2022

System Development of an Unmanned Ground Vehicle and Implementation of an Autonomous Navigation Module in a Mine Environment

Jonas Amoama Bredu Jnr
West Virginia University, jobredu@mix.wvu.edu

System Development of an Unmanned Ground Vehicle and Implementation of an Autonomous Navigation Module in a Mine Environment

Jonas Amoama Bredu Jnr

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science in
Mechanical Engineering

Yu Gu, Ph.D., Chair
Jason N. Gross, Ph.D.
Guilherme Pereira, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2022

Keywords: Mine Safety, Robotic Systems, Unmanned Ground Vehicle

Abstract

System Development of an Unmanned Ground Vehicle and Implementation of an Autonomous Navigation Module in a Mine Environment

Jonas Amoama Bredu Jnr

There are numerous benefits to the insights gained from the exploration and exploitation of underground mines. There are also great risks and challenges involved, such as accidents that have claimed many lives. To avoid these accidents, inspections of the large mines were carried out by the miners, which is not always economically feasible and puts the safety of the inspectors at risk. Despite the progress in the development of robotic systems, autonomous navigation, localization and mapping algorithms, these environments remain particularly demanding for these systems. The successful implementation of the autonomous unmanned system will allow mine workers to autonomously determine the structural integrity of the roof and pillars through the generation of high-fidelity 3D maps. The generation of the maps will allow the miners to rapidly respond to any increasing hazards with proactive measures such as: sending workers to build/rebuild support structure to prevent accidents. The objective of this research is the development, implementation and testing of a robust unmanned ground vehicle (UGV) that will operate in mine environments for extended periods of time. To achieve this, a custom skid-steer four-wheeled UGV is designed to operate in these challenging underground mine environments. To autonomously navigate these environments, the UGV employs the use of a Light Detection and Ranging (LiDAR) and tactical grade inertial measurement unit (IMU) for the localization and mapping through a tightly-coupled LiDAR Inertial Odometry via Smoothing and Mapping framework (LIO-SAM). The autonomous navigation module was implemented based upon the Fast likelihood-based collision avoidance with an extension to human-guided navigation and a terrain traversability analysis framework. In order to successfully operate and generate high-fidelity 3D maps, the system was rigorously tested in different environments and terrain to verify its robustness. To assess the capabilities, several localization, mapping and autonomous navigation missions were carried out in a coal mine environment. These tests allowed for the verification and tuning of the system to be able to successfully autonomously navigate and generate high-fidelity maps.

Contents

List of Figures

List of Tables

Acknowledgments

Words cannot express my gratitude for the tremendous support and mentorship I have received throughout my college and research career at West Virginia University and I would like to take a brief moment to acknowledge a few of them.

First, I would like to thank **Dr. Yu Gu**, for being an amazing advisor, mentor and for giving me the opportunity to grow as a person and as a researcher. I would also like to thank **Dr. Jason Gross** for his support, instruction, and insightful advice. In addition, I would like to thank **Dr. Guilherme Pereira** for his invaluable feedback and guidance on my research.

Getting through my thesis and research required more than academic support, and I have many people to thank especially from the amazing family of the WVU Robotics Team. I would like to express my sincere gratitude and appreciation to **Dylan Covell**, **Christopher Arend Tatsch**, **Henry Voss**, **Tyler Wolf**, **Trevor Smith**, and **Gio Molin** for their invaluable contributions to the development of the unmanned ground rover.

For the design and development of the hardware system, **Dylan Covell**, **Tyler Wolf**, **Trevor Smith**, and **Gio Molin's** hard work, and expertise have helped me make significant progress in this exciting project. I am grateful to **Christopher Arend Tatsch**, **Henry Voss** and **Dr. Nicholas Ohi** for the time and effort you have put into this project, and I am grateful for the expertise and knowledge you have brought to the table for the development of the software modules. Your contributions have been crucial to the success of the project, and I am grateful for your continued support and collaboration. Thank you for your invaluable contributions to this important work.

Special thanks go to **Dr. Nicholas Ohi**, **Dr. Cagri Kilic**, **Dr. Chizhao Yang**, **Jared Beard** and **Bernardo Martinez Rocamora Jr.** for their extraordinary support in this study.

I would like to also acknowledge **The Lane Innovation Hub** for their contribution to the manufacturing of the Robot. Their expertise and assistance have been invaluable in ensuring the successful production of the Rhino. Thank you, **Kelsey Crawford** and **Jordan Friend**, for your hard work and dedication.

This thesis was partially sponsored by the **Alpha Foundation for the Improvement of Mine Safety and Health, Inc (ALPHA FOUNDATION)**. The views, opinions, and recommendations expressed herein are solely those of the authors and do not imply any endorsement by the **ALPHA FOUNDATION**, its Directors and staff.

1

Introduction

Humans have been exploring subterranean spaces such as caves and mines since our existence. The exploration of subterranean spaces has immense value across many applications and fields such as resource extraction and planetary exploration. Although there are tremendous benefits to the insights gained from exploring these spaces, there are also great risks and challenges involved. The exploitation of these spaces have led to accidents that have claimed many lives such as the Quecreek Coal Mine [137] in Pennsylvania, the Daxing Coal Mine [92] in China, and the Karl Marx Mine [113] in Ukraine.

Over the years, the National Institute for Occupational Safety and Health (NIOSH) has conducted research to improve the structural design and working conditions in active mines. The research being conducted is to prevent accidents, protect human lives, and improve and stabilize structures. The guidelines detail modern pillar designs for all mines, including limestone mines which are the focus of this thesis. Underground limestone mines generally have strong structures and are generally stable, and the enhanced pillar designs developed by NIOSH have improved the stability of these mines even more. Although the guidelines provided are being enforced, previously mined sections stay open for years and are uninspected. Over time, these sections undergo time-dependent degradation which is not covered by the guidelines. The evolution of the cracks and damages to the roof and pillars result from factors such as geology, excavation geometry, in-situ stress state, mining-induced stress, and/or change in mechanical properties of the rock with time [134]. Since the implementation of NIOSH's guidelines are more recent, some sections of the mines might have older pillar designs that may not be up to current safety specifications and standards. Due to the outdated safety factors from the old designs and time degradation, over time the pillars can be affected by sloughing as well as reported roof falls. This can pose a threat to the miners, as they have to travel through some mined sections to get to the working face. In a limestone mine in Whitney, Pennsylvania, there was a massive pillar collapse that injured miners due to the air blast but luckily did not cause any fatalities [5, 46]. After investigations were conducted, it was discovered that the area that had collapsed had not been mined for approximately 15 years.

From 1983 to 2021, the fall of ground has been the leading cause of fatalities in underground mines, accounting for 40.20% [8] and more specifically 60% [9] for stone mines, reported by the Centers for Disease Control and Prevention (CDC). Due to the severity of these occurrences, inspections are necessary to prevent any collapse, but current methods involve human inspections, which are impractical due to the vast size of some mines as well as the risk to life to conduct the

inspections.

By proactively inspecting the roofs and pillars, areas with elevated hazards can be repaired or avoided to ensure the safety of all miners. In efforts to improve the safety conditions, continual research has led to useful insights into understanding the structure, pillars, and roof of mines [28, 88–90, 126, 147]. These standards include improving roof support performance, maintaining safe tailgate escape ways from longwalls, optimizing pillar design for retreat mining, controlling multiple seam interactions, predicting roof conditions during extended cuts, and preventing massive pillar collapses [7]. But due to the vastness of stone mines, the inspections of entire mines are infeasible due to the risks involved, the limited number of workers, and the economic impact.

1.1 Problem Statement

The goal of this research effort is to develop a robust and reliable autonomous unmanned ground vehicle that can operate in stone mine-related subterranean environments to generate high-fidelity 3D maps that can be used to prevent accidents. Most importantly, the development of the robotic platform is to reduce the human factor from the tedious and dangerous task of inspections which can alleviate some of the risks faced by mine workers. Additionally, in the case of accidents, the generated 3D maps can be used by search and rescue teams to help mine workers. The lack of such systems motivates the objective of this research which is the following:

> The implementation of a robust unmanned ground vehicle with localization, mapping, and autonomous navigation capabilities.

1.2 Contributions

The contribution to the research project is the development, implementation, system integration, and field testing of a robust unmanned ground vehicle (UGV). The primary objective of the UGV is to autonomously traverse the challenging terrain to create high-fidelity maps of subterranean environments for extended periods. Another objective of the system is to provide power to an unmanned aerial vehicle (UAV) when implemented in a stone mine to extend the operational time as well as the mapping capabilities to the blind spots of the UGV. The UGV autonomously navigates to different locations and once stationary a UAV is deployed to scan the tall pillars. Since there are

no weight limitations, the UGV can carry various sensors and electronics that can be used to infer as much information about these environments, unlike the UAV. The research contributions are summarized below:

1. Development and implementation of a robust and reliable system to complete all required missions.

2. Autonomously localize and map the mine environment for extended periods.

3. Create high-fidelity 3D maps.

1.3 Broader Impact

The goal of the research is to develop and implement modern robotic systems to enhance monitoring and warning systems of old workings in underground stone mines. Although the fall of ground and pillar stability are necessary to be studied for all mines, these accidents account for a higher fatality rate in stone mines. Therefore the proposed project focuses on underground stone mine operations, with 60% [9] of fatalities and 11.5% [1] of injuries.

The successful implementation of the system will allow mine workers to autonomously determine the structural integrity of the roof and pillars. As the structural integrity of the roofs and pillars are determined, it will allow the miners to rapidly respond to any increasing hazards with proactive measures such as: sending workers to build/rebuild support structure to aid in the prevention of accidents, warning miners of highly hazardous areas to allow for the use of alternate routes or the evacuation of the mine. Insights gained from the high-fidelity 3D maps can also be used to accurately determine the volumetric change of pillars over time, which can then be used to update strength degradation in pillar models.

The rest of this thesis will be organized as follows. In Chapter 2, the related works are discussed to motivate the research and the gaps. Chapter 3 details the state-of-the-art algorithms implemented to complete the missions required. Chapter 4 discusses the considerations made for the development of the system and the proposed approach. Finally, Chapter 5 details and discusses the experiments conducted by the system.

2

Related Works

Localization, mapping, and operations in subterranean environments or GPS-denied environments such as caves and mines is a challenging task that requires knowledge and understanding of different research fields and topics. Over the years, advances have been made in sensors, electronics, and algorithms for research purposes that have led to innovative solutions and algorithms. These advancements have also led to an increased interest for commercial purposes in applications such as autonomous unmanned vehicles for warehouse applications, self-driving cars, search and rescue applications, etc [14, 33, 136]. Although there is no solution that can be applied in all scenarios, the continuous advancements have led to innovative solutions such as Simultaneous Localization and Mapping (SLAM), Visual and Inertial Navigation (VIN) systems, etc. The application of these solutions for autonomous vehicles must be robust, precise, and able to handle software and hardware failures in challenging environments and scenarios [136].

Although there are innovative developments and advancements in localization and mapping algorithms, these environments are still very challenging for unmanned systems to explore autonomously. The difficulties are due to the extreme, rugged, and unstructured terrain and environments which can have reduced visibility, moisture, varying temperatures, and water which can be difficult to traverse even when being teleoperated. In order for an autonomous system to successfully traverse or operate in these environments, the hardware system has to be robust, reliable, and adaptable to different terrain. The sections below give an overview of some of the techniques and algorithms developed for GPS-denied environments. The section also details some of the unique robots developed, implemented, and tested in subterranean environments.

2.1 Overview of Localization Methods

Localization of unmanned and manned vehicles for autonomous operations is an essential and fundamental area of research. Localization is the process of estimating an object's position and orientation with respect to its environment and is usually based on a map and reference point. The process of localization requires both hardware (sensors) and algorithms to run in parallel for an agent to determine its pose. The localization solution is vital for path planning, waypoint navigation, decision-making, and other tasks that are required by any mission. This section gives a brief overview of localization models and the sensors that are commonly used.

Generally, localization algorithms can be described by their sensor perception modules, which

are proprioceptive and exteroceptive sensors [142]. Proprioceptive sensors are responsible for observations of the state of the robot such as accelerometers, magnetometers, gyroscopes, etc. Alternatively, exteroceptive sensors are responsible for the observations of the environment such as cameras, LiDAR, sonar, etc. In order to localize an unmanned vehicle, researchers usually combine both perception modules in various algorithms.

Fig. 2.1.1 characterizes some of the commonly used localization sensors. Tactile sensors are devices that provide measurements by responding to physical contact with the environment [29, 91]. These types of sensors can be used in Autonomous Underwater Vehicles (AUV) instrumented with tactile whiskers to generate haptic maps. The generation of the haptic maps allows operators to gain insightful knowledge on features such as surface form, texture, and compliance when used in the demining and maintenance of oil rigs [109]. Tactile sensors are usually of low complexity, but the implementation cost depends on the operating area's size and is more suitable for applications with small operational areas. Since these sensors require physical interaction to produce measurements, a disadvantage is their susceptibility to wear and damage [91]. Due to most conventional tactile sensors having limited sensing range, researchers have found their application in localization techniques to be inefficient for 3D outdoor applications [29]. Rotary sensors measure the angular position and speed, which is used to determine the current position at time step k and the previous time step $k-1$ when used in wheeled robots, also known as dead reckoning. Dead reckoning is the process of estimating the pose of an object based on estimates of speed, heading, and time from previous estimates [123]. A disadvantage of these systems is their susceptibility to the accumulation of errors due to sensor biases and drift after extended operational times.

External referencing systems position a target based on its relative pose with respect to a known landmark. For indoor applications, external referencing systems require two types of hardware components, which are signal transmitters and measurement units. A disadvantage of these types of sensors in the case of Global Navigation Satellite Systems (GNSS) are multipath issues that lead to incorrect pose estimates, which can be detrimental in some applications [82]. Vision and Active ranging sensors observe and take measurements of the surroundings and then localize. A disadvantage of these sensors, when used in localization algorithms, is apparent in environments that have ambiguous and/or lack features that lead to incorrect loop closures. This can cause the robot to perceive the environment as an endless corridor [49, 72, 116]. Vision and active ranges can also be affected in the presence of obscurants in environments such as dust, fog, and smoke, leading to

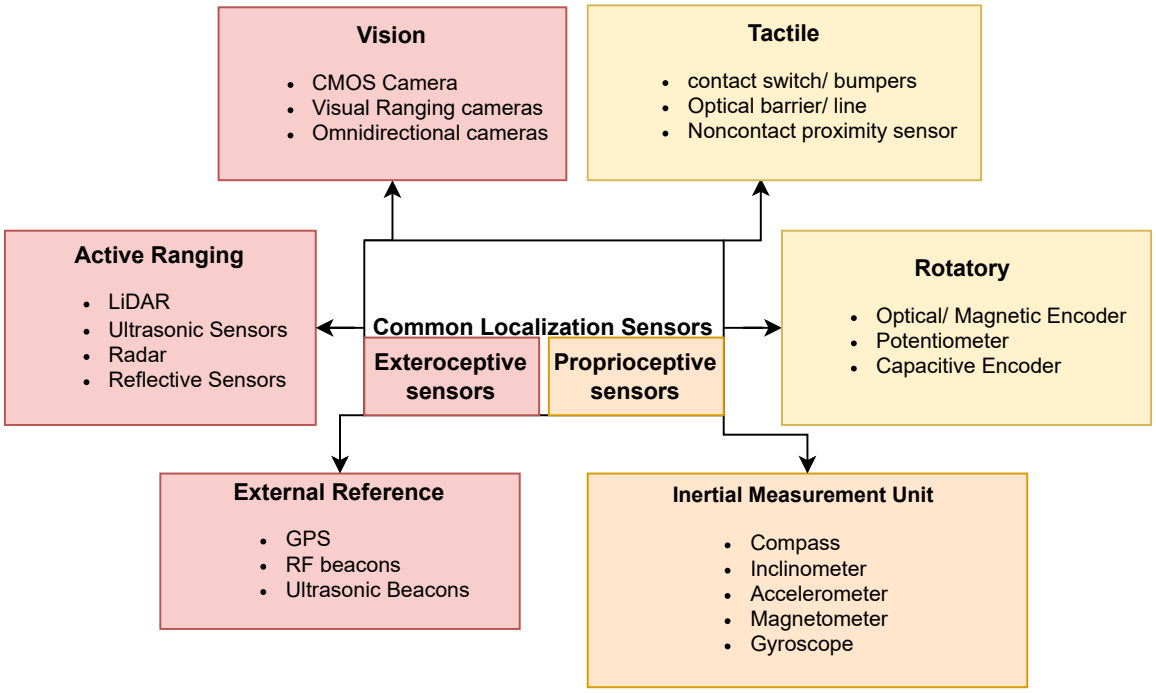


Figure 2.1.1: Example of Localization Sensors. Recreated by author based on [142]

incorrect measurements [111]. Although these sensors are affected by some particular conditions but not others, when used in a complementary manner in localization algorithms, they can yield robust algorithms [85, 120, 148].

For localizing agents in indoor applications, onboard localization systems are widely used when the environment is not modified with anchors. Onboard localization systems refer to the use of a single modality system to localize an unmanned vehicle by using either the incremental and/or relocalization models. The use of either is based on the availability of resources and the complexity of the application [142]. Incremental models refer to the gradual growth of the observation while simultaneously localizing the agent used when prior maps are unavailable. The relocalization model refers to agents that calculate the pose based on a known map and current observation [82]. This model is used primarily when a prior map is available, for example, in warehouse applications, while the incremental model is used when a map is unavailable or for exploration. The following section gives an overview of the techniques and algorithms that are widely used for unmanned vehicles.

Section 2.2 reviews LiDAR-based localization algorithms used by researchers to reliably localize unmanned vehicles in challenging environments without the use of external referencing such as GPS.

2.2 LiDAR Based Localization

LiDAR sensors are becoming crucial sensors in fully autonomous applications. Over the years, advancements in electronics have led to the development of even more precise devices and with high interest in autonomous applications, there has also been a decrease in the price and an increase in their availability [21, 45, 136]. In addition to the development and improvement of sensors, researchers have been developing algorithms to solve or improve state estimation techniques.

These algorithms use data from multiple sensors, such as point clouds from the LiDAR for pose estimation and velocity estimates (e.g., fusing sensor readings from IMUs and wheel encoders for localizing unmanned ground vehicles). Prior to SLAM and LiDAR-based techniques, researchers relied on odometry-based techniques which integrated wheel encoder readings. Due to sensor biases and noise, techniques that relied on wheel odometry would drift after a few meters which led to unwanted behaviors and poor state estimates [30]. However, there have also been developments in odometry algorithms that improve accuracy and robustness. Researchers have also been successfully improving visual and inertial-based algorithms that provide better state estimates [48, 81].

For autonomous exploration and navigation in underground environments, researchers are widely using LiDAR-based localization. Due to the versatility of LiDAR sensors, the ability to provide high-fidelity 3D measurements and work in various environments and conditions. Although there is research ongoing and progress made for this localization method, it is still a challenging problem. It requires in-depth knowledge of topics from different fields like signal processing, statistics, computer vision, and many more. Research on sonar- or vision-based localization techniques has led to the development and improvements of LiDAR-based localization methods and SLAM.

The inception of SLAM algorithms came about in 1986 when probabilistic methods were being introduced to improve or solve problems in robotics and artificial intelligence [21]. There had been ongoing research on the application of estimation and theoretic methods to solve problems in mapping and localization. SLAM algorithms aim to simultaneously estimate the pose of the unmanned vehicle while creating a model/map of the environment utilizing sensors onboard an unmanned

vehicle. The creation of a map supports other tasks such as path planning, obstacle avoidance, waypoint navigation and improves the state estimates. The map improves the state estimates when the robot revisits and recognizes known areas, known as loop closure. However, without loop closure, SLAM reduces to odometry.

The types of SLAM algorithms can be classified into two categories, which are filtering or smoothing [54]. Filtering approaches model the localization problem as an online estimation. In online estimation techniques, the states contain the current poses and map solution which are updated incrementally as new measurements become available [23, 52]. Alternatively, smoothing techniques solve the localization problem from a set of measurements and usually rely on least-square error minimization techniques [38, 93].

Over time as interest and progress were made in the development of algorithms in SLAM, this led to a formal characterization of SLAM. The structure of SLAM algorithms can be characterized by two main areas: front-end and back-end, Fig. 2.2.1 shows an overview of a SLAM architecture. The front-end is responsible for receiving data from sensors for feature extraction and data association, while the back-end uses information from the front-end for state estimation. An essential

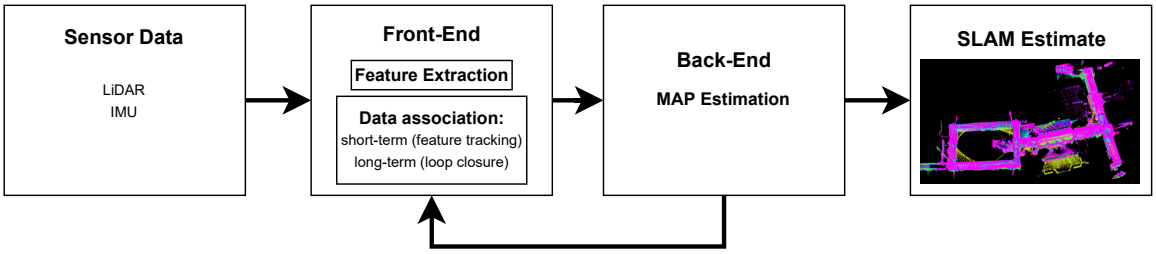


Figure 2.2.1: SLAM Overview. Adapted from [21]

aspect of state estimation is the representation of the sensor measurements as analytic functions required by maximum a posteriori (MAP) estimation, which is a challenging task. This makes the front-end a crucial step in SLAM formulations. Advances in both components will be further discussed in the following paragraphs.

