DECENTRALIZED COLLABORATIVE LOCALIZATION USING ULTRA-WIDEBAND RANGING

A Thesis

by

JACOB MARSHALL HARTZER

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Srikanth Saripalli
Co-Chair of Committee	Douglas Allaire
Committee Member,	Manoranjan Majji
Department Head,	Andreas A. Polycarpou

December 2020

Major Subject: Mechanical Engineering

ABSTRACT

This thesis summarizes the development of a collaborative localization algorithm simulation environment and the implementation of collaborative localization using Ultra-Wideband ranging in autonomous vehicles. In the developed simulation environment, multi-vehicle scenarios are testable with various sensor combinations and configurations. The simulation emulates the networking required for collaborative localization and serves as a platform for evaluating algorithm performance using Monte Carlo analysis. Monte-Carlo simulations were run using a number of situations and vehicles to test the efficacy of UWB sensors in decentralized collaborative localization as well as landmark measurements within an Extended Kalman Filter. Improvements from adding Ultra-Wideband ranging were shown in all simulated environments, with landmarks offering additional improvements to collaborative localization, and with the most significant accuracy improvements seen in GNSS-denied environments. Physical experiments were run using a by-wire GEM e6 from Autonomous Stuff in an urban environment in both collaborative and landmark setups. Due to higher than expected INS certainty, adding UWB measurements showed smaller improvements than simulations. Improvements of 9.2 to 12.1% were shown through the introduction of Ultra-Wideband ranging measurements in a decentralized collaborative localization algorithm. Improvements of 30.6 to 83.3% were shown in using UWB ranging measurements to landmarks in an extended kalman filter for street crossing and tunnel environments respectively. These results are similar to the simulated data, and are promising in showing the efficacy of adding UWB ranging sensors to cars for collaborative and landmark localization, especially in GNSS-denied environments. In the future, additional moving vehicles with additional tags will be tested and further evaluations of the UWB ranging modules will be performed.

DEDICATION

To my parents who taught me how to challenge my self and explore the world around me. And to my love, Heather, for her undying support and inspiration through these years.

ACKNOWLEDGMENTS

I would like to thank the North Texas Tollway Authority for their support of my graduate education and research. I would like to especially thank Dr. Srikanth Saripalli for his guidance and creativity that inspired the work I have performed.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

The serial driver used in the UBX ROS driver *ubxtranslator* was provided by GitHub user dalymople and expanded to interface with ROS and additional u-Blox modules.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

My Graduate Study has been supported by a fellowship from Texas A&M University and from the Unmanned Systems Lab under the direction of Dr. Srikanth Saripalli.

NOMENCLATURE

EKF	Extended Kalman Filter
EKF LMK	Extended Kalman Filter with Landmark Ranging
CL	Collaborative Localization
DCL	Decentralized Collaborative Localization
DCL LMK	Decentralized Collaborative Localization with Landmark Ranging
IMU	Inertial Measurement Unit
IPS	Indoor Positioning System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
MLE	Maximum Likelihood Estimation
MC	Monte Carlo
NLOS	Non-Line-of-Sight
ROS	Robot Operating System
RTK	Real-Time Kinematic
TOF	Time of Flight
UAV	Unmanned Aerial Vehicle
UWB	Ultra-Wideband
VANet	Vehicular *Ad hoc* Network
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2X	Vehicle-to-Everything

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Background

There are numerous industries that are in the process of developing autonomous passenger vehicles. Such vehicles have the capacity to reduce risk and increase free time of riders. In the development of autonomous vehicles, a key consideration is reliable localization in order to plan for and react to the environment. Typically, high accuracy positioning is achieved through the use of Global Navigation Satellite Systems (GNSS) such as the US based GPS or the European GALILEO. The use of GNSS with an Inertial Measurement Unit (IMU) or odometry measurements can lead to a reliable and accurate positioning estimate that can be used to control vehicles and robots [1].

However, reliable GNSS measurements are often not available. Because GNSS receivers must have line of sight with at least four GNSS satellites to fix time and location, environments that include obstructions pose significant challenges. Urban canyons, tunnels, or indoor environments can degrade or fully obstruct GNSS signals, forcing the system to rely on inertial and odometry measurements [2]. These measurements are subject to drift, and therefore cannot be trusted after continued interruption of GNSS measurements. As such, it would be advantageous to the continued development of autonomous vehicles to use a exteroceptive sensor for navigation through environments where GNSS is typically denied.

Ultra-Wideband (UWB), defined as radio technology exceeding 500 MHz or 20% of the arithmetic center frequency [3], utilizes low-energy pulse communication typically for short-range, high-bandwidth applications. By measuring time of flight (ToF) across various frequencies, it is possible to measure distance between modules while overcoming issues typically associated with multipath errors [4]. This has allowed UWB modules to be applied towards localization and tracking problems [5]. UWB ranging measurements can be generated in both indoor and outdoor environments, showing their promise in transitional areas and especially in GNSS-denied environ-

ments. With the continued development of UWB technology and roll out in commercial products, it's ability to aid in navigation should continue to be explored.

Additionally, by mounting UWB ToF sensors on autonomous vehicles, it is possible to generate ranging estimates to other independent vehicles. Just like measurements to landmarks, these measurements are not subject to drift when GNSS measurements are obscured, and can be made available in both indoor and outdoor environments. By producing a reliable ranging estimate between vehicles, localization estimates can be generated using a network approach. Known as collaborative localization, these measurements can leverage the growing number of intelligent vehicles on the road, and can increase the effective area of vehicles that are currently limited to GNSS-available environments.

