{\ell}^\star), \quad \mathbf{e}(\mathbf{x}^\star, \boldsymbol{\ell}^\star) = \begin{bmatrix} \mathbf{e}_{\mathbf{x}}(\mathbf{x}^\star) \\ \mathbf{e}_{\mathbf{y}}(\mathbf{x}^\star, \boldsymbol{\ell}^\star) \end{bmatrix} \tag{B.17}$$

Components of $\mathbf{H}$ and $\mathbf{W}^{-1}$:

$$\mathbf{H} = \begin{bmatrix} \mathbf{F}^{-1} & \mathbf{0} \\ \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix}, \quad \mathbf{W}^{-1} = \begin{bmatrix} \mathbf{Q}^{-1} & \\ & \mathbf{R}^{-1} \end{bmatrix} \tag{B.18}$$

Linearized motion model (inverse):

$$\mathbf{F}^{-1} = \begin{bmatrix} \mathbf{1} & & & & \\ -\frac{\partial \mathbf{f}(\mathbf{x}_0, \mathbf{v}_1, \mathbf{w}_1)}{\partial \mathbf{x}_0} & \mathbf{1} & & & \\ & -\frac{\partial \mathbf{f}(\mathbf{x}_1, \mathbf{v}_2, \mathbf{w}_2)}{\partial \mathbf{x}_1} & \ddots & & \\ & & \ddots & \mathbf{1} & \\ & & & -\frac{\partial \mathbf{f}(\mathbf{x}_{K-1}, \mathbf{v}_K, \mathbf{w}_K)}{\partial \mathbf{x}_{K-1}} & \mathbf{1} \end{bmatrix} \tag{B.19}$$

Motion model Jacobian:

$$\left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k)}{\partial \mathbf{x}_{k-1}}\right|_{\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_{k-1}} & \frac{\partial f_1}{\partial y_{k-1}} & \frac{\partial f_1}{\partial \theta_{k-1}} \\ \frac{\partial f_2}{\partial x_{k-1}} & \frac{\partial f_2}{\partial y_{k-1}} & \frac{\partial f_2}{\partial \theta_{k-1}} \\ \frac{\partial f_3}{\partial x_{k-1}} & \frac{\partial f_3}{\partial y_{k-1}} & \frac{\partial f_3}{\partial \theta_{k-1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T(\sin\theta_{k-1} v_k) \\ 0 & 1 & T(\cos\theta_{k-1} v_k) \\ 0 & 0 & 1 \end{bmatrix} \tag{B.20}$$

Linearized motion noise (inverse):

$$\mathbf{Q}^{-1} = \begin{bmatrix} \mathbf{Q}_0^{-1} & & & & \\ & (\mathbf{F}_{\mathbf{w},1}\mathbf{Q}_0\mathbf{F}_{\mathbf{w},1}^T)^{-1} & & & \\ & & (\mathbf{F}_{\mathbf{w},2}\mathbf{Q}_0\mathbf{F}_{\mathbf{w},2}^T)^{-1} & & \\ & & & \ddots & \\ & & & & (\mathbf{F}_{\mathbf{w},K-1}\mathbf{Q}_0\mathbf{F}_{\mathbf{w},K-1}^T)^{-1} \end{bmatrix} \tag{B.21}$$

Motion noise Jacobian:

$$\mathbf{F}_{\mathbf{w},k} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k)}{\partial \mathbf{w}_k}\right|_{\mathbf{x}_{k-1},\mathbf{v}_k,\mathbf{w}_k} = \quad \cdots \tag{B.22}$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial w_{v_{x,k}}} & \frac{\partial f_1}{\partial w_{v_{y,k}}} & \frac{\partial f_1}{\partial w_{\omega_k}} \\ \frac{\partial f_2}{\partial w_{v_{x,k}}} & \frac{\partial f_2}{\partial w_{v_{y,k}}} & \frac{\partial f_2}{\partial w_{\omega_k}} \\ \frac{\partial f_3}{\partial w_{v_{x,k}}} & \frac{\partial f_3}{\partial w_{v_{y,k}}} & \frac{\partial f_3}{\partial w_{\omega_k}} \end{bmatrix} = \begin{bmatrix} T\cos\theta_{k-1} & -T\sin\theta_{k-1} & 0 \\ T\sin\theta_{k-1} & T\cos\theta_{k-1} & 0 \\ 0 & 0 & T \end{bmatrix}$$