The back-end of SLAM is responsible for state estimation and is usually formulated as a MAP estimation problem with the advantage of this formulation being that the observation and motion models are not explicitly defined, unlike the Kalman Filter approach. MAP estimation allows both

models to be treated as factors in the graph allowing them to be seamlessly integrated into the estimation process. Although in linear Gaussian applications, both the Kalman Filter and MAP estimation methods provide the same estimates, but most problems in robotic applications are nonlinear. MAP estimation computes the robot and landmark poses X^*, equivalent to the mode of the posterior distribution, following Bayes theorem. The cost function for estimating the robot and landmark poses (X^*) is:

$$\mathbf{X}^* = \arg\min_{x} \sum_{k=0}^{m} \|h_k(X_k) - z_k\|^2_{\Omega_k} \tag{2.1}$$

where h_k is a nonlinear function, X_k is the unknown variable, z_k represents the measurements and Ω_k represents the information matrix. MAP estimation, therefore, aims to minimize the negative log-posterior a nonlinear least squares problem which can be solved using the Gauss-Newton or Levenberg–Marquardt methods

Due to the interdependence of the variables required for state estimation, factor graphs are used [38, 47, 53, 69, 118, 119]. Factor graphs characterize the unknown pose estimates as variables and functions that represent constraints of the variables as factors [37]. The observation and motion models, robot and landmark poses, calibration parameters, and factors that bound these variables can be represented as factors in the graph. The use of factor graphs makes it easier for researchers to visualize the problem which can lead to improvements in computational complexity. The structure of factor graphs also allows for the use of fast linear solvers [53, 69], as well as libraries of solvers that are capable of processing thousands of data points in seconds [10, 37, 78, 102].

The front-end of SLAM is responsible for taking in sensor inputs for feature extraction, data association, loop closure, and validation as well as providing an initial guess for the variables in the nonlinear optimization step. Prior to the back-end, the front-end extracts the relevant information from the sensor measurements to be used in the minimization in Equation 2.1. The data association in the front-end is responsible for short-term data association which happens in concurrent sensor measurements. While, long-term data association is responsible for associating older measurements with newer measurements, which occurs when there are loop closures. For loop closures and validation, the back-end provides information to the front-end. An example of the implementation of the front-end in vision-based SLAM is the extraction of important pixel locations

from the environment. Once the pixel locations are extracted, the information can be modeled in back-end for state estimation. An alternative formulation designed to solve SLAM is pose graph optimization, which samples poses along the robot path and imposes constraints on pairs of poses [23, 25, 26].

There have also been improvements in EKF-based SLAM systems that have been able to achieve great performance [34, 58, 76, 95] compared to earlier formulations. In [95], the measurement model is capable of representing geometric constraints of static features observed from multiple viewpoints. The performance of EKF-Based formulations and MAP estimation are similar when there is accurate linearization and when sliding-window filters are implemented [61, 122].

In the design of any system and/or algorithms, there is a need for robustness. The robustness of an algorithm means that the system can function properly even when there is an algorithmic or hardware failure. Current formulations of SLAM can fail in the presence of challenging environments or highly dynamic environments. The need for robust systems becomes more essential when there are long-term operations. A well-known source of failure in SLAM is the data association step, which can either be in the short-term or long-term. The occurrence of short-term data association failure can be easily addressed as compared to long-term data association which is harder to deal with. An example of failures during data association that can occur in feature-based SLAM, is when different sensor inputs register as the same, also known as perceptual aliasing. This occurrence can lead to incorrect detection of outliers or false positives which are then fed into back-end thereby compromising the state estimates from MAP algorithm [127]. Issues that occur during data association can be addressed in both the front-end and the back-end. Problems in short-term data association can be easily addressed, one method is to select a sensor that has a higher sampling rate than the dynamics of the unmanned vehicle. This allows for the tracking of the features that correspond to the same 3-D landmark in concurrent time steps but can increase the computational complexity. To address the long-term data association issues different methods have been developed for both laser-based approaches [131] and visual-based approaches [31, 59, 86].

Loop closure validation is also another key aspect, especially for long-term data association, since it is responsible for determining the quality of loop closures by using previously collected sensor measurements and current sensor measurements. Loop closure validation is essential for robust SLAM algorithms since incorrect closures that are fed back to the back-end will lead to incorrect state estimates. Researchers have proposed several methods to validate loop closures, such

as [39], commonly used for outlier detection and geometric verification. The calculation of the residual by examining the current laser scan and the existing scan in laser-based approaches can also be used to check the validity of a closure. Another area of research to deal with the issues of long-term and short-term data association is by making the back-end more robust to incorrect measurements [24, 80, 99]. A proposed method by [127] deals with incorrect loop closures in the back-end by allowing the back-end to make changes to the graph during the optimization step. This allows the back-end to disregard incorrect closures and allow convergence even in the presence of incorrect loop closures either in Pose Graph Optimization or Factor Graphs. All the approaches described in the previous paragraphs involve prior knowledge and combination in many research fields, such as sensor fusion, pointcloud registration, outlier detection, and loop closure detection. With great interest and the development of numerous algorithms, the next sections will describe some of the algorithms and their variations that are applied in localization techniques.

2.2.1 Point Cloud Registration Methods

In order to successfully localize and map, the techniques being implemented require the sensor measurements to be registered to be used in the algorithms. In LiDAR-Based SLAM, the front-end is responsible for the interpretation of range data from the LiDAR which is then used in the back-end for state estimation [140] As sensors such as LiDARs generate range data while traversing an environment or pixel location from cameras, the data must be registered for state estimation. Registration algorithms allow the integration of data from multiple sources by associating sets of data into a common coordinate frame while minimizing the alignment error [103]. Registration algorithms are used in many applications and industries such as photography [128], 3D reconstruction and mapping, robotic exploration [60], organ reconstruction [132], and many more [16].

Point cloud registration methods are formulated to estimate the transformation (rotation and translation) that best aligns corresponding data points into the same reference frame. Transformations can be defined as cost functions that aim to minimize errors using optimization techniques. The Iterative Closest Point (ICP) Point-to-Point Algorithm [17] finds the unknown correspondence by using the nearest-neighbor approach [18] and iteratively neglects outliers to improve the previous translation and rotation estimates. However, the original ICP algorithm [17] has the limitation of often falling into local minima due to incorrect correspondences and sensitivity to outliers.

Due to the limitation, several variations of the ICP algorithm have been developed to improve the accuracy and robustness. The Generalized ICP (GICP) does not assume that the point clouds are collected from known geometric surfaces, but rather through noisy sensor measurements [115]. GICP performs plane-to-plane matching by using a probabilistic interpretation of the optimization problem [115]. However, the registration of point clouds directly can still be susceptible to noisy sensor inputs, which can influence the accuracy of the estimation.

An alternative approach to pointcloud registration is the Normal Distribution Transform (NDT) algorithm proposed by [19] for 2D applications. NDT converts the pointclouds to a normal distribution map by converting them into grid cells. A Probability Density Function (PDF) is used to calculate the likelihood of measuring a sample in each cell, as well as the covariance matrix and average. Space segmentation is then used to generate a Gaussian distribution function to determine the dispersion of pointclouds. To determine the transformation, a cost function is updated based on the gradient vector and Hessian matrix. For 3D application, [87] expanded the 2D NDT algorithm by [19] and improved the accuracy by improving the cost function by implementing a Gaussian approximation of the log-likelihood.

Although both NDT and ICP algorithms and their variations aim to solve the issue of point cloud registration, NDT algorithms tend to be more robust and precise. As compared to ICP, NDT does not need to establish a definitive correspondence between points or features to estimate transformations. This makes NDT algorithms more robust since establishing correspondence is the most error-prone step in the pointcloud registration process. The 3D NDT algorithm can also execute appearance-based loop closures and requires no pose information [87] which is essential in mapping applications. Nevertheless, both NDT and ICP can provide incorrect transformation estimates when there are a number of sizable dynamic objects in the environment. Therefore, in applications involving motion or dynamic objects, these algorithms cannot be implemented independently for registration.

2.2.2 Sensor Fusion

Sensor fusion is the integration of data and readings from multiple sources and sensors to produce accurate and consistent data. The goal of sensor fusion in state estimation for autonomous applications is to reduce the probability of any perceptible errors and increase reliability by using infor-

mation from multiple sources and sensors. Sensor fusion is essential because navigation based on sensors from a single source is insufficient. This has led to the development of some state-of-the-art state estimation techniques such as Kalman Filters [70], Particle Filters[51, 83], Covariance Intersection [67], Factor Graphs [37], etc.

Kalman Filtering is a sensor fusion algorithm widely recognized in signal processing, control systems, navigation, and control. Kalman filters are algorithms that receive data from noisy sensor data recursively to produce statistically optimal estimates of the system's state by computing the Jacobians [70]. If the system can be modeled as a linear system and the error can be modeled as Gaussian noise, the Kalman filter acquires the optimal estimates. Most engineering problems are non-linear dynamic systems, therefore the development of variations of Kalman filters are required for state estimation. A variation of the Kalman Filter that aims to reduce the errors of linearization is the Iterated Kalman Filter (IKF), which continuously linearizes non-linear systems at every timestep [15]. The Unscented Kalman Filter solves the nonlinearity of systems by predetermining the approximated state estimates through Gaussian Random variables by estimating the mean and covariances [66]. A variation of the Kalman Filter that is widely used in navigation systems and nonlinear signal processing is the Extended Kalman Filter (EKF), which linearizes a nonlinear system about the current state estimate [43]. Particle Filters, also known as sequential Monte Carlo methods, solve for the posterior distributions from noisy and partial measurements from sensors using particles for state estimation [51, 83].

Sensors that are used for navigation systems have varying rates and asynchronous can pose issues for the variations of Kalman Filters. Although these methods employ curve fitting and interpolation to align the data, the accuracy of the alignment cannot be guaranteed and can lead to undesirable performances. This led to the development of the Factor Graph Methods, which allows multi-sensor measurements with varying frequencies, delays, and noise distributions [38]. If a sensor becomes unavailable during operation due to signal loss or faults, the algorithm will not automatically add the associated data [57, 139]. Factor graph methods use bipartite graphs to represent multi-variable global functions composed of multiple local functions. Using bipartite graphs as a basis, variations of factor graphs have been developed for navigation [36, 38, 68]. Based on the sensors required for the navigation task, factor graphs generate the state, measurement, prior, and sensor models. A joint optimization algorithm is then designed for state estimation.

Along with the review of localization techniques, SLAM, and the different topics required to

successfully implement SLAM, the upcoming section reviews some of the unique SLAM solutions and implementations. As well as some novelty mechanical systems that have been employed for the exploration of subterranean and urban environments. The section also details an overview of some of the other methods employed for surveying and mapping subterranean environments.

2.3 Techniques used for Subterranean Exploration

The prior methods used to survey and map subterranean spaces are direct observation, borehole observation, and remote sensing. Direct observation methods involve placing humans in these spaces for first-hand observations and data collection, which can be dangerous and sometimes fatal. Borehole observation probes involve drilling boreholes into the spaces and deploying structures that carry a range of sensors like cameras, thermal sensors, gas sensors, etc. Remote sensing methods use non-intrusive, geophysical procedures that employ the use of electromagnetic waves, and soil composition to locate voids without drilling. Although remote sensing and borehole observations provided satisfactory results, direct observation methods provided the most sufficient information without the application of robotic platforms. Using robots in subterranean environments allows for the elimination of the human factor, safety concerns, and higher sensing capabilities. In the case of exploring unknown or used caves/mines, there is greater uncertainty about the conditions, which makes the use of unmanned vehicles an essential component. As mentioned in [94], all subterranean spaces are different, and the choice of approach for exploration or data collection depends on some key factors:

1. Limited ingress: either openings or boreholes
2. Constrained volume: the volume of the cave
3. Amount of water, gases, debris
4. Lack of illumination
5. Communication

In earlier developments of instruments where spaces are inaccessible and boreholes can be drilled, researchers developed borehole-deployable lasers (BDL) [6] and borehole-deployable sonar (BDS).

The design of BDLs generally consists of laser range measurement sensors, actuators that adjust the laser range measurement sensors, inertial sensors for position information, and cameras that help with the deployment of the BDLs. BDLs and BDS are functionally the same with the only difference being the use of sonar instead of lasers for range measurements. These devices are usually tethered and remote-controlled by a human operator who monitors the scans. Because the device is remotely operated by a human operator, it allows for in-field analysis. Data acquired from the BDLs can be used to construct a 3D point-cloud representation of the environment surveyed. Although BDLs and BDS' can be useful, they are limited and rely heavily on the precise drilling of boreholes which is also a challenging task.

Although there are many benefits to the exploitation and exploration of subterranean spaces, initially there was not much interest by the robotics community [94]. One of the first robotic platforms to perform mapping missions in coal mines was the Terragator mobile robot. The mobile robot was a semi-autonomous six-wheeled multipurpose robot equipped with sonar, and laser scanners, for pose estimation and obstacle avoidance [27]. Along with the development of the Terragator, there were other efforts to adapt commercial all-terrain vehicles and bomb disposal robots for autonomous navigation of subterranean spaces. Some of the first platforms developed were Wolverine V2, Gemini Scout, Numbat, CSIRO Numbat, Cave Crawler, and Groundhog [97, 106, 107, 130]. The designs of the Groundhog and Cave Crawler were geared toward the exploration of abandoned coal mines. Groundhog had a structure comprised of an all-terrain vehicle, ground clearance of 16cm, the ability to climb slopes of 30°, and a 1600lb payload of electronics. To enable autonomous navigation, the robots had lasers, gyroscopes, encoders, tilt sensors for mapping and localization, gas sensors for the detection of methane gas, and flotation switches to detect sinking in water or mud. For the navigation and mapping of coal mines, Groundhog used incremental scan matching for short-term pose estimation and topological SLAM for global localization [124].

Despite the development of these robots being a step in the right direction for autonomous navigation and exploration, these systems had limited applications due to both mechanical and software issues. Some of the issues were weight (hard to transport), communication limitations (use of tethered systems), and the wireless communication systems used were unreliable for non-line-of-sight operations. In the case of autonomous exploration, the SLAM system used did not have loop closure, which meant that Groundhog perceived the environment as an endless corridor and would continue exploring previously visited areas. Recent developments in robotic systems can be used

to improve survey methods, search and rescue [84], and planetary exploration [114].

In addition to the interests and benefits of exploration and exploitation of these environments, there is an increase in interest in localization and mapping in GPS-denied environments. Despite the fact there has been a lot of research into developing LiDAR-based solutions for GPS-denied environments and infrastructure that lack prior maps, these environments pose an added challenge due to obscurants, lack of illumination, debris, lack of perceptual features, and extreme terrain. Along with the Robotics Institute of Carnegie Mellon University's developments of Terragator, Groundhog, and Cave Crawler, the heightened interests have led to the evolution of many innovative solutions and improvements in underground SLAM. The heightened interest can partly be attributed to the introduction of robotic competitions held all over the world. Competitions such as [35, 63, 77, 96, 98, 100] have led to innovative solutions and advances in robotics.

The Defense Advanced Research Projects Agency (DARPA) with the Subterranean (SubT) Challenge has been a major catalyst for the development of state-of-the-art solutions. DARPA SubT has led to innovation for mapping, navigation, cooperation of multiple robotic platforms for faster operation, and exploration of these complex underground environments. The use of multiple robots allows for the simultaneous mapping of larger areas, while the inter-agent interactions allow for the minimization of mapping and localization errors through loop closures. The competition aims to improve search and rescue efforts in these environments, where time and knowledge of environmental hazards are critical to finding survivors. The implementation of robotic systems allows for the rapid creation of maps, and the localization of artifacts can greatly help survivors and rescue teams. The key technical areas of the DARPA SubT challenge are autonomy, perception, networking, and mobility [2]. The competition enables participants to develop platforms that have the capability of operating in varying and degraded environments such as areas with dust, fog, mist, water, smoke, and low-light The challenge of SubT has also led to advances in robust communication systems in complex environments plagued by RF propagation, limited line of sight, and diverse geology. Some of the systems and algorithms showcased in the tunnel and cave systems competition at DARPA SubT will be reviewed in the following subsections.

2.3.1 Innovative Algorithms Showcased at DARPA SubT competition

The competition exhibited novel approaches to localization and mapping, especially multi-robot SLAM [13, 41, 105, 133]. The use of multiple robots enhanced mapping and localization solutions as well as the coverage of larger areas in a shorter period of time, the implementation of multi-robot SLAM was necessary for some teams [42].

Multirobot SLAM enables the data collected by multiple unmanned vehicles to simultaneously build a consistent map of larger areas and improve localization estimates. Multirobot SLAM algorithms can be classified as centralized, decentralized, or distributed [108]. Centralized algorithms [40, 41, 133] depend on a base station to calculate pose, map estimates, and loop closure detection from the data collected by the multi-robot system for all robots. While in decentralized systems [13, 105], each robot is responsible for the calculation of pose, map estimates, and loop closure using data collected individually and from other robots. Alternatively, in distributed multi-robot SLAM there is a partial exchange of data with neighboring robots for individual pose and map estimation [108].

Although significant progress has been made in LiDAR, Visual-Inertial Odometry (VIO), and Thermal-Inertial Odometry (TIO)-based localization and mapping techniques, they can still be negatively impacted by some environmental factors. In the presence of obscurants such as dust, fog, and smoke LiDAR and visual camera sensor measurements can be impacted compromising pose estimates [111]. Thermal cameras in thermal inertial odometry can be also negatively impacted by environments with minimal temperature gradients. Even though each sensor (LiDAR, visual, and thermal) is affected by some particular adverse condition, but not by the other, when used in a complementary manner in localization algorithms, yields robust algorithms [85, 120, 148].

To complete the missions in the allotted time required by the competition, CERBERUS [133] designed a robust SLAM architecture known as CompSLAM. CompSLAM is a loosely-coupled sensor fusion algorithm that uses visual and thermal sensor measurements to complement LiDAR data for robust pose and map estimation in perception-degraded environments [73]. Although CompSLAM provided a unique solution for localization and mapping, CERBERUS' implementation did not have loop closures on the individual robotic platforms. Loop closures were completed using a Multi-Robot Mapping and Optimization (M3RM), a centralized mapping server and node running on the base station, and deployed robots, respectively [133]. The M3RM server at the

base station is responsible for tracking the local submaps for each robot, as well as integrating them into a global map and transmitting them back to the robots. Since loop closures occur at the base station, when robots are not within communication range, the errors in CompSLAM may become unbounded and may break the global map if an operator does not intervene.

Due to the deployment of multiple robots, for quick and ease of integration on different agents, some teams developed modular perception systems that contained core sensors required for localization and mapping. The implementation of the modular perception system allowed for the development of an innovative SLAM formulation used in the competition such as Team CSIRO's [62] Wildcat SLAM, a LiDAR-Inertial SLAM [105]. Team CSIRO [62], developed a modular perception system known as the CatPack, which contained a spinning LiDAR mounted at a 45° angle, multiple RGB cameras, and an IMU. The configuration of the CatPack enhanced the robot's perception of the floor and roof of narrow tunnels and allowed the use of planar surface elements (surfels) which is a key feature for their localization algorithm. The use of the CatPack [62] in Wildcat SLAM allowed the use of surfels as dense features for estimating the path of the unmanned vehicle. WildCat SLAM employs the use of a sliding window filter to estimate the odometry and local map of the system by integrating the LiDAR and inertial measurements by using continuous-time representations of the trajectory. A unique feature of Wildcat SLAM is the use of surfels as dense features for trajectory estimation. Although most sensor fusion algorithms require extensive calibration for different environments, the implementation of Wildcat SLAM and CatPack once calibrated use the same parameters across multiple environments even when switched between robots[42].

Team Explorer developed a robust multimodal sensor fusion algorithm that receives inputs from LiDAR, thermal, and visual data by using a probabilistic factor graph known as Super Odometry (SO) [146]. The SO framework combines the advantages of tightly- and loosely-coupled estimators. SO is highly dependent on IMU sensor measurement due to its robustness to environmental factors, unlike LiDAR, visual, and thermal sensors. To mitigate errors caused by drifting and biases, SO uses VIO and LIO to provide pose priors to constrain IMU biases.

The competition showcased some of the recent progress in localization and mapping algorithms for subterranean spaces and environments that lack prior maps. Although there were similarities in the mechanical and sensor systems used, there were some interesting systems that were used in the competition. With interests in robotics growing there are emerging companies that sell com-

mercial robotics that can be adapted to a variety of applications. Some teams decided to design and manufacture their robots while others opted to adapt commercially available products to their algorithms. The subsection below showcases some of the interesting approaches such as the use of legged robots and the commonly used roving robots with a variety of drivetrain systems.

2.3.2 Novel Platforms Showcased at DARPA SubT competition

The competition showcased the collaboration of a variety of systems, some of which were adaptations of commercial products, as well as custom platforms for unmanned ground and aerial vehicles. There was a variety of legged (walking and wheeled), hybrid (ground/aerial), tracked, and aerial robots. Aerial vehicles were used to rapidly explore areas and spaces where unmanned ground vehicles had difficulty accessing [11, 62, 110, 133]. Most teams deployed different types of unmanned ground vehicles to traverse the challenging terrain of both the tunnel and urban circuits. Since the tunnel circuit is the most relevant for this body of work, the innovations presented in the competition will be discussed.

Numerous teams chose to use legged robots due to their ability to adapt to the challenging terrain, their ability to step over obstacles on their path [65], the modification of their stance to crawl under obstacles [20], and their omnidirectional locomotion to traverse narrow spaces. For ease of operation and implementation in different environments, [133] adapted their quadrupedal robot to have flat feet for urban environments and actuated wheels for tunnel environments. Teams that did not use quadrupedal robots employed either tracked or roving robots as their primary system. Alternatively, some teams that used quadrupedal robots used roving or tracked robots as beacons to improve their communication systems.