Linearized measurement model w.r.t. robot pose:

$$\mathbf{G}_1 = \begin{bmatrix} \frac{\partial \mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial \mathbf{x}_1} & & & \\ & \frac{\partial \mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial \mathbf{x}_2} & & \\ & & \ddots & \\ & & & \frac{\partial \mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial \mathbf{x}_K} \end{bmatrix} \tag{B.23}$$

Measurement model Jacobian w.r.t. pose:

$$\left.\frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell})}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k,\boldsymbol{\ell},\mathbf{0}} = \begin{bmatrix} \frac{\partial g_1}{\partial x_k} & \frac{\partial g_1}{\partial y_k} & \frac{\partial g_1}{\partial \theta_k} \\ \frac{\partial g_2}{\partial x_k} & \frac{\partial g_2}{\partial y_k} & \frac{\partial g_2}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} \frac{-\ell_{x,n}+x_k}{\left\|\boldsymbol{\ell}_n - \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right]\right\|} & \frac{-\ell_{y,n}+y_k}{\left\|\boldsymbol{\ell}_n - \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right]\right\|} & 0 \\ \frac{\ell_{y,n}-y_k}{\left\|\boldsymbol{\ell}_n - \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right]\right\|^2} & \frac{-\ell_{x,n}+x_k}{\left\|\boldsymbol{\ell}_n - \left[\begin{smallmatrix} x_k \\ y_k \end{smallmatrix}\right]\right\|^2} & -1 \end{bmatrix} \tag{B.24}$$

Linearized measurement model w.r.t. anchors:

$$\mathbf{G}_2 = \begin{bmatrix} \frac{\partial \mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}} \\ \frac{\partial \mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}} \\ \vdots \\ \frac{\partial \mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_1} & \frac{\partial \mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_2} & \cdots & \frac{\partial \mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_{N-1}} & \frac{\partial \mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_N} \\ \frac{\partial \mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_1} & \frac{\partial \mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_2} & \cdots & \frac{\partial \mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_{N-1}} & \frac{\partial \mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_N} \\ \vdots & \vdots & & \vdots & \vdots \\ \frac{\partial \mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_1} & \frac{\partial \mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_2} & \cdots & \frac{\partial \mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_{N-1}} & \frac{\partial \mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_N} \end{bmatrix} \tag{B.25}$$

Measurement model Jacobian w.r.t. anchors:

$$\left.\frac{\partial \mathbf{g}(\mathbf{x}_k,\boldsymbol{\ell})}{\partial \boldsymbol{\ell}_n}\right|_{\mathbf{x}_k,\boldsymbol{\ell},\mathbf{0}} = \begin{bmatrix} \frac{\partial g_1}{\partial \ell_{x,n}} & \frac{\partial g_1}{\partial \ell_{y,n}} & \frac{\partial g_1}{\partial d_{\text{fix},\boldsymbol{\ell}_n}} \\ \frac{\partial g_2}{\partial \ell_{x,n}} & \frac{\partial g_2}{\partial \ell_{y,n}} & \frac{\partial g_2}{\partial d_{\text{fix},\boldsymbol{\ell}_n}} \end{bmatrix} = \begin{bmatrix} \frac{\ell_{x,n}-x_k}{\left\|\boldsymbol{\ell}_n-\left[\begin{smallmatrix}x_k\\y_k\end{smallmatrix}\right]\right\|} & \frac{\ell_{y,n}-y_k}{\left\|\boldsymbol{\ell}_n-\left[\begin{smallmatrix}x_k\\y_k\end{smallmatrix}\right]\right\|} & 1 \\ \frac{-\ell_{y,n}+y_k}{\left\|\boldsymbol{\ell}_n-\left[\begin{smallmatrix}x_k\\y_k\end{smallmatrix}\right]\right\|^2} & \frac{\ell_{x,n}-x_k}{\left\|\boldsymbol{\ell}_n-\left[\begin{smallmatrix}x_k\\y_k\end{smallmatrix}\right]\right\|^2} & 0 \end{bmatrix} \tag{B.26}$$

Linearized measurement noise (inverse):

$$\mathbf{R}^{-1} = \begin{bmatrix} (\mathbf{G}_{\mathbf{n}_1}\mathbf{R}_0\mathbf{G}_{\mathbf{n}_1}^T)^{-1} & & & \\ & (\mathbf{G}_{\mathbf{n}_2}\mathbf{R}_0\mathbf{G}_{\mathbf{n}_2}^T)^{-1} & & \\ & & \ddots & \\ & & & (\mathbf{G}_{\mathbf{n}_K}\mathbf{R}_0\mathbf{G}_{\mathbf{n}_K}^T)^{-1} \end{bmatrix} \tag{B.27}$$

Measurement noise Jacobian:

$$\mathbf{G}_{\mathbf{n}_k} = \frac{\partial \mathbf{g}(\mathbf{x}_k,\boldsymbol{\ell})}{\partial \mathbf{n}_k} = \begin{bmatrix} \frac{\partial g_1}{\partial n_{d,k}} & \frac{\partial g_1}{\partial n_{\phi,k}} \\ \frac{\partial g_2}{\partial n_{d,k}} & \frac{\partial g_2}{\partial n_{\phi,k}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{B.28}$$

Switchable constraint error: $*$ and $**$ represent two arbitrary anchors

$$e_c(\boldsymbol{\ell}_*) = (d_{\text{fix},\boldsymbol{\ell}_*} - d_{\text{fix},\boldsymbol{\ell}_{**}}) - \|\boldsymbol{\ell}_* - \boldsymbol{\ell}_{**}\| \tag{B.29}$$

Linearized switchable constraint noise:

$$\mathbf{S}^{-1} = \begin{bmatrix} S_0^{-1} & & \\ & \ddots & \\ & & S_0^{-1} \end{bmatrix}, \quad S_0 = \sigma_{e_c}^2 \tag{B.30}$$

Dynamic Covariance Scaling:

$$\hat{\mathbf{S}}^{-1} = w(\mathbf{e}_c)\mathbf{S}^{-1}, \quad \text{where } \mathbf{e}_c \text{ is a vector of constraint errors} \tag{B.31}$$

$$w(\mathbf{e}_c) = \begin{cases} e_c^2 \leq s & \Big| \quad 1 \\ e_c^2 > s & \Big| \quad \frac{4s^2}{(s+e_c^2)^2} \end{cases}, \quad \text{where } w \to [0,1] \tag{B.32}$$

Lower Cholesky Decomposition (for efficiently solving the batch solution):

$$\begin{aligned}
\mathbf{A} &= \mathbf{L}\mathbf{L}^T, \\
\mathbf{L}\mathbf{p} &= \mathbf{b} \quad \text{(solve for } \mathbf{p}\text{)}, \\
\mathbf{L}^T\delta\mathbf{z} &= \mathbf{p} \quad \text{(solve for } \delta\mathbf{z}\text{)}, \\
\mathbf{L}^T\delta\mathbf{z} &= \mathbf{L}^{-1}\mathbf{b}
\end{aligned} \tag{B.33}$$

$$\underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{12}^T & \mathbf{L}_{22} \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{0} & \mathbf{L}_{22}^T \end{bmatrix}}_{\mathbf{L}^T} = \begin{bmatrix} \mathbf{L}_{11}^2 & \mathbf{L}_{11}\mathbf{L}_{12} \\ \mathbf{L}_{11}\mathbf{L}_{12}^T & \mathbf{L}_{12}\mathbf{L}_{12}^T + \mathbf{L}_{22}\mathbf{L}_{22}^T \end{bmatrix}$$

$$\mathbf{L}^{-1} = \begin{bmatrix} \mathbf{L}_{11}^{-1} & \mathbf{0} \\ -\mathbf{L}_{11}^{-1}\mathbf{L}_{12}^T\mathbf{L}_{22}^{-1} & \mathbf{L}_{22}^{-1} \end{bmatrix}$$