Even though there are several algorithms and techniques for localization and mapping and unique robotic systems used for operations in subterranean environments such as mines, the systems showcased are for the development of search and rescue applications. The researchers realized most of the applications are reactive instead of proactive which will further benefit operations in these environments and save human lives. The lack of such proactive systems led to the motivation of this research project which is the development, implementation, and testing of a robust and autonomous hardware and software system that can operate in subterranean environments for extended periods of time. The successful implementation of the proposed system will not only allow for the genera-

tion of high-fidelity maps which will allow operators in these environments to proactively address issues in areas with increasing hazards but will serve as an early warning system for miners. The system will also allow for the further development of techniques and insights into furthering the understanding of these environments.

3

Fundamentals

In order to successfully complete all the missions required, for unmanned vehicles the mechanical and software system needs to be robust. For unmanned vehicles designed to assist in localization and mapping, an efficient sensor fusion module needs to be implemented. Prior to implementing a state-of-the-art SLAM algorithm, the sensors used must be calibrated to obtain accurate measurements for sensor fusion. The localization and mapping module built on a tightly-coupled LiDAR-inertial-odometry via smoothing and mapping (LIO-SAM) SLAM framework [118] formulated as a Maximum a Prior (MAP) problem has demonstrated high accuracy and real-time performance. To successfully implement LIO-SAM, the algorithm requires roll, pitch, and orientation estimates inferred from sensor measurements from IMUs. Roll and pitch estimates are required to initialize the unmanned system at the correct attitude for accurate localization. Although most MicroElectroMechanical Systems (MEM)'s IMUs instrumented in unmanned vehicles are six-degree-of-freedom (accelerometer and gyroscope) inertial measurement units, the implementation of a complementary filter can provide attitude estimates. The complementary filter outputs orientation estimates in quaternion form from inertial observations and optionally magnetic measurements when available [135]. In order to support autonomous missions required for mapping, there must be a robust navigation system that contains a terrain analysis, obstacle avoidance, waypoint, and path planning [22].

Fig. 3.0.1 shows an example of a navigation module needed to complete the missions required. The subsections below will detail an overview of some of the state-of-the-art solutions implemented to complete the mission requirements.

3.1 IMU Complementary Filter Overview

Advancements in electronics have led to MEMs technology which has been used to develop modern miniaturized inertial sensors. The use of MEMS systems has been employed in navigation systems for high-precision inertial and magnetic sensors. Most inertial sensors provide measurements from three orthogonal rate gyroscopes (gyro) and three orthogonal accelerometers. The readings from the accelerometer and gyroscope provide linear acceleration and angular velocity measurements, respectively. Orientation estimates are then obtained by processing and integrating gyroscope readings. For orientation estimates, the implementation of a complementary filter

or Extended Kalman Filter (EKF) [112] that provides quaternion estimates from inertial and/or magnetic measurements [135] is essential. Although both complementary filters and EKFs are widely used to acquire orientation estimates, complementary filters are used due to their simplicity and are easier to understand [64]. EKFs are also complicated to implement robustly and can be computationally expensive in comparison to complementary filters [141].

A complementary filter analyzes signals in the frequency domain to combine signals to infer estimations that are least affected by noise. The orientation estimates are represented as quaternions to avoid the singularity state, which is affected by the orientation when represented as Euler angles and Direction Cosine Matrices (DCM). The complementary filter [135], fuses attitude estimates in the quaternion form from gyroscope readings with accelerometer readings in delta quaternion form that serves as a correction for roll and pitch estimates only and maintains yaw estimates.

To attain these estimates, the angular velocity vector is used to predict an initial estimate of the orientation. Prior to using the angular velocity measurements in the prediction step, a high-pass filter is applied, since these readings are affected by low-frequency noise. To avoid filtering useful information, the filter is only applied when the system is in a steady-state condition. If the system is in a steady state condition, the bias is updated by averaging the readings over a period of time and subtracted from the gyroscope measurements, if not the previous bias estimate is maintained. Equations Eq. 3.1 [135] represents the orientation of the global frame relative to the local frame at time t_k, which can be computed by the integration of the quaternion derivative by using a sampling time of the difference between the current and previous timestep.

$$ {}^{L}_{G}\mathbf{q}_{\omega,t_k} = {}^{L}_{G}\mathbf{q}_{t_{k-1}} + {}^{L}_{G}\mathbf{q}_{\omega,t_k}\Delta t \tag{3.1} $$

The parameter ${}^{L}_{G}\mathbf{q}_{w,t_k}$ represents the orientation at the current time step, ${}^{L}_{G}\mathbf{q}_{t_{k-1}}$ represents the orientation at the previous timestep, ${}^{L}_{G}\mathbf{q}_{w,t_k}\Delta t$ represents the derivative of the angular velocity (ω: angular velocity) with respect to time.

The correction of roll and pitch estimates are computed using a multiplicative technique of the delta quaternion obtained from accelerometer measurements as shown in Eq. 3.2 [135].

$$ {}^{L}_{G}\mathbf{q} = {}^{L}_{G}\mathbf{q}_{\omega} \otimes \widehat{\Delta\mathbf{q}_{acc}} \tag{3.2} $$

where $\widehat{\Delta\mathbf{q}_{acc}}$ represents the correction of the roll and pitch estimates from accelerometer measure-

ments.

First, the "predicted gravity" which will have a small variation from the real gravity vector is obtained by finding the inverse predicted quaternion from the result of Eq. 3.1. In order to account for the small variations, $\Delta\mathbf{q}_{acc}$ is found due to the orientation estimates being represented in the global frame relative to the local frame:

$$\Delta\mathbf{q}_{acc} = \begin{bmatrix} \sqrt{\frac{g_z+1}{2}} & -\frac{g_y}{\sqrt{2(g_z+1)}} & \frac{g_x}{\sqrt{2(g_z+1)}} & 0 \end{bmatrix}^T \tag{3.3}$$

Since the delta quaternion estimates from accelerometer measurements are affected by high-frequency noise, before the multiplicative procedure in Eq 3.2 it is scaled down. The delta quaternion estimates are scaled down using an interpolation with the identity matrix $\mathbf{q}_I$:

$$\mathbf{q}_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{3.4}$$

With accelerometers being affected by high-frequency noise, a low-pass filter is implemented before the measurements are used. The delta quaternion is scaled down by using Ω, the angle between the identity quaternion $\mathbf{q}_I$ and $\Delta\mathbf{q}_{acc}$ derived from the spherical linear interpolation (slerp) formulation [121].

$$\widehat{\Delta\mathbf{q}_{acc}} = \frac{\sin(1-\alpha)\Omega}{sin\Omega}\mathbf{q}_I + \frac{sin(\alpha\Omega)}{sin\Omega}\Delta\mathbf{q}_{acc} \tag{3.5}$$

where $\alpha \in [0, 1]$ represents the gain for the cut-off frequency of the filter.

Although complementary filter produces accurate estimates, a typical drawback of these filters occur when there is highly dynamic motion [135]. During highly dynamic motion, the accelerometer is unable to provide accurate estimates of the gravitational direction [141], due to the accelerometer not only detecting gravitational forces but also centrifugal forces leading to inaccurate orientation estimates. Therefore, to mitigate orientation errors that occur when an unmanned vehicle experiences highly dynamic motions due to the constant gain designed for static conditions, the filter employs an adaptive gain component. These errors occur when the magnitude and direction of the total measured acceleration vector are different as a result of high acceleration. The high acceleration affects the magnitude and direction of the total measured acceleration vector by being

different from gravity, which leads to a false reference in estimating the orientation resulting in significant results. To address these issues, the gain factor must account for the error e_m [135] when the system is in such state.

$$e_m = \frac{|\|{}^{L}\tilde{\mathbf{a}}\| - g|}{g} \tag{3.6}$$

In Eq. 3.6, $\|{}^{L}\tilde{\mathbf{a}}\|$ is the norm of the acceleration in the local frame before normalization and $g = 9.81\frac{m}{s^2}$ acceleration due to gravity. With the magnitude of the error known, it can be accounted for in the computation of the filtering gain as [135]:

$$\alpha = \alpha(e_m) = \overline{\alpha} f(e_m) \tag{3.7}$$

where $\overline{\alpha}$ is the constant gain that attains the best filtering results in static conditions and $f(e_m)$ represents the gain factor of a piecewise continuous function. The piecewise continuous function, $f(e_m)$ that provided the best results was empirically [135] found:

Therefore, the gain factor becomes constant and equal to 1 when the magnitude of the nongravitational acceleration is below the acceleration due to gravity. In the case when there is a high acceleration, the nongravitational acceleration rises above a threshold of 1, the function $f(e_m)$ decreases the gain linearly. The gain is decreased linearly with the increase of the magnitude of the error until zero. Once the roll, pitch, yaw, orientation estimates, and LiDAR measurements are available, they can be used as inputs for localization and mapping algorithm.

3.2 LIO-SAM Overview

To localize and map underground mining environments, a state-of-the-art and robust SLAM algorithm is required. In determining the localization and mapping modules that would best suit the application, several algorithms were evaluated based on accuracy, sensors required and ease of implementation [117, 118, 143]. After conducting trade studies, along with insights gained from Section 2.2 and prior experiences, the LiDAR-inertial-odometry (LIO) algorithms that were evaluated were either tightly- or loosely-coupled fusion methods.

In loosely-coupled LIO algorithms, such as [117, 143], the state estimates from the IMU are

used as an initial guess for the LiDAR scan alignment but not for global optimization. Although these methods are computationally efficient, they are less accurate since the estimates from the IMU are ignored during global estimation. Alternatively, tightly-coupled methods such as [118], offer improved accuracy and are more robust since estimates from LiDAR and IMU are utilized during the global optimization through the implementation of factor graphs.

The localization and mapping module built on a tightly-coupled LiDAR-inertial-odometry via smoothing and mapping (LIO-SAM) SLAM framework [118] formulated as a Maximum a Posteriori (MAP) problem has achieved highly accurate and real-time performance. LIO-SAM is built on a factor graph formulation that allows multisensor input such as IMU, LiDAR, GPS, odometry, etc., and allows for global optimization. To mitigate issues with pointcloud deskewing due to high acceleration, a nonlinear motion model that receives raw IMU measurements is used to estimate the motion of the vehicle during the LiDAR scan. The estimated motion from the IMU is also used as an initial guess in the LiDAR odometry optimization and the approximation of IMU biases in the factor graph. For real-time performance and to improve computational complexity, the framework utilizes scan matching at the local scale, as well as the use of selective keyframes. The keyframes are efficiently introduced via a sliding window filter to a fixed-size set of prior sub-keyframes. With the formulation being a MAP problem, a Gaussian noise model can be assumed and solved as a linear least-squares problem. Nodes represent the robot state, while the factors in the graph are IMU preintegration, LiDAR odometry, GPS, and loop closure factors. The factor graph has the capability to be expanded with the addition of different sensors. Fig. 3.2.1 shows an example of a factor graph with inputs from LiDAR and IMU, adapted from [118]. New nodes are added to the graph by selecting keyframes, which are based on a user-defined threshold of robot poses between x_i and x_{i+1}.

To optimize pose estimation with the introduction of new nodes, an incremental smoothing and mapping using the Bayes tree (iSAM2) [69] solver is implemented in LIO-SAM [118]. The use of iSAM2 improves computational efficiency by employing factor graphs, maintaining sparsity, identifying and updating key variables with the introduction of new measurements. With the introduction of new measurements, iSAM2 using Bayes tree data structure, selectively solves for some variables in the factor graph instead of all the variables. The use of iSAM2 allows real-time performance and optimization. Inertial measurement units publish at high rates with average pub-

lish rates of 100Hz to 1kHz, efficiently using data at such high rates is not feasible for optimization. The IMU preintegration is responsible for integrating these measurements into single relative motion constraints to estimate the relative motion of the vehicle between two time steps [48]. Preintegration also allows the use of iSAM2, which strikes a balance between efficiency and accuracy and the avoidance of linearization errors.

For the LiDAR odometry factor, feature extraction is performed in LIO-SAM [118] with the arrival of new scans. Features are extracted by analyzing the roughness of the points in a local region. The types of features are classified by their roughness, with features with a large roughness being edge features and smaller roughness being classified as planar features. The use of keyframes allows the maintenance of a sparse factor graph, making it suitable for real-time nonlinear optimization. The sliding-window filter is used in the creation of the pointcloud map, which includes a fixed number of LiDAR scans. For efficiency, a transformation is performed between the most recent n keyframes (sub-keyframes) for estimation. After the selection of the sub-keyframes, there is a translation from the LiDAR frame to the world frame which are then merged together to form a voxel map. Since there is an extraction of planar and edge features, each feature is downsampled for the removal of duplicates in the same voxel cell. For robust scan matching, a method based on [144] is used. The predicted motion of the agent estimated from the IMU is used for the initial transformation from the body to the world frame. The correspondences of both edge and planar features are then found in the map frame. To compute the distance correspondences between edge and planar features, equations Eq. 3.9 and Eq. 3.10 are used respectively. The indices e and p represent edge and planar features respectively.

$$\mathbf{d}_{e_k} = \frac{\left|\left(\mathbf{P}^e_{i+1,k} - \mathbf{P}^e_{i,u}\right) * \left(\mathbf{P}^e_{i+1,k} - \mathbf{P}^e_{i,v}\right)\right|}{\left|\mathbf{P}^e_{i,u} - \mathbf{P}^e_{i,v}\right|} \tag{3.9}$$

$$\mathbf{d}_{p_k} = \frac{\left|\left(\mathbf{P}^p_{i,u} - \mathbf{P}^p_{i,v}\right) * \left(\mathbf{P}^p_{i,u} - \mathbf{P}^p_{i,w}\right)\right|}{\left|\left(\mathbf{P}^p_{i,u} - \mathbf{P}^p_{i,v}\right) * \left(\mathbf{P}^p_{i,u} - \mathbf{P}^p_{i,w}\right)\right|} \tag{3.10}$$

Indices k, u, v, w in Eq. 3.9 and Eq. 3.10 represent a feature and the corresponding features. For example, for an edge feature, $P^e_{i+1,k}$, $P^e_{i,u}$, and $P^e_{i,v}$ are the points that correspond to that edge on the map. For optimal transformations, the Gauss-Newton method is used to minimize the cost function as shown in Eq. 3.11

$$\min_{T_{i+1}} \left\{ \sum_{P^e_{i+1,k}} \mathbf{d}_{e_k} + \sum_{P^p_{i+1,k}} \mathbf{d}_{p_k} \right\} \tag{3.11}$$

With the optimal transformation evaluated, the LiDAR odometry factor between timesteps can be computed using Eq. 3.12.

$$\Delta \mathbf{T}_{i,i+1} = \mathbf{T}_i^T \mathbf{T}_{i+1} \tag{3.12}$$

When a robot visits a previously known area, loop closures are determined in LIO-SAM [118] using a Euclidean distance-based approach. By visiting a known location, the robot can improve its state estimates When a new node x_{i+1} is added to the graph, prior states are searched to find nodes that are close to the Euclidean space. Loop closures are detected based on a predetermined distance. Once the close nodes are determined, scan matching is applied to the LiDAR frames and added as a loop closure factor on the graph.

3.3 Autonomous Navigation Module Overview

In order to complete the missions autonomously, after the implementation of a localization and mapping system the unmanned system requires an autonomous navigation module. The UGV will have to autonomously navigate through different environments and terrains, to operate in uncertain and unknown terrain, thus the surface assessment is an essential component for successful missions.

Therefore, for successful autonomous missions, the navigation module must contain some essential components such as terrain traversability analysis and a waypoint planner. The waypoint planner should include a path planner and an obstacle avoidance module for both static and dynamic obstacles since the UGV might operate in environments with humans and other vehicles. The Fast likelihood-based collision avoidance (FALCO) algorithm [145] plans paths by generating offline trajectories according to the constraints and parameters of the unmanned vehicle and formulates the paths as a likelihood problem.

In order to complete the autonomous missions successfully, the terrain analysis module employed uses LiDAR data to determine the traversability of the local terrain around the vehicle [22].

As the vehicle traverses the terrain, the module builds a cost map and determines the traversability of each point in the map by assigning a traversal cost. The traversal cost is determined by representing the environment as a voxel grid which is then used to determine the ground height by analyzing the distributions of the LiDAR points in adjacent voxels. Once the distributions of the voxels are determined, the smoothness of the terrain is determined. The points that are determined to be further apart from the ground are assigned a higher traversal cost.

The terrain traversability works in parallel with the FALCO algorithm to plan a collision-free path to waypoints by dividing the likelihood problem into two subtasks. The first subtask is determining a global path plan that also ensures the paths do not fall into a local minima while the second subtask runs in parallel with the global path planner to track paths as well as avoids obstacles. In order to reduce the computational complexity, [145] avoids online searches of the graph that is continuously updated by onboard sensors, while the planning problem is formulated as a likelihood problem. The formulation of the algorithm does not seek to find the lowest cost (shortest path) but maximizes the probability of reaching waypoints by modeling the configuration space as two separate regions. Although most path planning algorithms seek to find a single path with the lowest cost, the paths might lead the vehicle to narrower pathways. In the presence of dynamic obstacles and narrower pathways, the probabilities of avoiding these obstacles become lower. The FALCO algorithm, however, prefers paths with open areas in order to account for obstacles that are not within the sensor field of view. Although this behavior leads to higher probabilities of completing the navigation task, the generated paths can be longer.

Equation 3.13, represents the probability of reaching waypoint B, p_B given the state x_s of the vehicle [145]. As sensor measurements become available, obstacles that are within sensor range are considered to be known deterministically and probabilistically known beyond sensor range with the availability of a prior map. During autonomous navigation, the vehicle travels through areas covered by the LiDAR which can be represented by x_f. The conditional distribution of x_f given a state, $p(x_f|x_s)$ can be inferred from the LiDAR data.

$$p_B(x_s) \approx \frac{1}{n} \sum_{i=1}^{n} p_B(\xi_i) \tag{3.13}$$

where $p_{B(x_s)}$ represents the probability for the unmanned vehicle to successfully reach B from a given start state x_s.

To distribute and group the trajectories across all probabilities, a trajectory library is used which is then evaluated during navigation to determine paths. Given a start state x_s, the algorithm generates different paths that can reach the sensor frontier as cubic spline curves based on the vehicle motion constraints. During operation, the paths that have occluded regions are considered to have obstacles detected by the perception sensors are defined by a Boolean function. The path clearance defined by the Boolean function ξ_i:

$$c(\xi_i) = \begin{cases} 1, & \xi_i = \textit{unocludded} \\ 0, & \textit{otherwise}. \end{cases} \tag{3.14}$$

With the path clearance function defined, Eq. 3.13 can be modified and applied to all the generated paths to determine the path with the highest probability of reaching B:

$$p_B(x_s) \approx \frac{\sum_{i=1}^{n} c(\xi_i) p_B(\xi_i)}{\sum_{i=1}^{n} c(\xi_i).} \tag{3.15}$$

In order to successfully complete all the missions required, the implementation of the algorithms described in this chapter is essential for the UGV. With the localization, mapping, and autonomous navigation module discussed, the design of the UGV will be reviewed in the next chapter. The next chapter will discuss the requirements of the missions, design considerations, and implementation of the algorithms discussed in the paragraphs above.