B.2 Additional Results

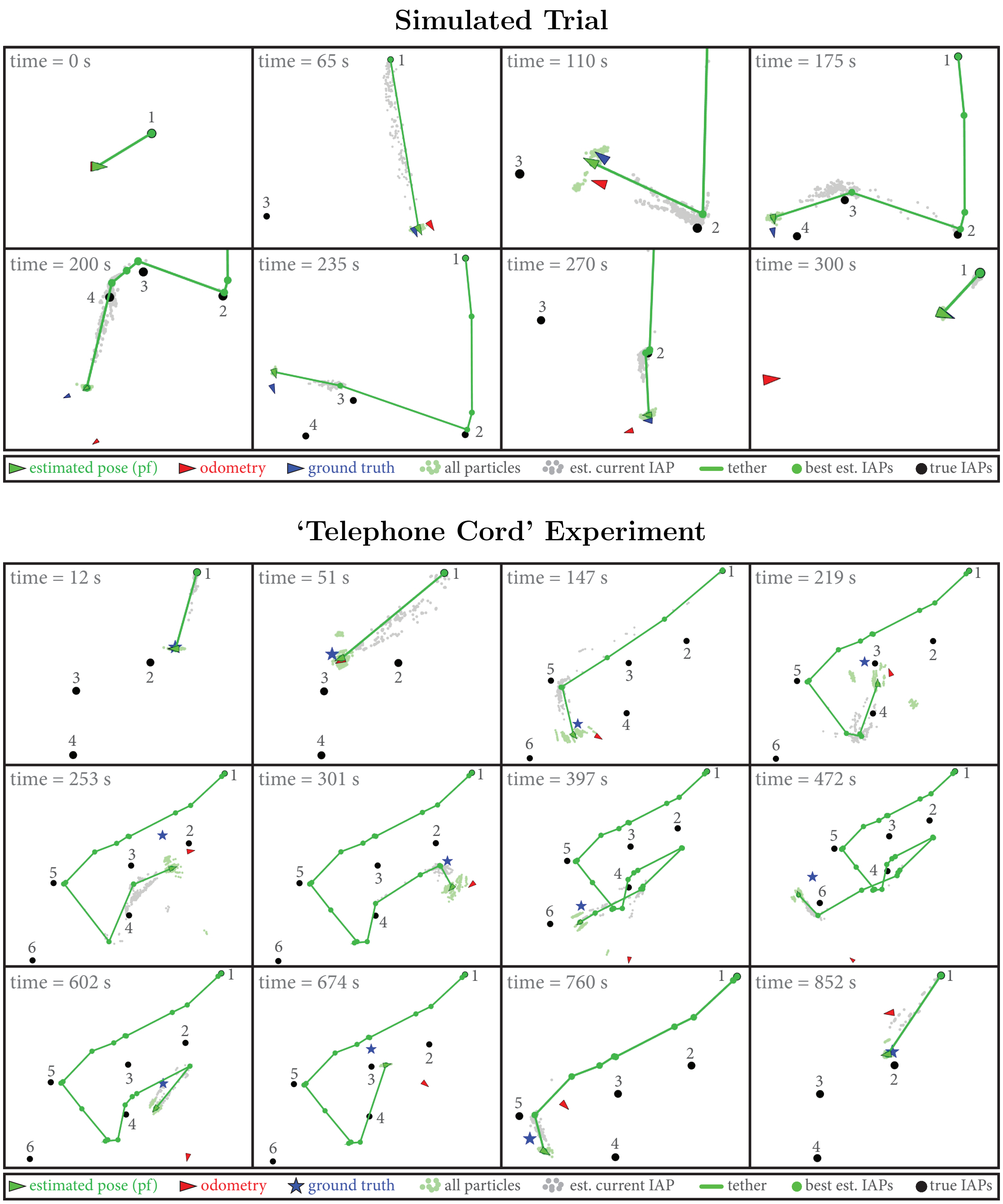


Figure B.1: *Particle Filter Time-Lapse Animation*: Time-lapse sequences illustrate the particle filter running on the simulated trial and 'Telephone Cord' experiment datasets from Chapter 3. Particles represent estimates of the robot's pose, each with their own map of anchors (IAPs). The best estimate for the robot's pose is calculated as the mean from the highest weighted particles at any given time. We use the most probable particle's map and illustrate its tether configuration by connecting a line though its anchors.