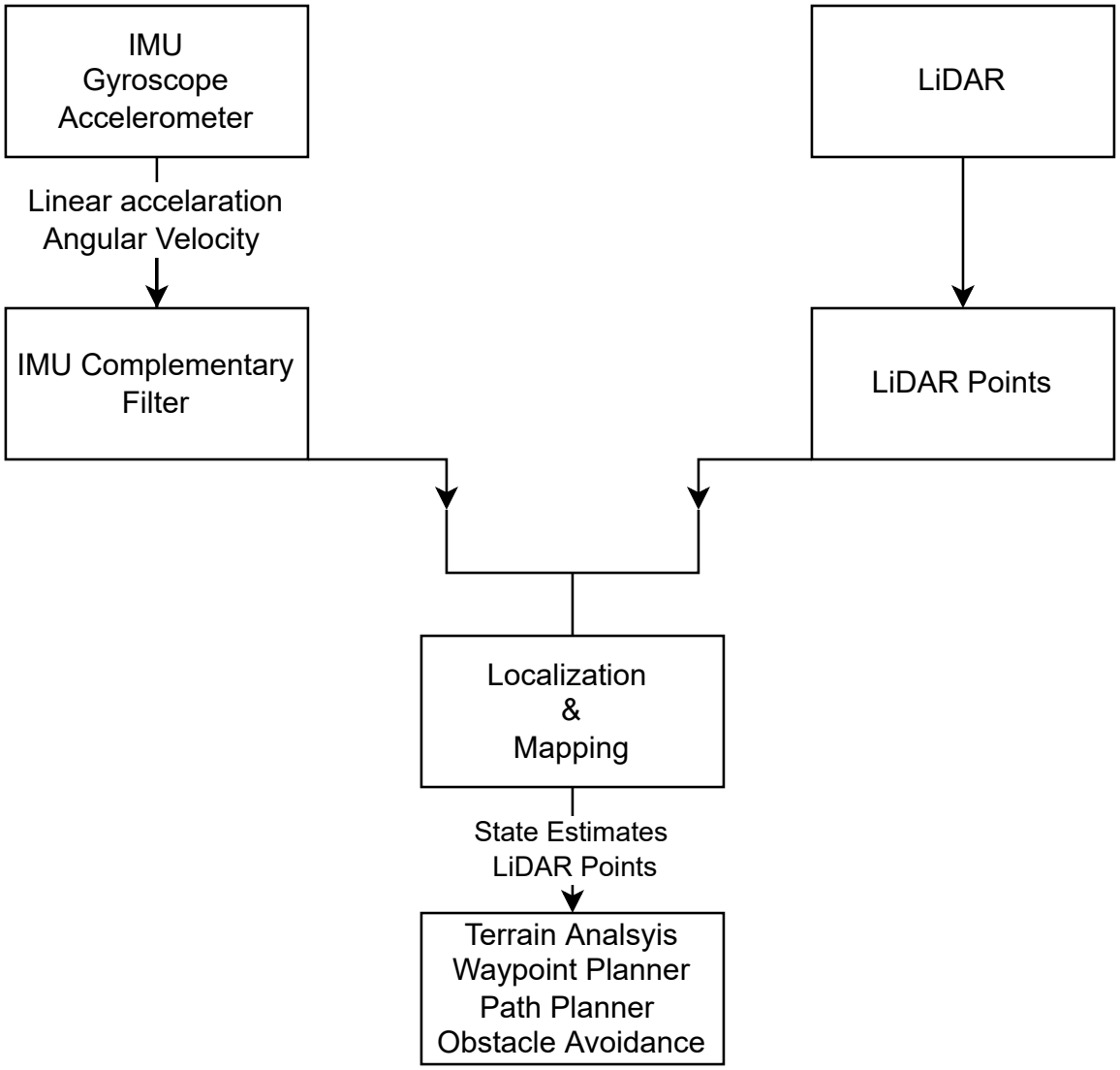


Figure 3.0.1: Diagram of Navigation Module

$$f(x) = \begin{cases} 1, & 0 \leq x < 0.1 \\ -10x + 2, & 0.1 \leq x < 0.2 \\ 0, & x \geq 0.2 \end{cases} \quad (3.8)$$

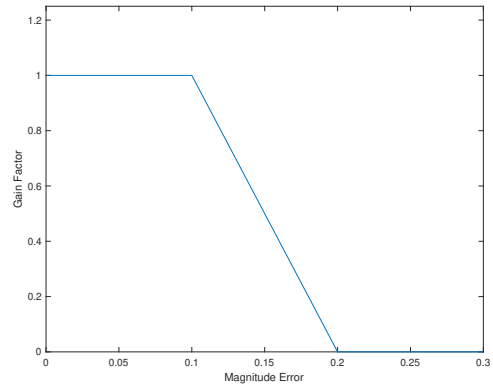


Figure 3.1.1: Picewise Function. Recreated by author based on [135]

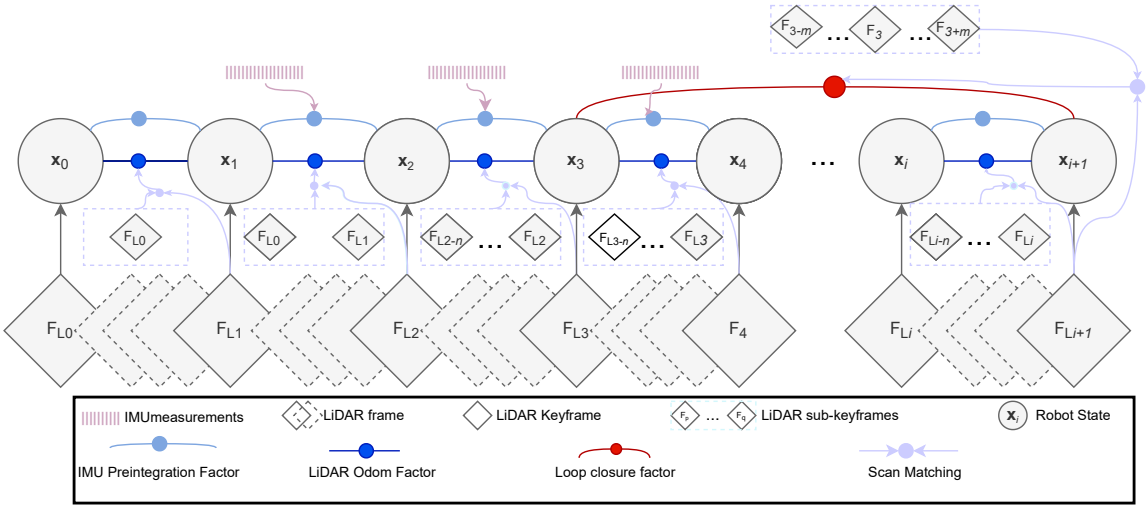


Figure 3.2.1: LIO-SAM Factor Graph. Recreated by author based on [118]

4

Technical Approach

Since the UGV is being developed to assist in the monitoring of previously mined sections of underground stone mines and a warning system, the system has to:

1. Traverse the mine terrain autonomously
2. Avoid obstacles and untraversable areas
3. Create high-resolution 3D maps of the mine and columns
4. Operate for extended periods of time of up to eight hours per day

With the desired mission specifications known, several systems were considered to successfully complete the tasks. After conducting trade studies to determine the system(s) that will be employed, the three that stood out were the applications of UAV and/or UGV systems. With the advancements made in UAV technology over the years, researchers considered the potential of the UAV to conduct surveys quickly as well as the ability to scan tall columns. But due to the vast size of both the columns and mines to be surveyed, the limitations in compact battery technology, and the required operational time, a standalone UAV system would not be feasible without having to recharge. Although the operational times of UAVs are limited due to the battery capacity, the system is highly advantageous for the application due to its ability to survey quickly, and scan tall columns. An additional benefit of the selection of the UAV by the researchers is their familiarity with the system from other projects. With the UAV being a potential system, research was conducted into potential charging setups being incorporated into the system.

In determining charging modules, some factors considered were the mission requirement of an operational time of up to eight hours per day, and the average flight time of most UAVs being 15-20 minutes. In order to extend the flight times, several methods were considered to meet the requirements. Some potential solutions were swapping batteries, wireless charging, multiple drones, and tethered power were considered. Although the mission requires an extended operational time of eight hours, the mine is only available one day per week for surveys to be conducted. Considering the limited availability of the mine, wireless charging will not be feasible since it will take several hours for a complete charge. Alternatively, swapping batteries although an easier substitute, will reduce the operational time since the drones will have to travel through previously scanned sections to the base station for the change to happen (same for wireless charging). Even though the

deployment of multiple UAVs has the potential to survey larger areas of the mine, it complicates the system.

Alternatively, the application of a standalone UGV system was considered due to its simplicity and high payload capacity. In order to scan the tall columns, the UGV will have to be designed with a telescoping boom which could be challenging for operators to transport easily. Therefore, the researchers decided to investigate the use of tethered power systems which will extend the operational time, with no downtime for charging or swapping batteries. The combination of the UAV and tethered system lead to the introduction of an unmanned ground vehicle for providing power, housing the tether system, and a stable platform.

To be able to successfully complete all these requirements, some desired specifications were considered during the design process of the UGV [32]. These specifications were selected by leveraging experiences from previous projects, research conducted on current warning methods, and the conditions in stone mines. The requirements:

1. Shall operate autonomously for eight hours per day
2. Shall operate in tandem with the UAV
3. Providing a support platform for the UAV
4. Size: to be able to fit standard service elevators
5. Drive-train system that maintains contact with terrain to allow for a stable platform
6. Robust and reliable system to allow for continual operations
   (a) The robustness of a system means that the system can function properly even when there is an algorithmic or hardware failure.
7. Relatively simple and serviceable system to allow missions to be conducted by two operators

Due to the stated mission specifications and design requirements, the design of the unmanned ground rover was split into three sections which are the Mechanical, Electrical, and Software overview. The mechanical overview consists of all the requirements needed to be able to drive the UGV in a stone mine environment. The electrical overview comprises the sensors and instruments needed

to operate the UGV autonomously and complete all its missions. The Software Overview delves into methods applied to successfully complete its missions.

4.1 Mechanical Overview

Several drive train systems were considered by the research team in order to successfully complete the missions. In the selection of the type of UGV to support the UAV, several types of unmanned ground vehicles such as wheeled, tracked, and legged robots were considered. Although there has been significant progress in the development and applications of legged robots, these systems are more complex. After conducting research into systems that are currently available, the research team soon discovered none of the readily available products have the payload capacity or the operational time required. Alternatively, tracked vehicles were considered, which can traverse steeper slopes and obstacles easier than most suspension systems, but are not very robust, very heavy, and require a lot of parts [71]. After conducting trade studies and from past experiences, researchers decided to use unmanned wheeled mobile platforms to support the UAV for simplicity and serviceability.

In the design of the structure of unmanned wheeled vehicles, there are several factors considered depending on the tasks needed to be completed. Some of the structural considerations are the number of wheels, wheel orientation, and wheel arrangement. The number of wheels used is important to ensure the stability of the system, with three nonaligned wheels being the minimum to achieve stability [50]. There were several types of wheel arrangements that were considered in the design process since this factor dictates the steering scheme and characteristics of the vehicle. The steering schemes that were considered were Ackerman, independent, and skid steering suspension systems. The Ackerman steering consists of a 4-bar trapezoidal mechanism that avoids wheel slippage by allowing the front wheels to be steered. An advantage of this steering mechanism is that it can be implemented with as few as two actuators with one for steering both wheels and the other for traversing. Although the Ackerman steering mechanism is relatively simple, it has a large turning radius which would be challenging for the system when maneuvering in tight spaces in the mine environment. Alternatively, with the independent steering scheme, each wheel is controlled individually and moved to the desired rotational speed and angle. Although the scheme allows the vehicle to have full mobility, it requires many parts and actuators, complex control algorithms, and

can be very expensive. Consequently, the skid steering scheme which can also be considered as a differential drive locomotion is the simplest structure for unmanned wheeled ground robots, consisting of four fixed active wheels. Steering is accomplished by actuating each side at different rates or in a different direction, causing the wheels to slip on the terrain. Differential drive locomotion is commonly used in tracked vehicles, such as tanks and bulldozers.

Due to the simplicity, ease of serviceability, and familiarity of this system to the researchers, the skid steer scheme was selected for the unmanned ground vehicle. The Unmanned Ground Rover (UGV) named Rhino is designed to have a split-body chassis. The split body chassis means the UGV consists of two halves with a turntable to connect both halves. This allows all four 0.5-meter tires to come into contact with the ground even on uneven terrain. Rhino also has to be able to house all the sensors and electronics to achieve autonomous operation. The UGV should also be able to transport the UAV, tether system, and batteries to extend the operational time for the UAV. Fig. 4.1.1 shows the 3D CAD rendering of the system which was used as a reference in the manufacturing process. For ease of manufacturability, the chassis was designed to prioritize the use of waterjet parts and minimize the number of Computerized Numerical Control (CNC) parts. Rhino was manufactured and partially assembled by West Virginia University's Lane Innovation Hub [125].

The chassis is designed with four main compartments to carry all the sensors for the operations of the UGV as shown in Fig. 4.1.3. One compartment is dedicated to housing all the power modules, the other for the computing module and the other two house the motor controllers and other auxiliary electronics. For the protection of the chassis and sensors, bumpers for the front and back as well as a roll cage were designed. The roll cage and the bumpers not only protect the UGV but also serve as mounting points for some of the electronics. The rear chassis of Rhino also carries the batteries, tether system, and landing platform of the UAV. To be able to operate for eight hours, the rover has to be able to carry high-capacity batteries. The chassis of the rover is designed to carry twelve 12V batteries with six dedicated to the operations of the UGV and six to the UAV and all its components. Fig. 4.1.2 shows Rhino in full configuration with the UAV system.

Figure 4.1.1: CAD Rendering of Rhino. Developed by Dylan Covell, Trevor Smith, and Gio Molin

Figure 4.1.2: Rhino in full configuration with Oxpecker (UAV). UGV was developed by Jonas Amoama Bredu Jnr, Dylan Covell, Henry Vos, and Christopher Arend Tatsch. UAV system developed by Bernardo Martinez, Rogerio Lima, and Jeremy Rathjen.

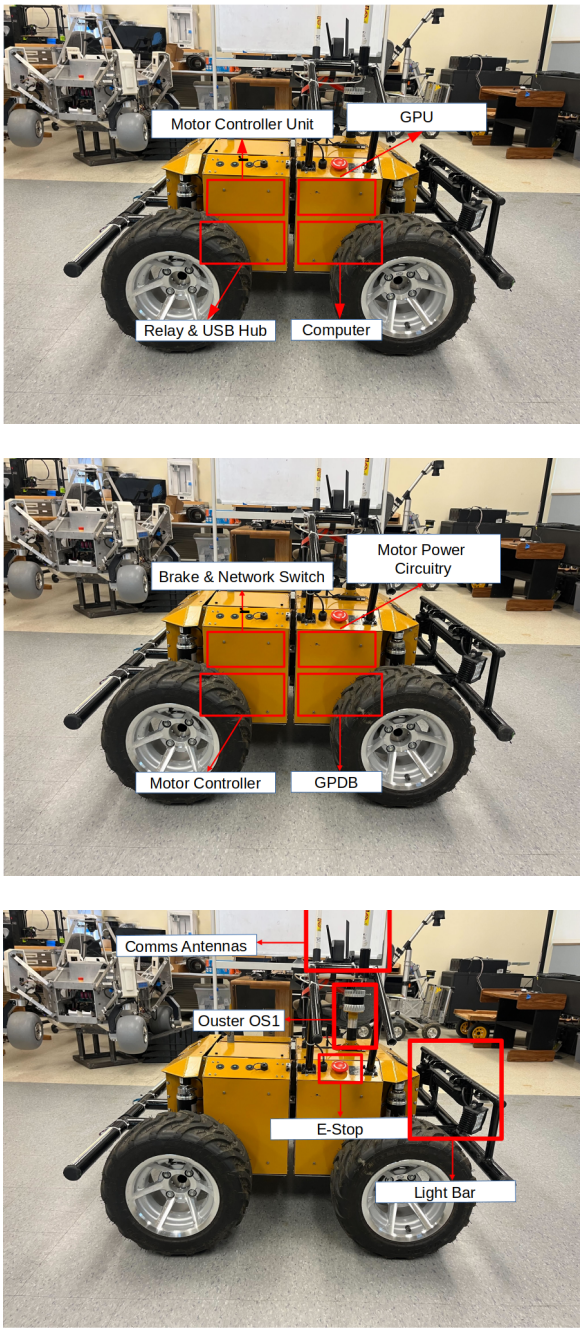


Figure 4.1.3: Photograph showing Rhino with its components labeled

4.2 Electronics and Instrumentation

In order to complete its operation (autonomous missions) as well as have the basic capabilities of being teleoperable, Rhino houses all the electronics and sensors to complete its mission. Fig. 4.2.2 and Fig. 4.2.3 show the layout of all the electronics needed to successfully complete its mission. Leveraging experiences from previous projects the General Power Distribution Board (GPDB) is used to power all the sensors. The GPDB is designed to receive inputs from batteries of different voltages (18-75 VDC) and regulate them to a range of voltages. The GPDB's regulators instrumented for Rhino are 12 VDC, 15 VDC, 24 VDC at 60 Watts, 5 VDC at 30 Watts, and 3.3 VDC at 13 Watts. These are essential since all the sensors and electronics require different voltages to operate. To power sensors and/or electronics that are not within the voltage range of the regulators, an external battery that provides the necessary voltage can be used as inputs in the unregulated section of the GPDB. For safety and emergency situations, power to all the electronics in the regulated and unregulated sections can be switched on or off via an emergency-stop (e-stop) switch.

The four motors instrumented on Rhino are LST11P 24 VDC 120 revolutions per minute (RPM) wheelchair motors with a gear reduction of 45.5:1 with electromagnetic brakes. In order to control the brakes during Rhino's missions, the electromagnetic brakes are interfaced with a relay. The motor controllers that were chosen to interface with the motors are the Roboteq MDC2460, which are dual-channel brushed direct motor controllers.

The initial setup was to power all the components through the GPDB but it could not handle such a high current draw from the motors. The high current demand of the motors damaged the GPDB as shown in Fig. 4.2.1, by burning traces as well as desoldering a Schottky diode designed to protect the board.

To allow the motors to draw as much current as it needs to operate, another circuit was designed. The circuit involves adding a relay, fuse, fly-back diode, an emergency-stop (e-stop) button, and an In-Rush current limiter. The In-Rush Current limiter protects the motor controllers from the large impulse of current when the device is switched on. Without the In-Rush current limiters, as the systems continuously start and stop during operations would lead to damage to the motor controllers. The relay and e-stop act as a switch to power the motors on and off and also allows power to the motors to be cut off in emergency situations. Fig. 4.2.2 shows the circuitry designed

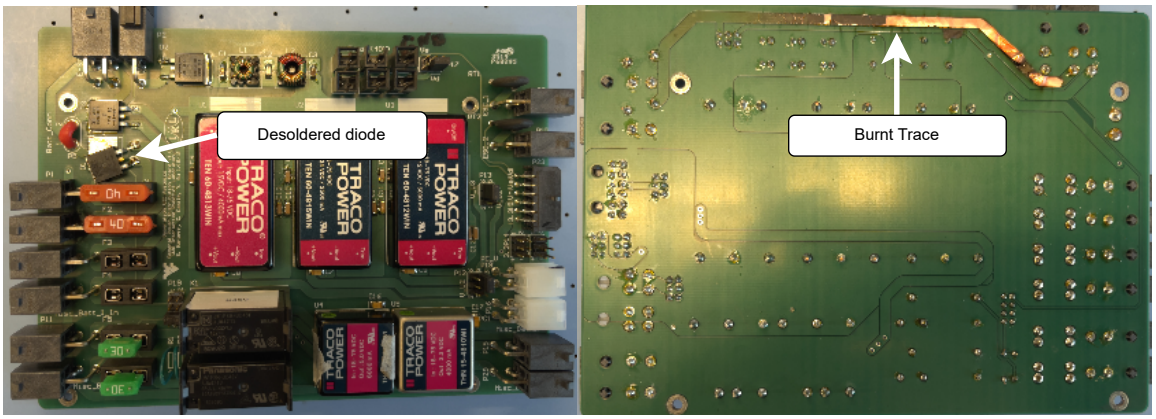


Figure 4.2.1: Damage caused by the high current draw of motors

to power the motors as well as the back Electromotive Force (EMF) protection circuitry.

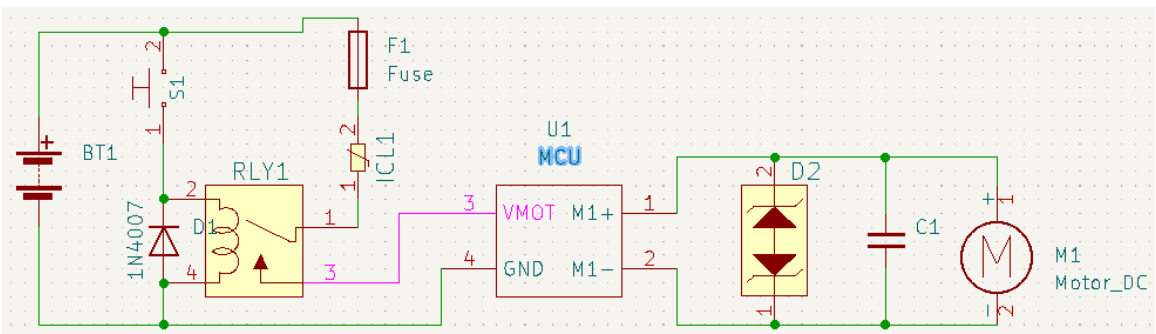


Figure 4.2.2: Motor Power Circuit. S1: switch, D1: diode, RLY1: relay, ICL1: inrush current limiter, MCU: motor controller unit, D2: TVS diode

During initial testing, it was observed that when the Rhino was brought to an abrupt stop when driving, the motor controller would shut down. After investigations, it was discovered that this behavior was a result of a high voltage transient caused by the back EMF due to the collapsing magnetic field inside of the motors. The magnitude of this back EMF was exacerbated by the size of the motors and the high inertia of Rhino due to its large mass. To stop the back EMF from the motors from being fed back into the motor controller, a circuit was designed for protection. The back EMF would cause the motor controllers to shut down as a safety mechanism because the back EMF pulse exceeded the motor controller's maximum voltage input. The circuit designed is isolated from the

General Power Distribution due to the sensitivity of the rest of the sensors. To deal with the issue of the back EMF shutting down the motor controllers, a transient voltage suppression (TVS) diode and a bypass capacitor were put in parallel with each motor's leads. The TVS diode protects sensitive electronics (motor controllers) from high-voltage transients and overvoltage events faster than most other types of circuit protection devices [4]. While the capacitor reduces the voltage pulsation and also provides a low-impedance path for the back EMF to be dissipated.

Rhino is equipped with Dual Carbine lightbars for improved visibility for the operator during teleoperation missions. The lightbar interfaces with a relay which allows the operators to control the lights when conducting missions in low lightning conditions in the mine. The system allows operators to send commands via a joystick to either turn on or off the lights.

The next paragraph delves into the sensors selected to complete the mission successfully. The core sensors required to complete the mission are segmented into five sections which are

1. Localization and Perception
2. Communications
3. Actuators
4. Processing Unit
5. Accessories

The actuators, electronics, and sensors have been described in the paragraphs above. For localization and perception, the sensors selected are a Light Detection and Ranging (LiDAR) sensor, an inertial measurement unit (IMU), a fiber optic gyroscope, and depth sensing cameras. The sensor measurements will be fused for localization, path planning, obstacle avoidance, object tracking, mapping, and object classification. Fig. 4.2.3 displays the overview of the rover's electronics and sensors.