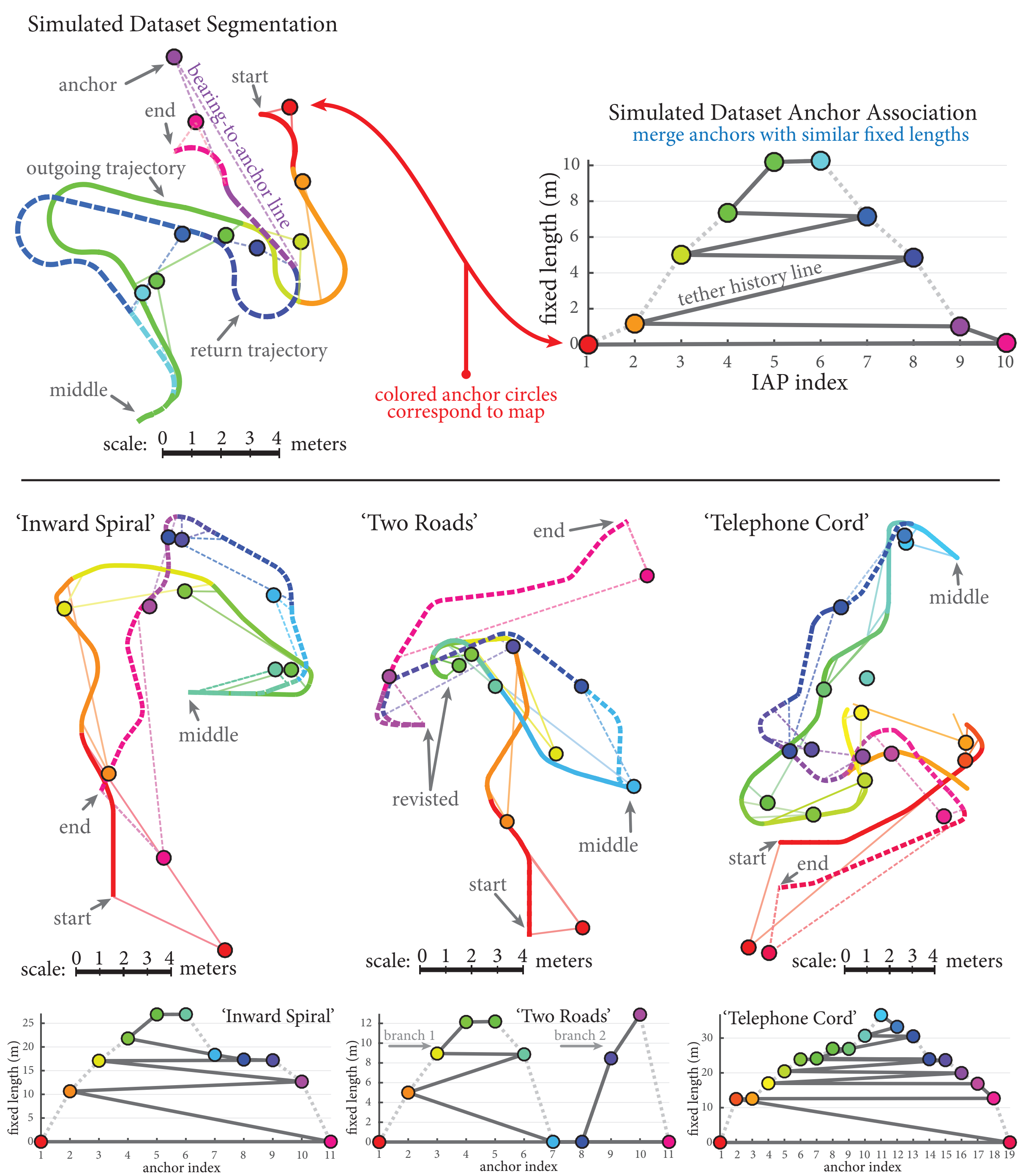


Figure B.2: *Batch Segmentation and Association*: We illustrate the anchor association and trajectory segmentation steps for the simulated and experimental datasets from Chapter 3. *anchor Association*: The fixed-length tree is used to determine tether history by associating anchors with similar fixed tether lengths that should be merged in the map (Section 3.3.2 provides details). *Segmentation*: The trajectory from odometry is divided into subtrajectories, each associated to a single anchor of matching color using the RANSAC method described in Section 3.3.2. Solid and dashed lines represent outgoing and return trajectories. Dashed, colorized bearing-to-anchor lines are shown.

Bibliography

Abad-Manterola, P. (2012). *Axel rover tethered dynamics and motion planning on extreme planetary terrain*. PhD thesis, California Institute of Technology. (ref. pages 61 and 106)

Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W. (2013). Robust map optimization using dynamic covariance scaling. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 62–69. (ref. pages 38 and 42)

Ahn, H.-S., Na, S.-I., Lee, Y.-C., and Yu, W. (2006). A controller design of a tethered-robot guiding system. In *International Conference on Ubiquitous Robots and Ambient Intelligence*, pages 43–46. Citeseer. (ref. page 61)

Anderson, S. and Barfoot, T. D. (2015). Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on se (3). In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 157–164. (ref. pages 85 and 87)

Apostolopoulos, D. and Bares, J. (1995). Locomotion configuration of a robust rappelling robot. In *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 280–284. (ref. page 61)

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117. (ref. page 25)

Bainum, P. M. and Kumar, V. (1980). Optimal control of the shuttle-tethered-subsatellite system. *Acta Astronautica*, 7(12):1333–1348. (ref. page 61)