Data is collected and processed using a 3.6 GHz Intel i7-9700K processor with eight cores, a Corsair 32 Gigabyte (GB) RAM with a speed of 2400 MHz in a GIGABYTE Z390 M Mini Atx computer. For processes that require graphical computation and visualization like SLAM, there is

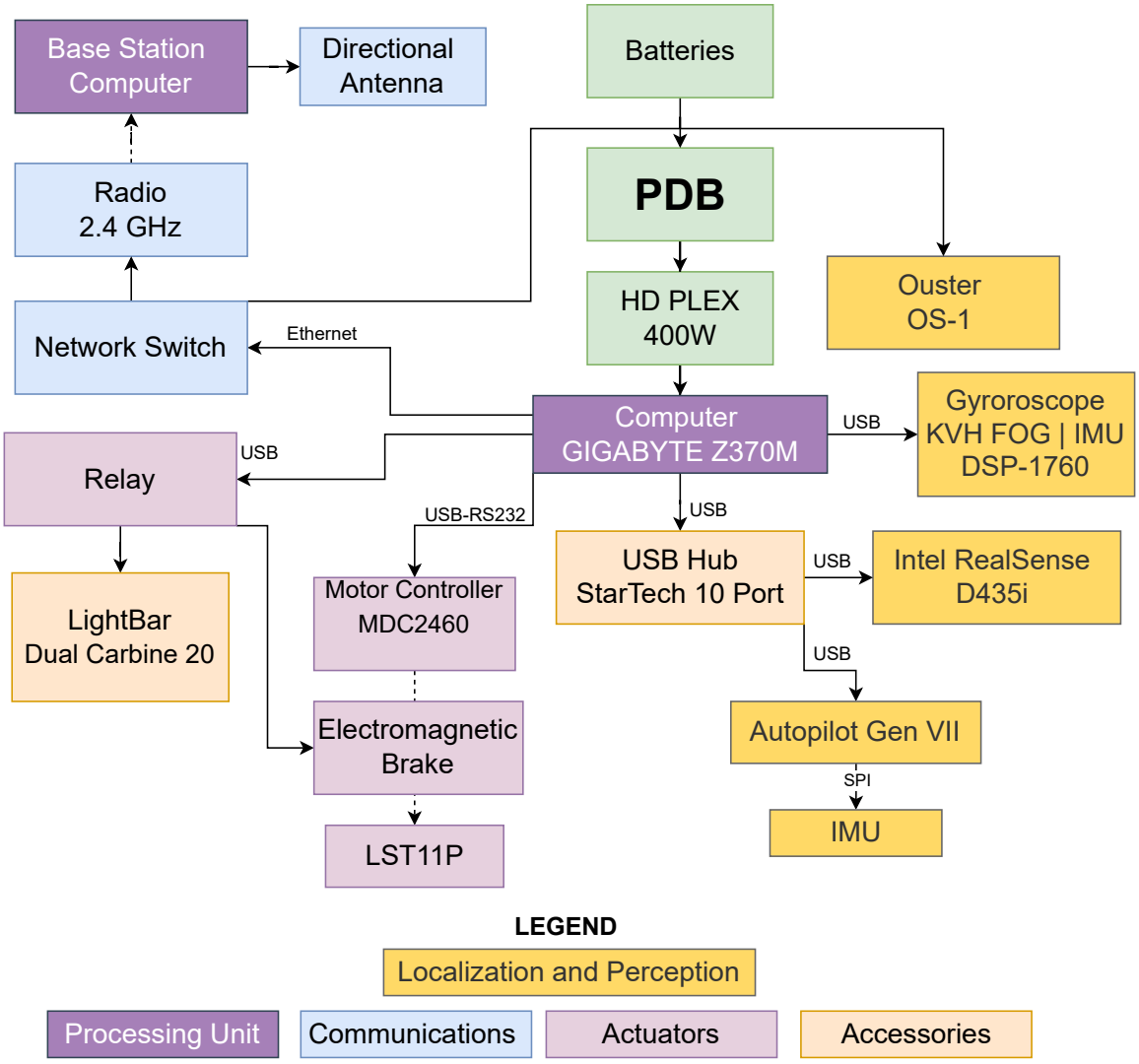


Figure 4.2.3: Schematic for Electronics and Sensors

a Gigabyte GeForce GTX 1650 OC Low Profile 4G Graphics Card with a four gigabytes of RAM and 8002 MHz graphics processing unit (GPU) speed.

The LiDAR has the ability to generate point clouds that provides updated distance measurement, with centimeter-level accuracy, between the robot and the surrounding objects. This sensor's ability to operate is not impeded by low lighting conditions, so will be operable in mines. The LiDAR is

an essential sensor for this project because of its many capabilities and uses. The LiDAR selected is a 64-channel Ouster OS-1 [101] with a 100m range, range accuracy of $\pm$3cm, range resolution of 0.1cm, and interfaces with the computing unit via the network switch. The depth cameras mounted in the front and the rear of the UGV have a similar purpose as the LiDAR but are mounted in the blind spots of the LiDAR. With the combination of LiDAR and depth cameras, the rover can have a better perception of the environment.

The selected IMU is the ADIS16495 by Analog Devices [12], a tactical grade six degrees of freedom inertial sensor. The sensor has a triaxial digital gyroscope and a triaxial accelerometer. The specifications provided by the manufacturer are in-run bias and angular random walk values are $1.6^o/hr$, $0.1^o/\sqrt{hr}$ for the gyroscope, and 3.2 μg, 0.008 m/sec/$\sqrt{\text{hr}}$ for the accelerometer, respectively. The IMU measures the angular velocity which is integrated to infer the orientation and linear acceleration of the vehicle. There is a second IMU mounted in the back half of the chassis mainly to estimate the orientation. The IMUs interface with the system via the Autopilot Gen vii, [56] which uses the Netburner MOD54115 microcontroller that collects and processes the data. The implementation of the Netburner which has a 250 MHz processor, allows the system to receive IMU data at a rate of 100 Hz via the Serial Peripheral Interface (SPI).

Fiber optic gyroscopes offer higher sensitivity, its rotational sensitivity is high grade and faster in comparison to the gyroscope from the IMU. The fiber optic gyroscope senses changes in orientation using the Sagnac effect to measure the angular velocity of the UGV. The selected gyroscope is a KVH DSP-1760 single-axis fiber optic gyroscope [79] with the specifications of an in-run bias and angular random walk values are $0.05^o/hr - 1\delta$, $0.012^o/\sqrt{hr}$ for the gyroscope, respectively. Both the fiber-optic gyroscope and the IMU allow the measurement of the angular velocity and linear acceleration to provide an estimation of the orientation of the UGV in its environment.

The communication system was selected based on previous experiences by Team Mountaineers at the University Rover Challenge where long-distance non-line of sight communication is a requirement [55]. The system runs off a 2.4 GHz and 900 MHz frequency radio system by Ubiquiti. The 2.4 Ghz radio system consists of a high-gain directional antenna at the base station and for the rover side, there are two omnidirectional antennas. Prior to testing in the mine, the 2.4 gigahertz radio system provided up to 0.5 kilometers of non-line-of-sight communication. The communication system will be able to transmit crucial information back to the base station while the rover is operating autonomously or teleoperated. Information like the health status, orientation, battery

level, and also the point cloud data are transmitted back to the base station for monitoring.

The complete instrumentation of Rhino, including all sensors, electronics, and batteries, weighs approximately 190 kilograms. Table 5 presents the breakdown of the weight the system required to complete all the tasks required.

Table 4.2.1: Weight Estimates of Rhino

Component	***Value (kg)***
Chassis	56.41
Drive Motors	14.47
Wheels and Tires	36.46
Sensors and Electronics	10.64
Batteries	73.21
Total weight	191.20

4.3 Software Overview

To complete the required tasks required by the project, the actuators, sensors, and software have to work seamlessly. The researchers elected to use the Robot Operating System (ROS) framework, which defines components, interfaces, and tools for building robots [3]. Since the UGV is made up of the frame, batteries, actuators, sensors, control systems, etc, ROS allows for their interconnectivity through topics and messages which makes development and testing easier. Figure 4.3.1 shows the software the researchers developed and implemented for the hardware interface, state estimation, planner, traverse, and teleoperation modules. The software is structured to allow for basic missions such as teleoperation and more challenging tasks such as autonomous navigation, planning, and mapping. The developed software allows the different sensor measurements and modules to have individual nodes, topics, and messages. The nodes are responsible for computations and can receive inputs from sensor measurements [104]. Different nodes may publish or subscribe to messages through topics to exchange information. Figure 4.3.1 shows the software overview for Rhino for all missions.

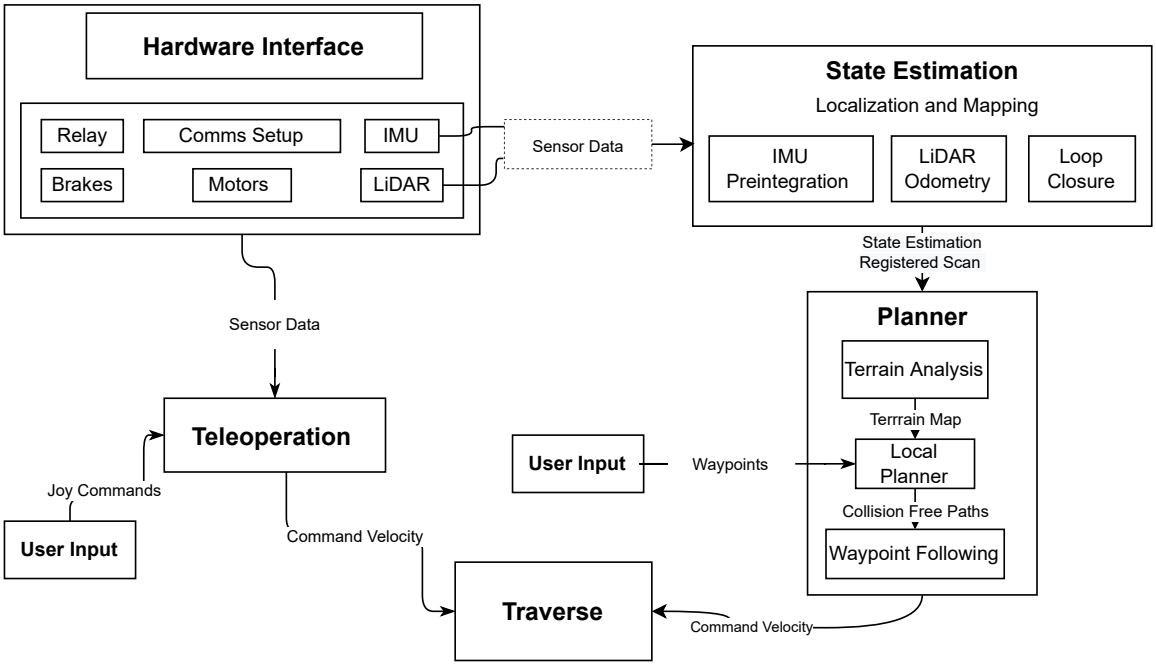


Figure 4.3.1: Software Overview

The hardware interface layer of Figure 4.3.1, involves developing software that allows for communication with the sensors and electronics on Rhino which are essential for all the modules. In order to communicate with the actuators, custom software was developed to allow for motor commands to be sent to the motors by either a user or the planner. Once a motor command is sent, the node sends feedback messages from the encoder to determine the state of the motor. For the safety of operators and the system, a node was developed to allow users to engage and disengage the brakes. The node publishes the state of the brakes and can also receive inputs from users. To interpret sensor outputs from the LiDAR and IMU, nodes were also developed to allow the measurements to be used. The IMU node converts sensor readings from the gyroscope and accelerometer into angular velocity and linear acceleration measurements which can then be used by the state estimation and planner modules. While the range measurements from the LiDAR are converted into points by the LiDAR node and are also used by state estimation and planner modules as well.

The teleoperation module receives inputs from users and sensor data which are converted to command velocities to control Rhino remotely. It also receives user inputs via joy commands that

are then converted into command velocity messages. During teleoperation tasks, if the brakes are engaged, the command velocities are not sent to the motors.

Prior to the implementation of the autonomous navigation, localization and mapping modules in Chapter 3, it is essential to calibrate the intrinsic and extrinsic parameters of the sensors. The accuracy of the calibration performed directly affects the performance of the algorithm being implemented. To effectively fuse all sensors, especially for localization, mapping, and autonomous navigation, the sensors required need to be spatially and temporally registered with respect to each other. IMUs are particularly affected by errors that lead to inaccurate sensor readings. Errors such as constant bias, white noise/angle random walk, bias stability, and calibration errors. Most non-mechanical IMUs are affected the most by intrinsic errors such as white noise and uncorrected bias errors. White noise-introduced errors cause an angular random walk whose standard deviation increases proportionally to the square root of time [138]. While uncorrected bias causes errors in orientation estimates that increase linearly over time [138]. To mitigate these unavoidable errors, the IMU was calibrated by calculating the white noise and bias instability of both the gyroscope and the accelerometer using the Allan Variance method. The Allan Variance method is a time domain analysis technique that represents the root mean square (RMS) random-drift error as a function of averaging time [44]. To calibrate these sensors, data was collected for a period of two hours when the robot was not undergoing any rotation or translation. An example of the bias and noise characteristics of the ADIS 16495 used on Rhino are shown in Table. 4.3.1.

Table 4.3.1: ADIS 16495 Error Parameters

Parameter	***Value***	**Symbol**	**Unit**
Accelerometer White Noise	1.813e−02	δ_a	$\frac{\mathrm{m}}{\mathrm{s}^2}\frac{1}{\sqrt{\mathrm{Hz}}}$
Gyroscope White Noise	9.204e−04	δ_g	$\frac{\mathrm{rad}}{\mathrm{s}}\frac{1}{\sqrt{\mathrm{Hz}}}$
Accelerometer Bias Instability	8.293e−05	δ_{ba}	$\frac{\mathrm{m}}{\mathrm{s}^2}\sqrt{\mathrm{Hz}}$
Gyroscope Bias Instability	7.414e−06	δ_{bg}	$\frac{\mathrm{rad}}{\mathrm{s}}\sqrt{\mathrm{Hz}}$

With white noise and bias instability identified, they can reduce errors when determining attitude and position estimates by fusing LiDAR and IMU estimates in the state estimation module. Although these parameters have been identified, the IMU cannot be fully relied upon for pose estimates because the noise will drift over time. To mitigate the errors caused by drift and noise, the

state estimation module also estimates the biases to avoid errors which will be discussed in the paragraphs below. As discussed in Chapter 3, measurements from the IMU are used to infer roll, pitch, and yaw estimates.

Once the intrinsic parameters for the IMU were identified, the extrinsic calibration parameters for the IMU and LiDAR were calculated. For LiDAR-based localization techniques, LiDAR calibration is essential for successful operations. During operation, LiDARs generate 3D points which are grouped in scans, if the sensor is not properly calibrated the readings are affected by motion distortions when the agent moves through the environment. In order to obtain accurate localization and mapping estimates, the sensor is extrinsically calibrated with an IMU to track the motion distortion in each LiDAR scan. Through a factor included in the LIO-SAM framework [118] to deskew the LiDAR pointcloud, the angular rate from the IMU must be provided to the localization and mapping module in a frame aligned with the LiDAR plane. To make the transformation between the LiDAR and IMU simple to compute for this purpose, the robot was designed so that IMU was placed below the LiDAR with only a translation offset in the Z-direction. After the development of the hardware interface and calibration, the measurements from proprioceptive and exteroceptive sensors are used in the implementation of the algorithms in Chapter 3.

For the localization and mapping, LIO-SAM [118], the module requires IMU measurements that provide roll, pitch, and orientation estimates as well as LiDAR pointcloud data. The task of localization and mapping can be categorized into four sub-modules which are IMU preintegration, image projection, feature extraction, and map optimization. The roll and pitch estimates are used to initialize the unmanned vehicle at the correct attitude. The IMU preintegration factor receives IMU measurements and LiDAR odometry from the map optimization module which are then used for graph optimization and estimation of the IMU biases and outputs of the IMU odometry. The image projection module receives LiDAR point cloud measurements, IMU measurements, and odometry estimates from the IMU preintegration for the initial transformation and organizing of the point cloud data. Due to the acceleration of the unmanned vehicle, the LiDAR measurement can become deskewed which may lead to incorrect measurements. To mitigate the deskewing of the LiDAR point clouds, the image projection module is also responsible for deskewing the point cloud data. The feature extraction module is responsible for extracting the edge and planar features which can be used for state estimation. The goal of the map optimization module is for the registration of the point clouds, attaining the LiDAR odometry, and optimizing the graph.

For autonomous navigation missions, the planner [145] can also be categorized into three submodules that work in parallel with each other in order to complete the missions. The three submodules are the terrain analysis, local planner, and the waypoint following modules. The terrain analysis receives the registered point cloud data from the state estimation module which is used to assess the traversability of the terrain. The terrain analysis module accesses the traversability of the terrain by using LiDAR data to represent the environment as a voxel grid. The voxel grid is then used to determine the ground height by analyzing the distribution of the adjacent voxels which then determines the smoothness of the terrain. With the smoothness of the terrain determined, points that are further from the ground are assigned a higher traversal cost. With the traversability of the terrain determined, the local planner uses the cost map generated and state estimates to determine paths to the waypoints which are determined by the user. The local planner generates plans offline in the sensor frontier for obstacle avoidance while the global planner ensures the planned paths do not fall into a local minima in order to reach the waypoints. The planner generates collision-free paths by pre-computing motions the system can take and associates the motions with 3D locations close to the system. If a location is occupied by an obstacle, the module can determine pre-computed motions that are collided with obstacles and avoids those paths. The module instead selects a group of motion primitives with the maximum likelihood of reaching the waypoint

After the implementation of the state estimation and the planner, Rhino was tested in several environments and terrain to assess its capabilities and limitations which will be discussed in Chapter 5.

5

Experiments and Results

To successfully and reliably complete the tasks of localization, mapping, and autonomous navigation of subterranean environments, several tests were conducted to verify the capabilities and robustness of Rhino. To evaluate the performance of the system prior to missions in the limestone mines, several tests were carried out on the West Virginia University campus to validate the performance, expose issues, and fine-tune the system. The tests on the campus were carried out for both indoor and outdoor environments, which allows the assessment of the performance of the robot on a variety of terrain and environments. After conducting several tests, validating and fine-tuning the system, Rhino was deployed in a coal mine to perform localization, mapping, and autonomous navigation missions.

5.1 Experiments on the West Virginia Campus

Several tests were carried out on the West Virginia University Campus, to validate, expose issues, and fine-tune the performance of the unmanned ground robot Rhino. Initial tests were conducted to assess the capabilities through teleoperation by conducting tests on concrete, grass, tiled indoor surfaces, gravel, and muddy terrain. Once the performance of the drivetrain system was assessed, localization and mapping experiments were conducted to verify the state estimates and then autonomous navigation missions.

Initial tests were conducted indoors to validate and tune the response of the system to motor commands via user inputs to access its driving capabilities. Due to the weight and inertia of Rhino, acceleration limits were set to prevent damage to the gearbox caused by the system starting and stopping which was observed from initial tests. As a consequence of the current and acceleration limits set for hardware safety, the motor speed control was tuned according to the limitations, mass, start, and stop inertia. Tuning of these parameters are particularly important for autonomous missions where it will be critical for Rhino to come to a complete stop when a waypoint is reached or in the vicinity of an obstacle.

After initial parameter tuning, several teleoperation tests were carried out to assess the capabilities of the system on different terrain and slopes. With all the components and batteries required for localization, mapping autonomous navigation, and extended operational times, the weight of the robot (190 kg) is a significant factor in the ability of the actuator to move the robot. While the tiled surfaces are not representative of the intended surfaces in the mine, testing on these surfaces

at WVU was convenient for initial testing and for validating that the overall system was working. The system was able to handle the tiled surfaces well but marginally struggled when turning. After performing such tests, Rhino was driven outdoors to be tested on concrete, gravel, and grass. Although Rhino was able to be driven on different terrain, researchers realized the system struggled when commanded to turn as well as traverse inclined terrain. Even though full power was being commanded to the robots, Rhino was unable to turn in place on concrete and grass. It was then determined that the motors were not able to provide enough torque to overcome the weight of the robot and the coefficient of friction of the terrain. The researchers determined that Rhino was able to traverse on tiled surfaces easily due to these surfaces having a lower coefficient of friction in comparison to the other terrain. The system also struggled on such terrain due to the steering scheme being skid-steer, which requires the wheels to slip in order to turn. As a result of the skid-steer system, the higher friction coefficient of the other terrain and high weight made turning on concrete, grass, and gravel a challenge for the system.

The ability to turn on different terrain is essential for any and all operations, therefore research was conducted to improve the systems driving capabilities and turning ability on different terrain. The solutions that were considered were switching the tires or upgrading to higher torque motors. The researchers chose to upgrade the motors to higher torque motors since it did not require any significant change in the system. The actuators were upgraded from motors with a gear ratio of 32:1 with a torque specification of $30\ N \cdot m$ to motors with a gear ratio of 45.5:1 and a torque rating of $42.66\ N \cdot m$.

After the upgrade of the motors, the researchers saw a significant improvement in the driving capabilities of the system, with the system being able to turn on grass, concrete, gravel, and the ability to traverse inclined terrain. Experiments to verify and assess the capabilities of Rhino required extensive testing and extended periods of time, which allowed validation of the operational time of the 12V 42 Ah batteries. The researchers discovered that after extensive testing, the battery modules will allow for up to 3-4 hrs of continuous power supply to the system. But some issues were discovered after such tests, if the batteries are continuously used without being charged, once the batteries are fully drained the modules go into a peculiar state. As discussed in Chapter 4, the batteries are in series to provide 24V power to the system, so one of the batteries drains faster than the other. From further examinations and quantitative experiments, the researchers realized that the battery monitoring systems go into an error state whereby it falsely indicates a full charge. To

mitigate these issues, the batteries were upgraded to 24V 50 Ah batteries which simplified the battery system without the need to put the system in series. The upgrade allowed the researchers to conduct more tests without the need to worry about issues with the power module.