Bares, J. E. and Wettergreen, D. S. (1999). Dante II: Technical description, results, and lessons learned. *The International Journal of Robotics Research*, 18(7):621–649. (ref. page 10)

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359. (ref. page 71)

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517. (ref. page 40)

Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296. (ref. page 61)

Booij, O., Terwijn, B., Zivkovic, Z., and Krose, B. (2007). Navigation using an appearance based topological map. In *2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3927–3932. (ref. page 61)

Bosse, M. and Zlot, R. (2009). Continuous 3D scan-matching with a spinning 2D laser. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4312–4319. (ref. page 85)

Britton, N., Yoshida, K., Walker, J., Nagatani, K., Taylor, G., and Dauphin, L. (2015). Lunar micro rover design for exploration through virtual reality tele-operation. In *Field and Service Robotics*, pages 259–272. Springer. (ref. page 10)

Brown, D. C. (1958). A solution to the general problem of multiple station analytical stereotriangulation. Technical report, RCA-MTP Data Reduction Technical Report No. 43 (or AFMTC TR 58-8). Patrick Airforce Base, Florida. (ref. page 25)

Choset, H. M. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press. (ref. page 60)

Chum, O., Matas, J., and Kittler, J. (2003). Locally optimized RANSAC. In *Pattern Recognition*, volume 2781, pages 236–243. Springer. (ref. page 39)

Cole, D. M. and Newman, P. M. (2006). Using laser range data for 3D SLAM in outdoor environments. In *2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1556–1563. (ref. page 85)

Corominas-Murtra, A. and Tur, J. M. M. (2013). Imu and cable encoder data fusion for in-pipe mobile robot localization. In *2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6. (ref. page 25)