Once the issues discussed in the previous paragraphs were addressed, the localization and mapping module [118] was tested in several indoor and outdoor environments. Although ground truth was not available for indoor tests, the generated 3D maps were evaluated qualitatively. The tests were carried out to access Rhino's localization and mapping capabilities prior to autonomous navigation missions. During the missions, the SLAM parameters were tuned in order to improve the performance of Rhino. The first parameter that was tuned was the voxel size which is different for indoor and outdoor experiments. The voxel size determines how granular or coarse the system interprets the environment, by using smaller voxel parameters more features can be extracted. Although more features can be extracted, the system will become more computationally expensive if there are too many features. To reduce the computational complexity and processing time, it was determined that the indoor parameters performed better with smaller voxel parameters than the outdoor parameters since there were usually more features. The loop closure parameters were then tuned to avoid inaccurate loop closures such as the distance from the current position of Rhino to consider a loop closure and the number of keyframes that are fused into the submap for a loop closure. Fig. 5.1.1 demonstrates a map generated indoors, with multiple loop closures, where the yellow lines represent multiple loop closures.

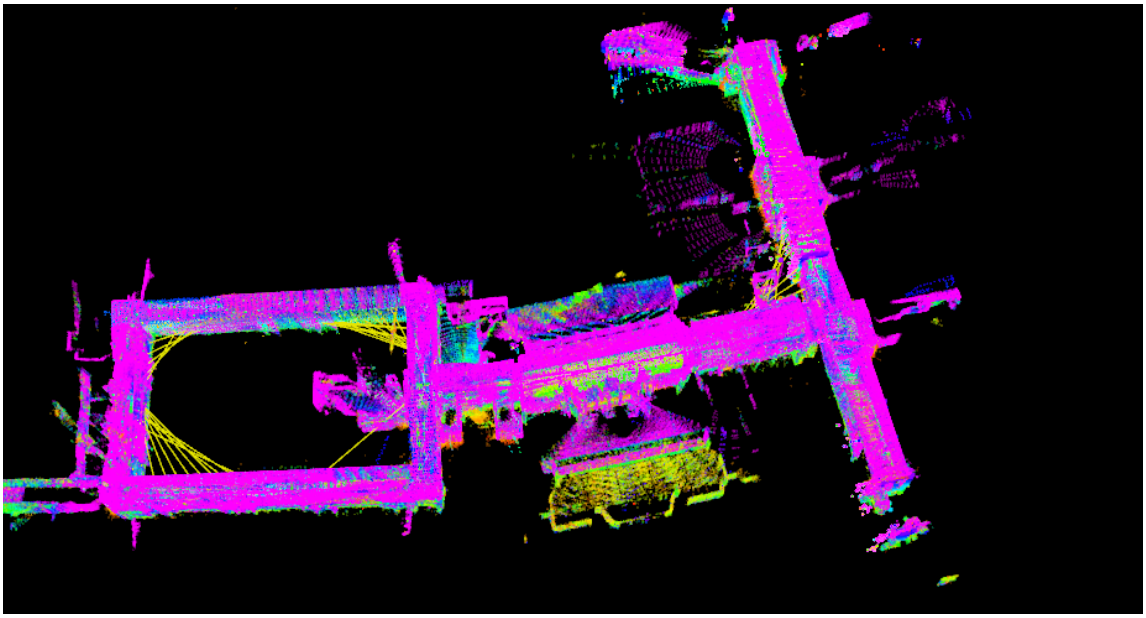
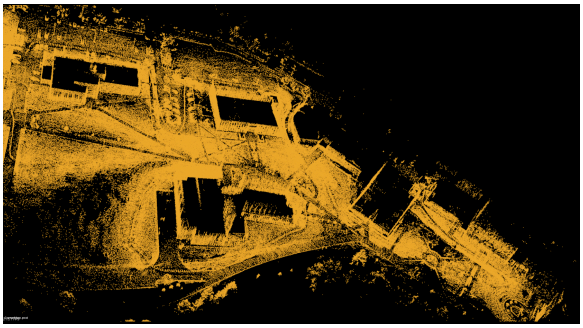
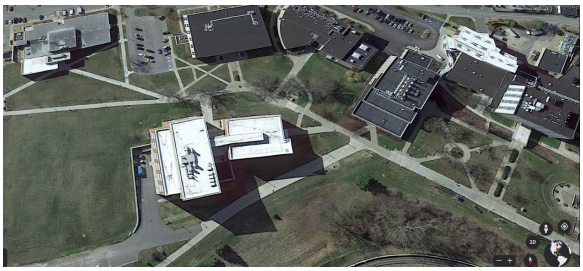


Figure 5.1.1: 3D Map of Advanced Engineering Building, WVU Second Floor

Fig. 5.1.2 shows a comparison of an outdoor 3D map generated by LIOSAM and Google Earth on the West Virginia University campus.



(a) 3D generated Map of West Virginia University, Evansdale Campus



(b) West Virginia University, Evansdale Campus, Google Earth view for comparison

Figure 5.1.2: Comparison of 3D generated map of the Evansdale Campus, West Virginia University

After the performance of the localization and mapping module [118] was assessed, the autonomous navigation module [22, 145] was tuned for the dynamics of Rhino. For Rhino to autonomously navigate to the different waypoints provided by the user, the terrain analysis module [22] was tuned for Rhino's setup such as the dimensions of the vehicle, the height of the LiDAR from the ground, the obstacle height threshold which determines which obstacles the system can traverse over, the speed and acceleration when autonomously navigating. To verify the state estimates from the localization and mapping module the user-given waypoints were used as inputs in the autonomous navigation module to allow the system to drive to multiple waypoints for an extended period of time and compared with ground truth from GPS measurements. The truth reference solution is determined by a carrier-phase differential GPS (DGPS) setup. The setup for the DGPS solution consists of two dual-frequency Novatel OEM-615 GPS receivers, and L1/L2 Pinwheel antennas, one mounted to the rover and the other mounted on a base station [74, 75]. The solution is collected with 1 Hz

GPS pseudo-range and GPS carrier phase on both base station and rover computer, then it is post-processed using RTKLIB 2.4.2 [129].

In order to compare the GPS truth and the pose estimates, the initial time of the estimates from LIO-SAM had to be synchronized to align data. Since the GPS solution was published at a rate of 1 Hz and the localization at a rate of 10 Hz, the GPS data had to be interpolated in order to have an equal number of data points for comparison.

In an effort to access the capabilities of the system, one of the tests conducted involved Rhino autonomously navigating between two waypoints. During the tests, Rhino traversed 9km for 3.6 hours with the longest autonomous mission being 2.5 hours. In the course of the 2.5 hr mission, Rhino autonomously traversed 4.5 km on gravel and sandy terrain. Fig. 5.1.3 shows the map generated by the LIO-SAM [118], the planner and the trajectory Rhino travelled. The blue lines represent the trajectories traveled and the yellow arcs represent the free paths from the local planner [145]. Although the trajectories were different for each run, Rhino consistently arrived at the waypoint each time.

Fig. 5.1.4 and Fig. 5.1.6 show the comparison of the pose estimates between the GPS truth, the optimized LIOSAM odometry, and LiDAR odometry estimates over the time period the tests were conducted. The figure shows that although both odometry estimates were similar in the beginning, as the test continued, the LiDAR odometry slowly drifted.

Fig. 5.1.5 and Fig. 5.1.7 represent the difference between the GPS truth and the pose estimates from the odometry solutions from the localization and mapping module. The figure confirms the trend that can be seen in Fig. 5.1.4, which shows that as the tests continued, the LiDAR odometry becomes less reliable after an extended period of time. The Fig. 5.1.4, Fig. 5.1.6, Fig. 5.1.7 and Fig. 5.1.5 shows that by fusing IMU measurements in the graph optimization improves the odometry estimates.

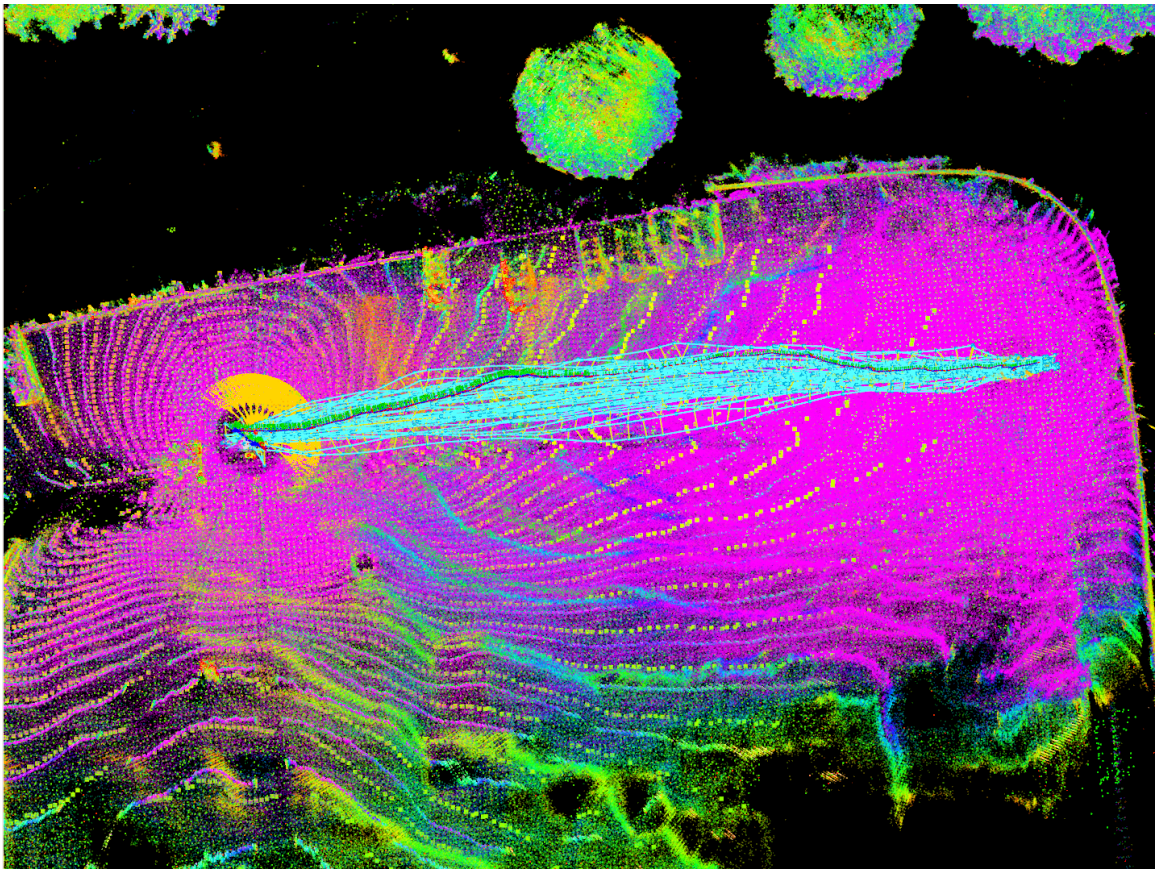
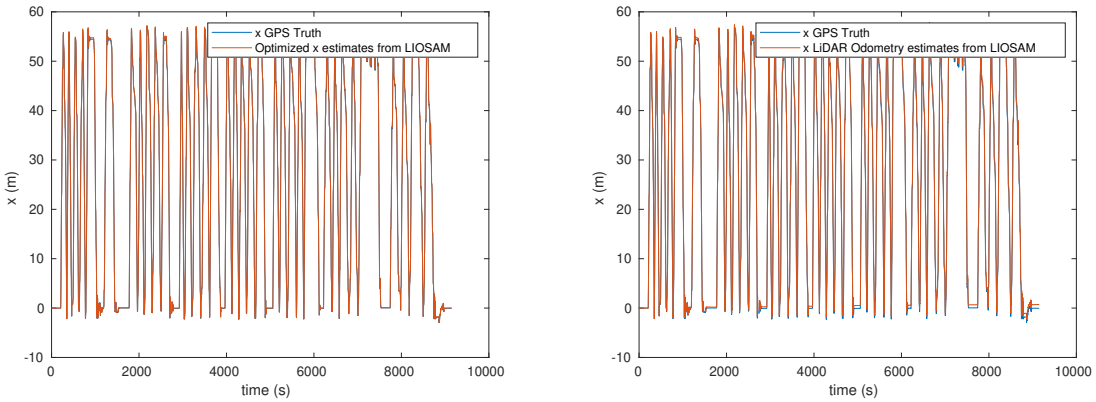


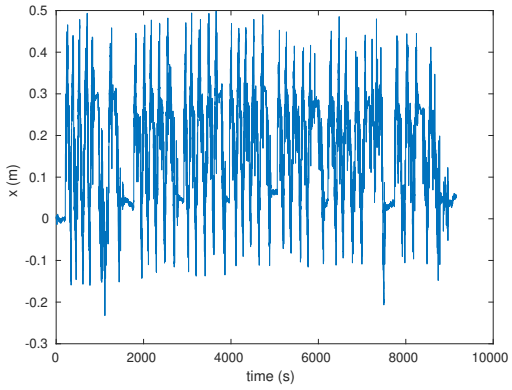
Figure 5.1.3: Map generated during the mission where Rhino traveled between 2 waypoints.



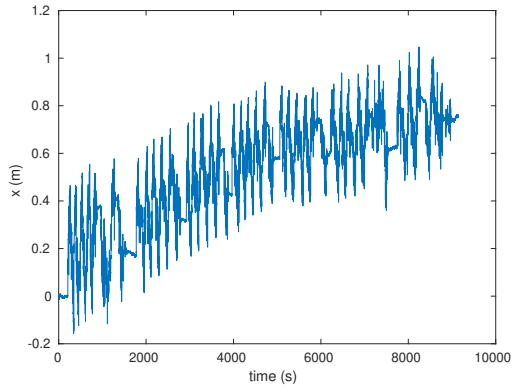
(a) x estimates for GPS truth and Optimized LIOSAM Odometry

(b) x estimates for GPS truth and LIOSAM LiDAR Odometry

Figure 5.1.4: Comparison of x pose estimates between GPS truth and the odometry solutions from LIOSAM for back and forth test

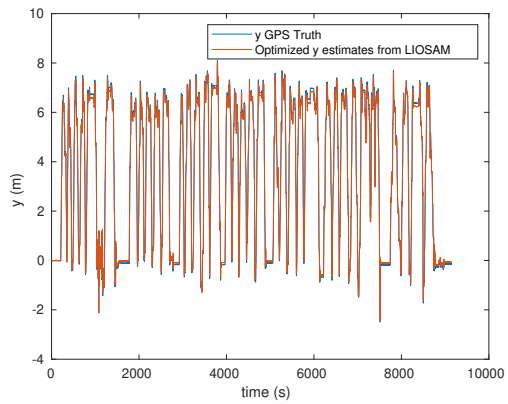


(a) x estimates for GPS truth and Optimized LIOSAM Odometry

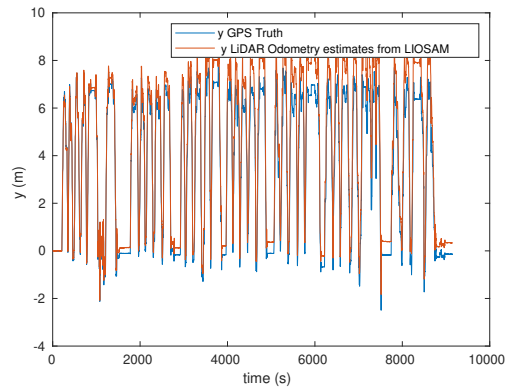


(b) x estimates for GPS truth and LIOSAM LiDAR Odometry

Figure 5.1.5: x Error comparison of x pose estimates between GPS truth and the odometry solutions from LIOSAM for back and forth test

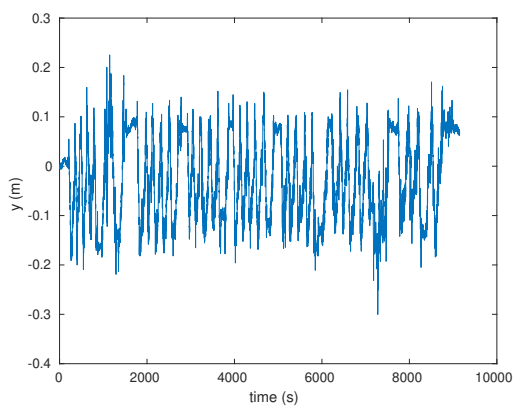


(a) y estimates for GPS truth and Optimized LIOSAM Odometry

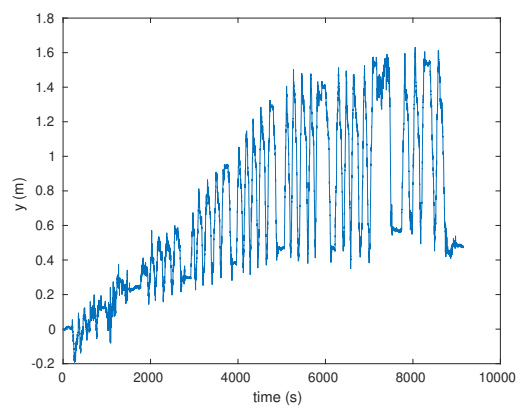


(b) y estimates for GPS truth and LIOSAM LiDAR Odometry

Figure 5.1.6: Comparison of y pose estimates between GPS truth and the odometry solutions from LIOSAM for back and forth test



(a) y estimates for GPS truth and Optimized LIOSAM Odometry



(b) y estimates for GPS truth and LIOSAM LiDAR Odometry

Figure 5.1.7: Error comparison of y pose estimates between GPS truth and the odometry solutions from LIOSAM for back and forth test

Table 5.1.1 compares the LIO-SAM [118] optimized pose estimates against the uncorrected LiDAR odometry which drifts in comparison to the optimized estimates.

Table 5.1.1: RMSE in East-axis and North-axis for Autonomous Navigation between two waypoints (2.5 hrs)

Reference System	*LIOSAM Optimized*	LIOSAM LiDAR Odometry
East (meters)	**0.207**	0.571
North (meters)	**0.0896**	0.821

For better visualization of the test, Fig. 5.1.8 shows the pose estimates from a shorter run time.

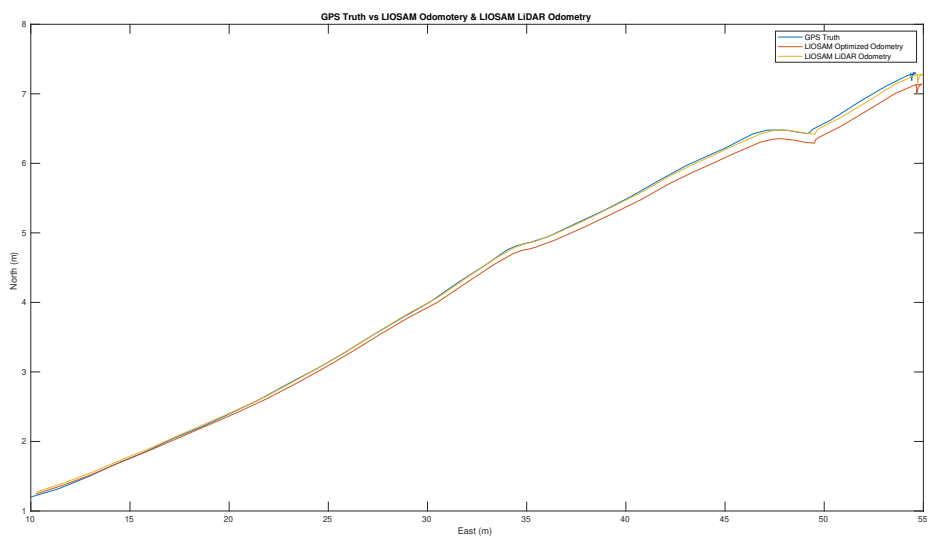


Figure 5.1.8: LIOSAM Odometry vs LIOSAM LiDAR Odometry vs GPS Truth between 2 Waypoints (shorter test)

Table 5.1.2 shows the same trend as the results from the test conducted for 2.5 hours.

For a more challenging experiment in comparison to the previous back-and-forth experiment, the system was given a set of waypoints to drive in a zigzag pattern. The set of zigzag waypoints allowed the researchers to verify the hardware (actuators and chassis) and the planner's [145] ability

Table 5.1.2: RMSE in East-axis and North-axis for Autonomous Navigation between two waypoints (shorter test)

Reference System	*LIOSAM Optimized*	LIOSAM LiDAR Odometry
East (meters)	**0.335**	0.571
North (meters)	**0.105**	0.821

to handle more challenging missions. During the zigzag missions, Rhino operated for an hour and traveled 2.3 km. Fig. 5.1.9 shows a segment of the mission where Rhino autonomously traversed 475 meters.

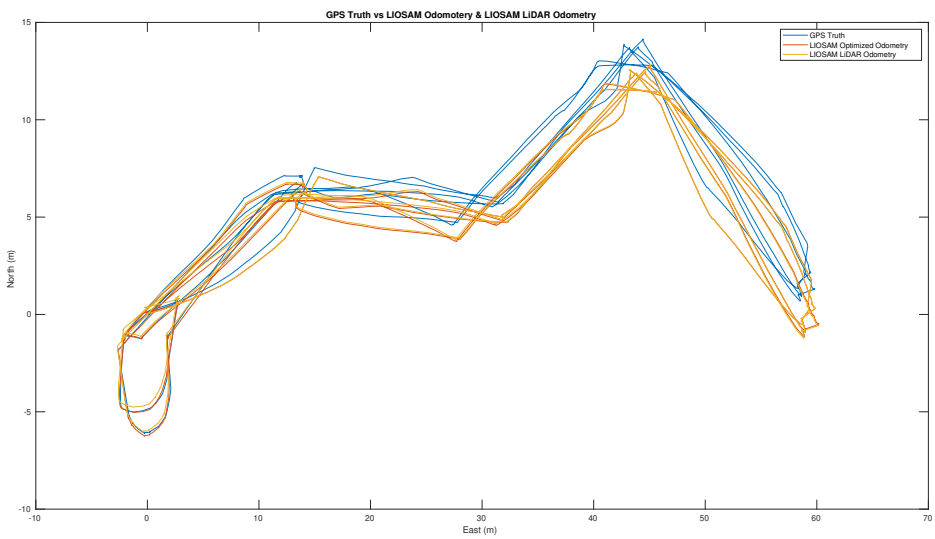


Figure 5.1.9: LIOSAM Odometry vs LIOSAM LiDAR Odometry vs GPS Truth between ZigZag

Table 5.1.3 compares the LIO-SAM [118] optimized pose estimates against the uncorrected LiDAR odometry which drifts in comparison to the optimized estimates and shows the same trend as the other tests.

Although Rhino successfully completed other missions, during the final zigzag test, the researchers realized the robot began having issues that were not seen in the prior tests. The localization, map-

Table 5.1.3: RMSE in East-axis and North-axis for Autonomous Navigation for zigzag test

Reference System	*LIOSAM Optimized*	LIOSAM LiDAR Odometry
East (meters)	**0.365**	0.358
North (meters)	**1.061**	1.0058

ping and autonomous navigation modules' performances had not diverged but the hardware was facing challenges traversing to the waypoints. There was also a perceivable smell of smoke coming from the robot, which made the researchers concerned about the state of the robot. After taking the robot back to the lab, for examinations the researchers noticed that although all actuators were hotter than usual the back right motor was considerably hotter than the others. Although the actual cause of the behavior is yet to be determined, the zigzag tests were the toughest challenge the system had undergone during autonomous missions and Rhino was at its heaviest configuration by carrying backup batteries.

Fig. 5.1.10 shows the current draw from the right side of the robot when given the same motor command. The graph further confirms the damage caused to the motor where there are significant random current spikes. As a result of the damage, the motor was replaced to allow for continued testing.

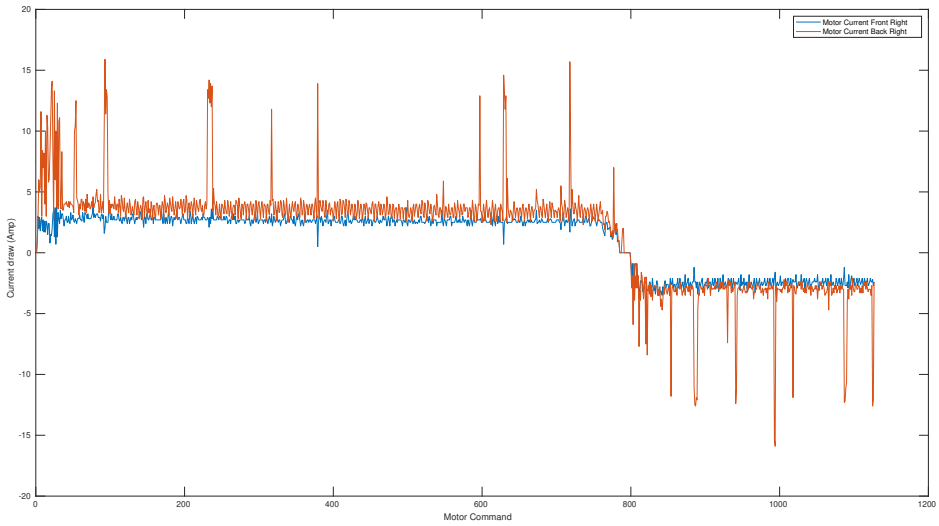


Figure 5.1.10: Current draw of Motor Front Right and Back Right

Fig. 5.1.11 shows the temperatures of the front right motor and the damaged back right motors.

(a) Temperature of Back Right Motor

(b) Temperature of Front Right

Figure 5.1.11: Temperatures of the right side of Rhino's motors

By extensively testing Rhino on the West Virginia University campus, it allowed the researchers to evaluate the performance and limitations of the system. Due to the limited accessibility of the mine, these tests allowed the researchers to address issues and fine-tune the system prior to visiting the mines. After conducting several tests, Rhino was deployed in a coal mine to perform localization, mapping and autonomous navigation missions. The insights gained from conducting missions in the mine will be further discussed in the upcoming section.

5.2 Experiments in Coal Mine Environment

To access the capabilities in GPS-denied and subterranean environments, Rhino was deployed in the NIOSH Safety Research Coal mine as seen in Fig. 5.2.1 and Fig. 5.2.2. Two experiments were carried out to demonstrate and verify the robustness of the software and hardware of Rhino in the

mine. The initial tests carried out in the mine allowed the researchers to evaluate the capabilities and robustness of the system to handle the challenging terrain via teleoperation, remote operations via the communications setup, and localization and mapping module. After verifying the robustness of the mechanical systems, localization and mapping modules, and the autonomous navigation module was assessed.

In an effort to demonstrate the capabilities of the system, Rhino was teleoperated to verify its ability to handle subterranean surfaces. The system successfully traversed a variety of terrain in the mine which were gravel, dry, and muddy terrain. While conducting these tests, the researchers demonstrated the capacity for split body chassis to maintain a stable platform for the system to traverse over some obstacles.

Fig. 5.2.1 and Fig. 5.2.2 show Rhino operating in the NIOSH Safety Research Coal Mine, Pittsburgh, Pennsylvania.

(a) Rhino in front of NIOSH Safety Research Coal Mine

(b) Rhino in the coal mine on loose soil terrain

Figure 5.2.1: Rhino operating in NIOSH Safety Research Coal Mine (**video**)

(a) Rhino operating in NIOSH Safety Research Coal Mine with the help of the instrumented lights

(b) Rhino in the coal mine on muddy terrain

Figure 5.2.2: Rhino operating in NIOSH Safety Research Coal Mine (**video**)

Although Rhino was able to traverse the mine terrain there was some damage caused to the frame, due to a defect in the chassis. The frame that allows the motors to be mounted to the chassis received a considerable amount of stress which led to the welds of the frame being broken. Fig. 5.2.3 shows the damage to the frame with the broken welds shown.

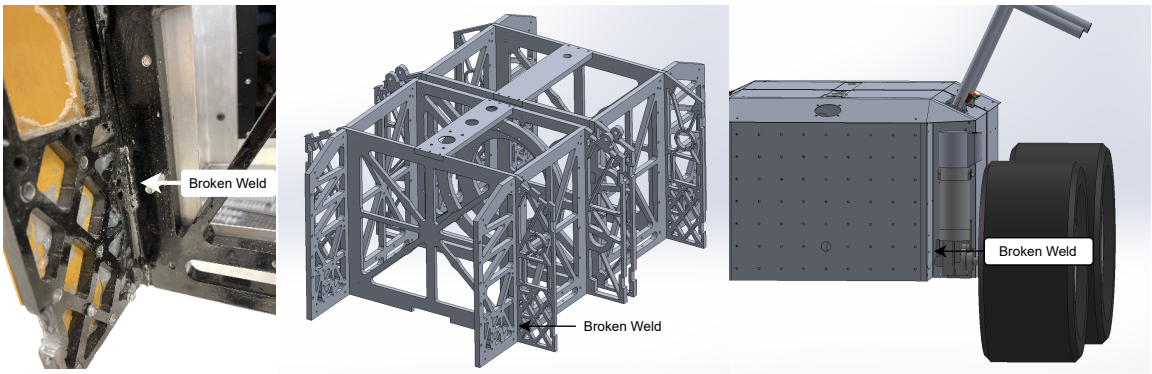


Figure 5.2.3: Broken Chassis Frame

To strengthen and reinforce the chassis, Rhino had to be completely disassembled to allow the welds to be fixed. The damage sustained was fixed by reinforcing the frame with stronger welds and brackets that allow the stress induced by the wheels and motors to be distributed across the frame as shown in Fig. 5.2.4. The process of reconstructing and rebuilding Rhino can be seen **here**.

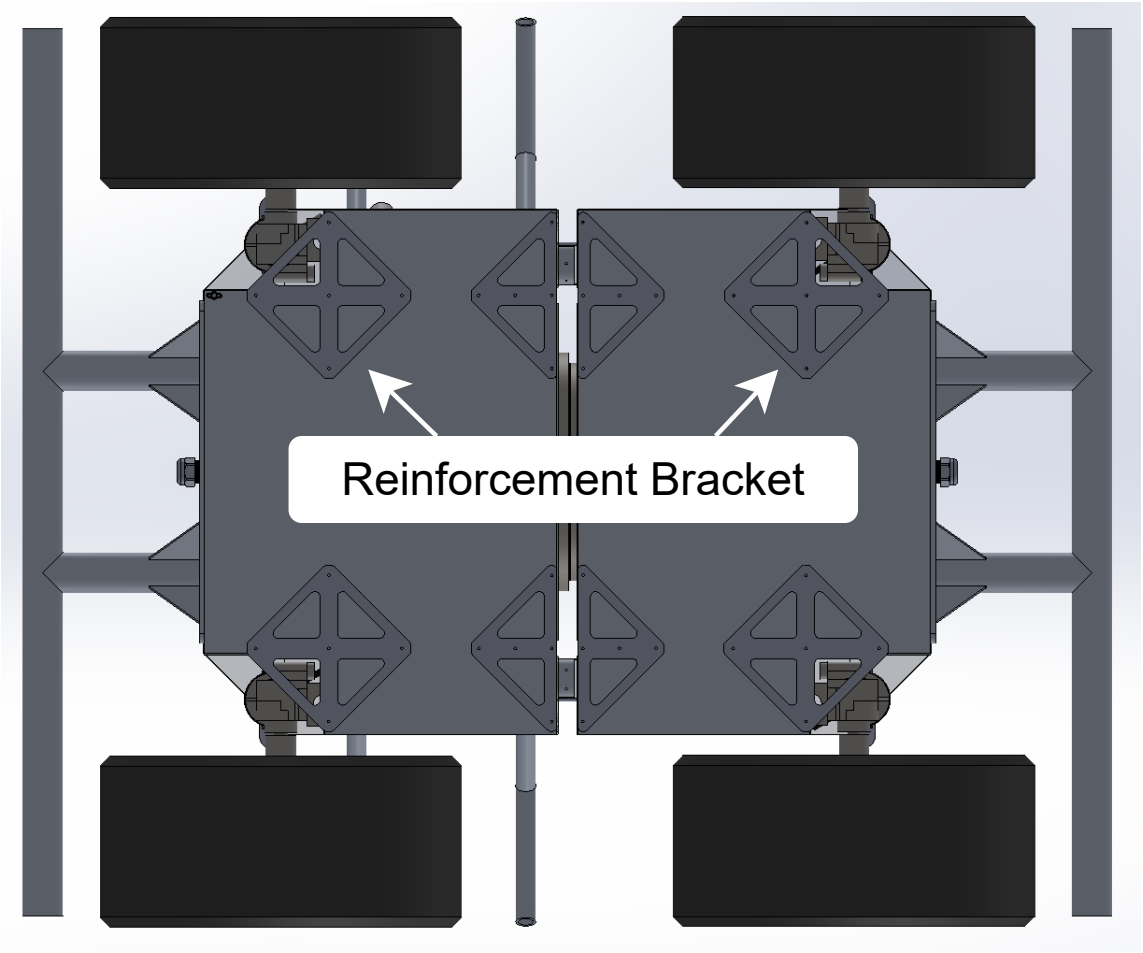


Figure 5.2.4: Reinforcement Brackets

Although Rhino was fixed and reinforced, the damage sustained meant one of the wheels was no longer perfectly aligned with the forward direction of the robot. This results in additional friction while driving which requires more torque for that wheel.

To test the capabilities of the communication setup, a ground station with a base station computer and 2.4 GHz 90° directional antenna was set up. The goal of the communications test was to verify the capabilities of the communications (comms) setup instrumented on Rhino. The test was set up to allow operators to teleoperate Rhino from a base station through the user interface which transmits camera feedback, orientation, current draw from actuators, the status of the brakes, and

lights. The operators were able to drive line-of-sight over the communications setup by using the camera feed and controller. Although the operators could drive Rhino and monitor the generation of the 3D map, some limitations were discovered. The limitations were a result of the harsh mine environment attenuating the radio signals. This was discovered when the robot was driven out-of-line-sight of the base station directional antenna, the operators would instantly lose connection to the system.

In order to generate the high-fidelity maps, Rhino's localization and mapping module was used to generate a 3D map and traversed 1,018 meters as seen in Fig. 5.2.5. During the mapping missions, the loop closure detection was qualitatively evaluated by driving in several loops to assess the system's ability to recognize loop closures. For autonomous missions, waypoints were selected by inputting the robot's pose as reported by the LIOSAM's [118] state estimates and used as inputs for the FALCO planner [145]. Prior to conducting the missions, the planner was tuned for the mine terrain, some of the parameters tuned were the autonomous navigation speed, acceleration, and maximum yaw rate. Some of the initial waypoints selected were for Rhino to autonomously drive to multiple waypoints in a straight line. Rhino was able to successfully traverse to the selected waypoints, after which the researchers decided to select more challenging waypoints. The team decided to select waypoints that allowed Rhino to autonomously navigate to waypoints that form a square pattern. Although the system was able to create high-fidelity maps, there were some missions where the localization estimates were observed to be inaccurate as a result of Rhino traversing to an inaccurate waypoint. The researchers noticed that the system would autonomously navigate to incorrect waypoints, and soon discovered that the localization and mapping module had drifted significantly. After analyzing the data collected during these missions, the researchers realized an issue with incorrect timestamps from the LiDAR. The incorrect timestamps would cause the state estimates to drift drastically since the estimates are used in the global optimization step. In an effort to debug the issues with the localization and mapping, the data collected was used to adjust the algorithms to improve the solution. After tuning the parameters, the mission was replayed in a simulation environment and showed improved pose and map estimates as Rhino would traverse to the selected waypoints.

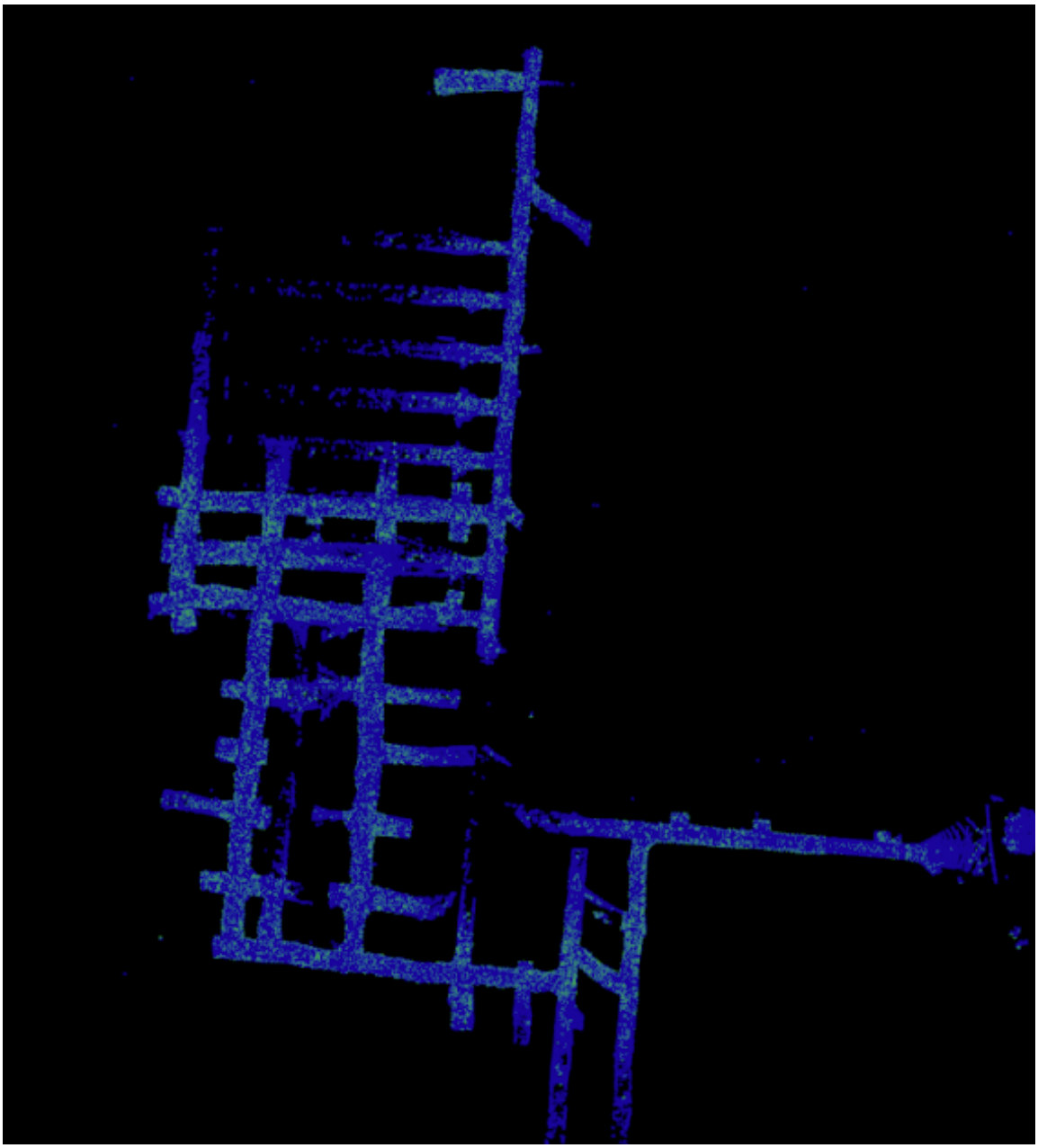


Figure 5.2.5: 3D Generated Map of NIOSH Safety Research Coal Mine by Rhino

The test conducted in the NIOSH Safety Research Mine allowed the researchers to verify Rhino's ability to operate in a challenging GPS-denied environment. The missions conducted showcased the ability of Rhino's hardware to handle the harsh and challenging terrain. By performing experiments in the mine, the researchers were able to verify and improve the performance of Rhino. Although Rhino's chassis' got damaged in one of the experiments, once the issue was addressed and reinforced, the system was able to traverse without issues. By conducting localization, mapping, and autonomous navigation missions, the researchers were able to successfully complete its missions.

6

Conclusion and Future Work

6.1 Conclusion

The goal of the research presented is to develop and implement modern robotic systems to enhance monitoring and warning systems of old workings in underground stone mines. In order to successfully complete autonomous navigation and generate high-fidelity maps, the goal of the project was the development and testing of a robust unmanned ground vehicle in several environments and terrain. During the development of the system, research was conducted to gain insight into some of the state-of-the-art solutions and the gaps in the current systems. In order to successfully complete the autonomous navigation missions, a robust robotic platform was developed to operate in the harsh and challenging stone mine environment.

The research being conducted is to aid in the prevention of accidents, protect human lives, improve and stabilize structures. Underground limestone mines generally have strong structures and are generally stable, and the enhanced pillar designs developed by NIOSH have improved the stability of these mines even more. Although the guidelines provided are being enforced, previously mined sections stay open for years and are uninspected. Due to the outdated safety factors from the old designs and time degradation, over time the pillars can be affected by sloughing as well as reported roof falls. This can pose a threat to the miners, as they have to travel through some mined sections to get to the working face. Due to the severity of the accidents that occur in these mines, inspections are necessary to prevent any collapse, but current methods involve human inspections, which are impractical due to the vast size of some mines as well as the risk to life to conduct the inspections.

In efforts to aid in the prevention of accidents, the development of the system presented in this research had to be tested extensively. In order to successfully operate and conduct missions in the mine, the system developed will have to be robust and reliable. Rhino is an unmanned ground vehicle with a split-body chassis based on the skid-steer driving scheme with the ability to operate for extended periods of time. The robot is instrumented with a variety of sensors and electronics that were selected based on insights gained from research into current systems implemented in mine environments and previous projects. The design and instrumentation of Rhino were to allow for ease of serviceability during field experiments. To complete the autonomous missions in stone mine environments, a state-of-the-art tightly-coupled LiDAR-inertial-odometry via smoothing and mapping (LIO-SAM) SLAM framework [118] is implemented. For autonomous naviga-

tion missions, a terrain analysis module [22] is implemented to allow the system to autonomously access the terrain. To navigate to waypoints and areas of interest during missions the FALCO planner [145] is implemented. The FALCO plans paths by generating offline trajectories according to the constraints and parameters of the unmanned vehicle and formulates the paths as a likelihood problem. The results in this thesis shows the extensive testing and successful implementation of Rhino in these environments. Although there were some issues, by extensively testing and iterating the system, the missions were able to be completed.

The successful implementation of the system will allow mine workers to autonomously determine the structural integrity of the roof and pillars. As the structural integrity of the roofs and pillars are determined, it will allow the miners to rapidly respond to any increasing hazards with proactive measures such as: sending workers to build/rebuild support structure to prevent accidents, warning miners of highly hazardous areas to allow for the use of alternate routes or the evacuation of the mine. Insights gained from the high-fidelity 3D maps can also be used to accurately determine the volumetric change of pillars over time, which can then be used to update strength degradation in pillar models. By updating the models, workers can immediately and proactively respond to possibly catastrophic or fatal pillar and/or roof failures.

6.2 Future Work

Currently, there are several limitations to the system presented in the thesis that needs to be addressed in the future.

Although Rhino can complete the required missions, some changes can be made to improve the performance of the system. Due to the combination of the high weight of the system and the damage sustained in the mine, Rhino's driving capabilities can be further improved by reducing the weight. The researchers plan on upgrading the battery system for the sensors and electronics to four 40V Greenworks batteries and removing four of the 12V batteries. From initial tests conducted on the West Virginia University Campus, the driving capabilities improve when the weight of the system is reduced by removing the backup batteries. By upgrading to the 40V Greenworks batteries, issues discovered from the battery monitoring system of the 12V batteries will also be alleviated.