Dereniowski, D. and Kubale, M. (2004). Cholesky factorization of matrices in parallel and ranking of graphs. *Parallel Processing and Applied Mathematics*, pages 985–992. (ref. page 35)

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part I. *Robotics & Automation Magazine, IEEE*, 13(2):99–110. (ref. pages 25 and 85)

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395. (ref. page 38)

Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560. (ref. pages 59 and 61)

Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3946–3952. (ref. page 37)

Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2011). Visual odometry and mapping for autonomous flight using an RGB-D camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16. (ref. page 87)

Huntsberger, T., Stroupe, A., Aghazarian, H., Garrett, M., Younse, P., and Powell, M. (2007). TRESSA: Teamed robots for exploration and science on steep areas. *Journal of Field Robotics*, 24(11-12):1015–1031. (ref. pages 2, 9, and 10)

Huntsberger, T. L., Trebi-Ollennu, A., Aghazarian, H., Schenker, P. S., Pirjanian, P., and Nayar, H. D. (2004). Distributed control of multi-robot systems engaged in tightly coupled tasks. *Autonomous Robots*, 17(1):79–92. (ref. page 61)

Iqbal, J., Heikkila, S., and Halme, A. (2008). Tether tracking and control of ROSA robotic rover. In *2008, 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 689–693. (ref. page 61)

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2011). iSAM2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235. (ref. pages 53 and 57)

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *Robotics, IEEE Transactions on*, 24(6):1365–1378. (ref. pages 53 and 57)

Krishna, M., Bares, J., and Mutschler, E. (1997). Tethering system design for Dante II. In *1997 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1100–1105. (ref. page 61)

Kubelka, V., Oswald, L., Pomerleau, F., Colas, F., Svoboda, T., and Reinstein, M. (2015). Robust data fusion of multimodal sensory information for mobile robots. *Journal of Field Robotics*, 32(4):447–473. (ref. page 85)

Kumar, T. R. and Richardson, R. C. (2008). Tether monitoring techniques for environment monitoring, tether following and localization of autonomous mobile robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2109–2114. (ref. page 25)

Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349. (ref. page 25)

MacTavish, K. and Barfoot, T. D. (2015). At all costs: A comparison of robust cost functions for camera correspondence outliers. In *2015 Computer and Robot Vision (CRV)*, pages 62–69. IEEE. (ref. page 42)

Matthews, J. B. and Nesnas, I. A. (2012). On the design of the Axel and DuAxel rovers for extreme terrain exploration. In *Aerospace Conference, 2012 IEEE*, pages 1–10. IEEE. (ref. pages 1, 2, 10, and 106)

McGarey, P., MacTavish, K., Pomerleau, F., and Barfoot, T. D. (2016). The line leading the blind: Towards nonvisual localization and mapping for tethered mobile robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4799–4806. (ref. pages 6, 58, and 106)

McGarey, P., MacTavish, K., Pomerleau, F., and Barfoot, T. D. (2017a). TSLAM: Tethered simultaneous localization and mapping for mobile robots. *To appear in the International Journal of Robotics Research.* (ref. pages 6 and 58)

McGarey, P., Polzin, M., and Barfoot, T. D. (2017b). Falling in line: Visual route following on extreme terrain for a tethered mobile robot. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2017–2034. (ref. pages 6, 59, and 82)

McGarey, P., Pomerleau, F., and Barfoot, T. D. (2015). System design of a tethered robotic explorer (TReX) for 3D mapping of steep terrain and harsh environments. In *2015 International Conference on Field and Service Robotics (FSR)*. Springer. (ref. pages 6 and 22)

McGarey, P., Tang, T., Yoon, D., Pomerleau, F., and Barfoot, T. D. (2017c). Field deployment of the tethered robotic explorer to map extremely steep terrain. In *2017 International Conference on Field and Service Robotics (FSR)*. Springer. (ref. pages 6 and 100)

McManus, C., Furgale, P., Stenning, B., and Barfoot, T. D. (2013). Lighting-invariant visual teach and repeat using appearance-based lidar. *Journal of Field Robotics*, 30(2):254–287. (ref. page 61)

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598. (ref. pages 25 and 29)

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., and Kawatsuma, S. (2013). Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63. (ref. page 8)

Nohmi, M. (2004). Attitude control of a tethered space robot by link motion under microgravity. In *2004 IEEE International Conference on Control Applications*, volume 1, pages 424–429. (ref. page 61)

Osinski, G. R., Barfoot, T. D., Ghafoor, N., Izawa, M., Banerjee, N., Jasiobedzki, P., Tripp, J., Richards, R., Auclair, S., Sapers, H., Thomson, L., and Flemming, R. (2010). Lidar and the mobile Scene Modeler (mSM) as scientific tools for planetary exploration. *Planetary and Space Science*, 58(4):691–700. (ref. page 84)

Paton, M., MacTavish, K., Warren, M., and Barfoot, T. D. (2016). Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (ref. page 61)

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, 34(3):133–148. (ref. pages 85 and 87)

Rajan, V. A. K. T., Nagendran, A., Dehghani-Sanij, A., and Richardson, R. C. (2014). Tether monitoring for entanglement detection, disentanglement and localisation of autonomous robots. *Robotica*, pages 1–22. (ref. page 25)

Rupp, C. C. (1975). A tether tension control law for tethered subsatellites deployed along local vertical. *NASA STI/Recon Technical Report N*, 76:11194. (ref. page 61)

Scaramuzza, D., Fraundorfer, F., and Siegwart, R. (2009). Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4293–4299. (ref. page 38)

Schenker, P. S., Elfes, A., Hall, J. L., Huntsberger, T. L., Jones, J. A., Wilcox, B. H., and Zimmerman, W. F. (2003). Expanding venue and persistence of planetary mobile robotic exploration: new technology concepts for Mars and beyond. In *Photonics Technologies for Robotics, Automation, and Manufacturing*, pages 43–59. International Society for Optics and Photonics. (ref. page 1)

Sinden, F. (1990). The tethered robot problem. *The International Journal of Robotics Research*, 9(1):122–133. (ref. pages 2, 8, and 23)

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer. (ref. pages 25 and 85)

Stenning, B., Bajin, L., Robson, C., Peretroukhin, V., Osinski, G. R., and Barfoot, T. D. (2015). Towards autonomous mobile robots for the exploration of steep terrain. In *Field and Service Robotics*, pages 33–47. Springer. (ref. pages 10 and 15)

Sünderhauf, N. and Protzel, P. (2012). Switchable constraints for robust pose graph slam. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1879–1884. (ref. pages 38 and 42)

Teshnizi, R. H. and Shell, D. (2014). Computing cell-based decompositions dynamically for planning motions of tethered robots. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6130–6135. (ref. pages 23, 38, and 105)

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1):99–141. (ref. page 29)

Thrun, S. and Montemerlo, M. (2006). The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429. (ref. page 25)

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372. Springer. (ref. page 25)

Tsai, D., Nesnas, I. A., and Zarzhitsky, D. (2013). Autonomous vision-based tethered-assisted rover docking. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2834–2841. (ref. page 61)

Wettergreen, D., Thorpe, C., and Whittaker, R. (1993). Exploring Mount Erebus by walking robot. *Robotics and Autonomous Systems*, 11(3):171–185. (ref. pages 1, 2, and 9)

Yoerger, D. and Slotine, J. (1985). Robust trajectory control of underwater vehicles. *IEEE Journal of Oceanic Engineering*, 10(4):462–470. (ref. page 61)

Zhang, A. M. and Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments. *The International Journal of Robotics Research*, 28(3):331–356. (ref. page 61)

Zhang, F. (2006). *The Schur complement and its applications*, volume 4. Springer Science & Business Media. (ref. page 35)

Zhang, J. and Singh, S. (2014). LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems Conference (RSS)*, pages 109–111. (ref. page 85)

Zhang, J. and Singh, S. (2015). Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. (ref. page 85)