Research will also be conducted to further improve the performance of the localization and mapping estimates. Although a fiber optic gyroscope is instrumented on the platform, its mea-

surements are only used for the dead reckoning odometry estimates. Since the fiber optic gyroscope measurements have high accuracy, the development of a sensor fusion algorithm that receives inputs from the fiber optic gyro and IMU will improve the attitude estimates. In efforts to improve the localization and mapping module, research will be conducted into the addition of a visual-inertial odometry module. By implementing a robust tightly-coupled-LiDAR-visual-inertial odometry, if one of the exteroceptive sensors fails the system can function during missions.

References

[1] Number and percentage of nonfatal lost-time injuries by accident class at underground mining locations, stone operators, 1983 - 2021. URL https://wwwn.cdc.gov/NIOSH-Mining/MMWC/Injuries/Count?StartYear=1983&EndYear=2021&SelectedMineType=1&SelectedCommodity=4.

[2] Darpa subterranean (subt) challenge. URL https://www.darpa.mil/program/darpa-subterranean-challenge.

[3] Robot operating system. URL https://www.ros.org/.

[4] Tvs diodes. URL https://www.littelfuse.com/products/tvs-diodes.aspx.

[5] 2 workers injured in mine accident remain hospitalized. *Pittsburg's Action 4 News*, Apr 2015. URL https://www.wtae.com/article/3-workers-injured-by-ground-shifting-at-pennsylvania-quarry/7471375.

[6] Ferret, 2018. URL https://www.ri.cmu.edu/robot/ferret/.

[7] Mining topic: Ground control overview, Nov 2019. URL https://www.cdc.gov/niosh/mining/topics/groundcontroloverview.html.

[8] Number and percentage of occupational fatalities by accident class at underground mining locations, 1983 - 2021 (n=1179), 2021. URL https://wwwn.cdc.gov/NIOSH-Mining/MMWC/Fatality/Count?StartYear=1983&EndYear=2021&SelectedMineType=1.

[9] Number and percentage of occupational fatalities by accident class at underground mining locations, stone operators, 1983 - 2021 (n=35), 2021. URL https://wwwn.cdc.gov/NIOSH-Mining/MMWC/Fatality/Count?StartYear=1983&EndYear=2021&SelectedMineType=1&SelectedCommodity=4.

[10] Sameer Agarwal and Keir Mierle. Ceres solver: Tutorial & reference. *Google Inc*, 2(72):8, 2012.

[11] Ali Agha, Kyohei Otsu, Benjamin Morrell, David D Fan, Rohan Thakker, Angel Santamaria-Navarro, Sung-Kyun Kim, Amanda Bouman, Xianmei Lei, Jeffrey Edlund, et al. Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge. *arXiv preprint arXiv:2103.11470*, 2021.

[12] *ADIS16495 Data Sheet*. Analog Devices, 2017. Rev. A.

[13] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J Pappas. Decentralized active information acquisition: Theory and application to multi-robot slam. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4775–4782. IEEE, 2015.

[14] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.

[15] Bradley M Bell and Frederick W Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.

[16] Ben Bellekens, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn. A survey of rigid 3d pointcloud registration algorithms. In *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pages 8–13, 2014.

[17] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[18] Nitin Bhatia et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

[19] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.

[20] Russell Buchanan, Lorenz Wellhausen, Marko Bjelonic, Tirthankar Bandyopadhyay, Navinda Kottege, and Marco Hutter. Perceptive whole-body planning for multilegged robots in confined spaces. *Journal of Field Robotics*, 38(1):68–84, 2021.

[21] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[22] Chao Cao, Hongbiao Zhu, Fan Yang, Yukun Xia, Howie Choset, Jean Oh, and Ji Zhang. Autonomous exploration development environment and the planning algorithms. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8921–8928. IEEE, 2022.

[23] Luca Carlone, Rosario Aragues, José A Castellanos, and Basilio Bona. A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, 33(7):965–987, 2014.

[24] Luca Carlone, Andrea Censi, and Frank Dellaert. Selecting good measurements via ℓ 1 relaxation: A convex approach for robust estimation over graphs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2667–2674. IEEE, 2014.

[25] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d slam: a survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4597–4604. IEEE, 2015.

[26] Luca Carlone, Giuseppe C Calafiore, Carlo Tommolillo, and Frank Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Transactions on Robotics*, 32(3):545–565, 2016.

[27] Lee Champeny-Bares, Syd Coppersmith, and Kevin Dowling. The terregator mobile robot. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1991.

[28] Frank E Chase and Christopher Mark. Pillar design and strategies for retreat mining. *Improving Safety at Small Underground Mines: Proceedings*, 18:15, 1994.

[29] Cheng Chi, Xuguang Sun, Ning Xue, Tong Li, and Chang Liu. Recent progress in technologies for tactile sensors. *Sensors*, 18(4):948, 2018.

[30] Kok Seng Chong and Lindsay Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 2783–2788. IEEE, 1997.

[31] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.

[32] Dylan Michael Covell. Top-down & bottom-up approaches to robot design. Master's thesis, West Virginia University, Morgantown, WV, 2022.

[33] Ítalo Renan da Costa Barros and Tiago Pereira Nascimento. Robotic mobile fulfillment systems: A survey on recent developments and research opportunities. *Robotics and Autonomous Systems*, 137:103729, 2021.

[34] Shounak Das, Cagri Kilic, Ryan Watson, and Jason Gross. A comparison of robust kalman filters for improving wheel-inertial odometry in planetary rovers. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, pages 2621–2632, 2021.

[35] Ken Davidian. Prize competitions and nasa's centennial challenges program. In *International Lunar Conference*, pages 09–08, 2005.

[36] Roozbeh Dehghannasiri, Mohammad Shahrokh Esfahani, Xiaoning Qian, and Edward R Dougherty. Bayesian kalman filtering in the presence of unknown noise statistics using factor graphs. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 166–170. IEEE, 2017.

[37] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[38] Frank Dellaert and Michael Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[39] Konstantinos G Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 4(1): 2–3, 2010.

[40] Renaud Dubé, Abel Gawel, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena. An online multi-robot slam system for 3d lidars. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1004–1011. IEEE, 2017.

[41] Kamak Ebadi, Yun Chang, Matteo Palieri, Alex Stephens, Alex Hatteland, Eric Heiden, Abhishek Thakur, Nobuhiro Funabiki, Benjamin Morrell, Sally Wood, et al. Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 80–86. IEEE, 2020.

[42] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. Present and future of slam in extreme underground environments. *arXiv preprint arXiv:2208.01787*, 2022.

[43] Garry A Einicke and Langford B White. Robust extended kalman filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599, 1999.

[44] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on instrumentation and measurement*, 57(1):140–149, 2007.

[45] Mahdi Elhousni and Xinming Huang. A survey on 3d lidar localization for autonomous vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1879–1884. IEEE, 2020.

[46] Gabriel S Esterhuizen, Paul L Tyrna, and Michael M Murphy. A case study of the collapse of slender pillars affected by through-going discontinuities at a limestone mine in pennsylvania. *Rock Mechanics and Rock Engineering*, 52(12):4941–4952, 2019.

[47] John Folkesson and Henrik I Christensen. Graphical slam for outdoor applications. *Journal of Field Robotics*, 24(1-2):51–70, 2007.

[48] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1): 1–21, 2016.

[49] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.

[50] Pablo Gonzalez-De-Santos, Roemi Fernández, Delia Sepúlveda, Eduardo Navas, and Manuel Armada. Unmanned ground vehicles for smart farms. In *Agronomy-Climate Change and Food Security*, page 73. 2020.

[51] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[52] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007.

[53] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems*, volume 3, page 9, 2007.

[54] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[55] Yu Gu. Capstone project – robotics section. URL https://researchrepository.wvu.edu/wvu_syllabi/17147.

[56] Scott Edward Harper. Instrumentation and inertial navigation systems design for tensegrity robot implementations. Master's thesis, West Virginia University, Morgantown, WV, 2018.

[57] Ross Hartley, Maani Ghaffari Jadidi, Lu Gan, Jiunn-Kai Huang, Jessy W Grizzle, and Ryan M Eustice. Hybrid contact preintegration for visual-inertial-contact state estimation using factor graphs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3783–3790. IEEE, 2018.

[58] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

[59] Kin Leong Ho and Paul Newman. Loop closure detection in slam by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

[60] Dirk Holz and Sven Behnke. Mapping with micro aerial vehicles by registration of sparse 3d laser scans. In *Intelligent Autonomous Systems 13*, pages 1583–1599. Springer, 2016.

[61] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. An observability-constrained sliding window filter for slam. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 65–72. IEEE, 2011.

[62] Nicolas Hudson, Fletcher Talbot, Mark Cox, Jason Williams, Thomas Hines, Alex Pitt, Brett Wood, Dennis Frousheger, Katrina Lo Surdo, Thomas Molnar, et al. Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge. *arXiv preprint arXiv:2104.09053*, 2021.

[63] Jirı Iša and Martin Dlouhỳ. Robotour-robotika. cz outdoor delivery challenge. In *Proceedings of the 1-st Slovak-Austrian International Conference on Robotics in Education, Bratislava, Slovakia.(To appear)*. Citeseer, 2010.

[64] Tariqul Islam, Md Saiful Islam, Md Shajid-Ul-Mahmud, and Md Hossam-E-Haider. Comparison of complementary and kalman filter based data fusion for attitude heading reference system. In *AIP Conference Proceedings*, volume 1919, page 020002. AIP Publishing LLC, 2017.

[65] Fabian Jenelten, Takahiro Miki, Aravind E Vijayan, Marko Bjelonic, and Marco Hutter. Perceptive locomotion in rough terrain–online foothold optimization. *IEEE Robotics and Automation Letters*, 5(4):5370–5376, 2020.

[66] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[67] Simon J Julier and Jeffrey K Uhlmann. Using covariance intersection for slam. *Robotics and Autonomous Systems*, 55(1):3–20, 2007.

[68] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Fast incremental smoothing and mapping with efficient data association. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 1670–1677. IEEE, 2007.

[69] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[70] Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960.

[71] Gavin Kenneally, Avik De, and Daniel E Koditschek. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, 2016.

[72] Alif Ridzuan Khairuddin, Mohamad Shukor Talib, and Habibollah Haron. Review on simultaneous localization and mapping (slam). In *2015 IEEE international conference on control system, computing and engineering (ICCSCE)*, pages 85–90. IEEE, 2015.

[73] Shehryar Khattak, Huan Nguyen, Frank Mascarich, Tung Dang, and Kostas Alexis. Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1024–1029. IEEE, 2020.

[74] Cagri Kilic, Jason N. Gross, Nicholas Ohi, Ryan Watson, Jared Strader, Thomas Swiger, Scott Harper, and Yu Gu. Improved planetary rover inertial navigation and wheel odometry performance through periodic use of zero-type constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 552–559, 2019. doi: 10.1109/IROS40897.2019.8967634.

[75] Cagri Kilic, Nicholas Ohi, Yu Gu, and Jason N. Gross. Slip-based autonomous zupt through gaussian process to improve planetary rover localization. *IEEE Robotics and Automation Letters*, 6(3):4782–4789, 2021. doi: 10.1109/LRA.2021.3068893.

[76] Dimitrios G Kottas, Joel A Hesch, Sean L Bowman, and Stergios I Roumeliotis. On the consistency of vision-aided inertial navigation. In *Experimental Robotics*, pages 303–317. Springer, 2013.

[77] Tomáš Krajnık, Jan Faigl, Vojtech Vonásek, Hana Szücsová, Ondrej Fišer, and Libor Preucil. A visual navigation system for robotour competition. In *First International Conference on Robotics in Education, Bratislava*, volume 1, pages 95–100, 2010.

[78] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[79] *DSP-1760 Multi-axis Gyroscope Technical Manual*. KVH, 2019. Rev. B.

[80] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph slam. *The International Journal of Robotics Research*, 32(14):1611–1626, 2013.

[81] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[82] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[83] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[84] Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, 72(2):147–165, 2013.

[85] Thomas Lowe, Soohwan Kim, and Mark Cox. Complementary perception for handheld slam. *IEEE Robotics and Automation Letters*, 3(2):1104–1111, 2018.

[86] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2015.

[87] Martin Magnusson. *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection*. PhD thesis, Örebro universitet, 2009.

[88] C Mark and GM Molinda. The coal mine roof rating (cmrr)—a decade of experience. *International Journal of Coal Geology*, 64(1-2):85–103, 2005.

[89] Christopher Mark. *Analysis of longwall pillar stability*. The Pennsylvania State University, 1987.

[90] Christopher Mark, Frank E Chase, and Deno M Pappas. Analysis of multiple seam stability. 2007.

[91] U. Martinez-Hernandez. Tactile Sensors. *Scholarpedia*, 10(4):32398, 2015. doi: 10.4249/scholarpedia.32398. revision #150526.

[92] Wang Ming-Xiao, Zhang Tao, Xie Miao-Rong, Zhang Bin, and Jia Ming-Qiu. Analysis of national coal-mining accident data in china, 2001–2008. *Public Health Reports*, 126(2):270, 2011.

[93] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.

[94] Aaron Morris, Dave Ferguson, Zachary Omohundro, David Bradley, David Silver, Chris Baker, Scott Thayer, Chuck Whittaker, and William Whittaker. Recent developments in subterranean robotics. *Journal of Field Robotics*, 23(1):35–57, 2006.

[95] Anastasios I Mourikis, Stergios I Roumeliotis, et al. A multi-state constraint kalman filter for vision-aided inertial navigation. In *ICRA*, volume 2, page 6, 2007.

[96] Sivadev Nadarajah and Kenneth Sundaraj. A survey on team strategies in robot soccer: team strategies and role description. *Artificial Intelligence Review*, 40(3):271–304, 2013.

[97] Andress Nuchter, Hartmut Surmann, Kai Lingemann, Joachim Hertzberg, and Sebastian Thrun. 6d slam with an application in autonomous mine mapping. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pages 1998–2003. IEEE, 2004.

[98] David Obdrzalek. Eurobot junior and starter-a comparison of two approaches for robotic contest organization. *Robotics in Education, Bratislava*, 2010.

[99] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013.

[100] Koichi Osuka, Robin Murphy, and Alan C Schultz. Usar competitions for physically situated robots. *IEEE Robotics & Automation Magazine*, 9(3):26–33, 2002.

[101] *Ouster OS1 Datasheet*. Ouster, Inc, 2022. REV06.

[102] Lukas Polok, Viorela Ila, Marek Solony, Pavel Smrz, and Pavel Zemcik. Incremental block cholesky factorization for nonlinear least squares in robotics. In *Robotics: Science and Systems*, pages 328–336, 2013.

[103] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015.

[104] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[105] Milad Ramezani, Kasra Khosoussi, Gavin Catt, Peyman Moghadam, Jason Williams, Paulo Borges, Fred Pauling, and Navinda Kottege. Wildcat: Online continuous-time 3d lidar-inertial slam. *arXiv preprint arXiv:2205.12595*, 2022.

[106] Dip N Ray, R Dalui, A Maity, and S Majumder. Sub-terranean robot: A challenge for the indian coal mines. *The Online Journal on Electronics and Electrical Engineering*, 2(2):217–222, 2009.

[107] A Hemanth Reddy, Balla Kalyan, and Ch SN Murthy. Mine rescue robot system–a review. *Procedia Earth and Planetary Science*, 11:457–462, 2015.

[108] Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.

[109] Thomas Rooney, Anthony G Pipe, Sanja Dogramadzi, and Martin J Pearson. Towards tactile sensing applied to underwater autonomous vehicles for near shore survey and de-mining. In *Conference Towards Autonomous Robotic Systems*, pages 463–464. Springer, 2012.

[110] Tomáš Rouček, Martin Pecka, Petr Čížek, Tomáš Petříček, Jan Bayer, Vojtěch Šalanskỳ, Teymur Azayev, Daniel Heřt, Matěj Petrlík, Tomáš Báča, et al. System for multi-robotic exploration of underground environments ctu-cras-norlab in the darpa subterranean challenge. *arXiv preprint arXiv:2110.05911*, 2021.

[111] Julian Ryde and Nick Hillier. Performance of laser and radar ranging devices in adverse environmental conditions. *Journal of Field Robotics*, 26(9):712–727, 2009.

[112] A.M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering*, 53(7):1346–1356, 2006. doi: 10.1109/TBME.2006.875664.

[113] Joseph H Saleh and Amy M Cummings. Safety in the mining industry and the unfinished legacy of mining accidents: Safety levers and defense-in-depth for addressing mining hazards. *Safety science*, 49(6):764–777, 2011.

[114] Martin J Schuster, Sebastian G Brunner, Kristin Bussmann, Stefan Büttner, Andreas Dömel, Matthias Hellerer, Hannah Lehner, Peter Lehner, Oliver Porges, Josef Reill, et al. Towards autonomous planetary exploration. *Journal of Intelligent & Robotic Systems*, 93(3):461–494, 2019.

[115] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[116] Mo Shan, Yingcai Bi, Hailong Qin, Jiaxin Li, Zhi Gao, Feng Lin, and Ben M Chen. A brief survey of visual odometry for micro aerial vehicles. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6049–6054. IEEE, 2016.

[117] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[118] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

[119] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 5692–5698. IEEE, 2021.

[120] Young-Sik Shin and Ayoung Kim. Sparse depth enhanced direct thermal-infrared slam beyond the visible spectrum. *IEEE Robotics and Automation Letters*, 4(3):2918–2925, 2019.

[121] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

[122] Gabe Sibley. Sliding window filters for slam. *University of Southern California, Tech. Rep*, 2006.

[123] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

[124] David Silver, Dave Ferguson, Aaron Morris, and Scott Thayer. Feature extraction for topological mine maps. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 773–779. IEEE, 2004.

[125] Dustin Spayde. The lane innovation hub. URL https://innovationhub.wvu.edu/the-hub.

[126] Yunjiang Sun, Jianping Zuo, Murat Karakus, and Jintao Wang. Investigation of movement and damage of integral overburden during shallow coal seam mining. *International Journal of Rock Mechanics and Mining Sciences*, 117:63–75, 2019.

[127] Niko Sünderhauf and Peter Protzel. Towards a robust back-end for pose graph slam. In *2012 IEEE international conference on robotics and automation*, pages 1254–1261. IEEE, 2012.

[128] Richard Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007.

[129] Tomoji Takasu and Akio Yasuda. Development of the low-cost rtk-gps receiver with an open source program package rtklib. In *International symposium on GPS/GNSS*, volume 1. International Convention Center Jeju Korea, 2009.

[130] Sebastian Thrun, Dirk Hahnel, David Ferguson, Michael Montemerlo, Rudolph Triebel, Wolfram Burgard, Christopher Baker, Zachary Omohundro, Scott Thayer, and William Whittaker. A system for volumetric robotic mapping of abandoned mines. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 3, pages 4270–4275. IEEE, 2003.

[131] Gian Diego Tipaldi and Kai O Arras. Flirt-interest regions for 2d range data. In *2010 IEEE International Conference on Robotics and Automation*, pages 3616–3622. IEEE, 2010.

[132] Arthur W Toga and Paul M Thompson. The role of image registration in brain mapping. *Image and vision computing*, 19(1-2):3–24, 2001.

[133] Marco Tranzatto, Takahiro Miki, Mihir Dharmadhikari, Lukas Bernreiter, Mihir Kulkarni, Frank Mascarich, Olov Andersson, Shehryar Khattak, Marco Hutter, Roland Siegwart, et al. Cerberus in the darpa subterranean challenge. *Science Robotics*, 7(66):eabp9742, 2022.

[134] Ihsan Berk Tulu, Jason N. Gross, Yu Gu, and Guilherme A. S. Pereira. Autonomous robotic early warning system for underground stone mining safety, Aug 2019. URL https://www.alpha-foundation.org/afc820-69_wvu_synopsis/.

[135] Roberto G Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. *Sensors*, 15(8):19302–19330, 2015.

[136] Liang Wang, Yihuan Zhang, and Jun Wang. Map-based localization method for autonomous vehicles using 3d-lidar. *IFAC-PapersOnLine*, 50(1):276–281, 2017.

[137] Gary L Winn, Robert Rice, and James Dean. After quecreek: A review of projects to prevent coal mine inundations in west virginia. *Professional Safety*, 51(3):35, 2006.

[138] Oliver J Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007.

[139] WU Xiwei, XIAO Bing, WU Cihang, GUO Yiming, and LI Lingwei. Factor graph based navigation and positioning for control system design: A review. *Chinese Journal of Aeronautics*, 2021.

[140] Chizhao Yang. *Localization Algorithms for GNSS-denied and Challenging Environments*. PhD thesis, West Virginia University, 2021.

[141] Tae Suk Yoo, Sung Kyung Hong, Hyok Min Yoon, and Sungsu Park. Gain-scheduled complementary filter design for a mems based attitude and heading reference system. *Sensors*, 11(4):3816–3830, 2011.

[142] Shenghai Yuan, Han Wang, and Lihua Xie. Survey on localization systems and algorithms for unmanned systems. *Unmanned Systems*, 9(02):129–163, 2021.

[143] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[144] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401–416, 2017.

[145] Ji Zhang, Chen Hu, Rushat Gupta Chadha, and Sanjiv Singh. Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation. *Journal of Field Robotics*, 37(8):1300–1313, 2020.

[146] Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE, 2021.

[147] RK Zipf Jr and C Mark. Ground control for highwall mining in the united states. *International Journal of Surface Mining, Reclamation and Environment*, 19(3):188–217, 2005.

[148] Xingxing Zuo, Patrick Geneva, Woosik Lee, Yong Liu, and Guoquan Huang. Lic-fusion: Lidar-inertial-camera odometry. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5848–5854. IEEE, 2019.