1.2 Purpose and Aims

It is the goal of this thesis work to explore the application of UWB ranging technology in the collaborative localization of autonomous vehicles.

By utilizing the Decawave EVK1000 modules [6], it will be possible to explore the error distribution of this sensor in various configurations between vehicles and landmarks. This will allow for the creation of a sensor model to validate its use in Kalman filter update equations. Further, to better incorporate these measurements in real time, necessary drivers will be created to interface between the UWB sensor modules and the Robot Operating System (ROS) middleware.

In order quickly to test various algorithms and configurations, a simulation environment will be created. This environment will simulate kinematic models of vehicles as well as contain sensor models for IMU, odometry, GNSS, and UWB sensors. This environment will be used to evaluate the efficacy of adding UWB sensors, as well as explore different estimation algorithms.

Because UWB ranging measurements also offer relative measurements between vehicles, this work will explore the application of decentralized collaborative localization across a network of vehicles. With the prospect of using both landmark measurements and relative vehicle to vehicle (V2V) measurements, this work will evaluate the benefit of using UWB ranging on autonomous vehicles.

1.3 Contributions

This work has produced a number of contributions, including a collaborative localization simulation, sensing models for GNSS and UWB modules, Monte Carlo simulations of collaborative and landmark sensing environments, and experimental results of collaborative and landmark sensing environments.

Through the developed MatLab simulation environment, multi-vehicle scenarios are testable with various sensor combinations and different configurations. These scenarios provide repeatable and realistic environments to test various algorithms for collaborative localization and filtering. The simulation also serves as a platform for evaluating algorithm performance using Monte Carlo analysis. Additionally, animations of the environments, vehicles, and estimates are producible in real time to visually aid evaluation.

Sensing models were created for both the GNSS and UWB modules. The GNSS sensor model accepts CEP values and transforms the value using a circularly symmetric Rayleigh distribution. The UWB ranging residuals were shown to follow a normal distribution using a normal probability plot and a Jarque-Bera test.

Using Monte Carlo simulations, improvements were shown in all simulated environments, with landmarks offering more improvement to collaborative localization, and with the most significant accuracy improvements seen in GNSS-denied environments.

ROS driver packages to connect RTK GNSS modules from u-blox and UWB ranging modules from Decawave were developed. These packages wrapped serial communication protocols for each module with a ROS package that include node, drivers, message files, and launch files necessary to access real-time data.

To validate the simulation results, physical experiments were run using a by-wire GEM e6 from Autonomous Stuff in an urban environment in both collaborative and landmark setups. Improvements were shown using UWB in collaborative localization environments as well as using UWB ranging measurements to landmarks in an extended kalman filter.

2. LITERATURE REVIEW

2.1 Indoor Localization using UWB

Ultra-Wideband, with it's previously mentioned benefits, is a very promising technology for robotic localization. There are many examples of using UWB ranging modules for high-accuracy indoor localization of ground or aerial systems. These systems typically measure ranges with respect to fixed landmark anchors that are map specific and do not incorporate mobile anchors [7, 8]. In [9], an Indoor Positioning System (IPS) utilizing a 8-DOF Extended Kalman Filter (EKF) to fuse IMU and UWB measurements is compared to the non-inertial UWB system. A MatLab simulation was created to estimate the benefit of fusing inertial measurements which resulted in more than a 100% improvement of localization accuracy. These results were validated with a 3-DOF test bed, showing similar results. In [10], non-line-of-sight (NLOS) errors and multipath effects were considered for complex indoor environments. By implementing a Sage-Husa Fuzzy Adaptive Filter, outlier measurements can be detected and corrected. Through this work, an IPS utilizing UWB was validated using simulation and experimental results. In [11], an UWB sensor framework is incorporated into a Extended Kalman Filter with a low cost IMU and vision-odometry. The framework was originally tested in SimplySim and validated indoors physically with results showing an accuracy of 10 cm. In [12], multiple omnidirectional robots each with UWB sensors were coordinated along multiple trajectories. Using an Extended Kalman Filter, the UWB range measurements were incorporated and a dynamics compensated control schema with PID control was utilized to coordinate the robots. Similarly, in [13], an Unscented Kalman Filter was utilized for indoor navigation using wheel odometry and UWB measurements. This framework was validated in a real-time control experiment with a mean error of 7 cm.

2.2 Outdoor Localization using UWB

There are also examples of using UWB modules for outdoor vehicular environments. Such a framework has the capacity to increase localization accuracy of existing systems, or provide accu-

rate localization in environments where GPS is not reliable, such as tunnels or urban canyons. In [14], an EKF was utilized to fuse UWB, and IMU data. In order to simulate GNSS deprivation, the on board RTK GPS unit was not used in the filter, but rather as a ground truth comparison. In the outdoor environments that were tested, an accuracy of below 10 cm was achieved, which was an improvement for a purely compromised GNSS system. In [15], a UWB sensing framework is utilized where multiple UWB modules are used on each vehicle, and these measurements are fused with a Kalman filter. Using experimental measurements for a vehicle position, UWB measurements were simulated, producing position error standard deviations of less than 1 meter. In [16], odometry and inertial measurements were fused with UWB using an EKF and a Non-Line-of-Sight removal algorithm to further improve accuracy. Utilizing fixed road landmarks, a real-time positioning system was developed capable of operating with a mean error of less than 18 cm. Finally, in [17], a vehicle to everything (V2X) cooperative position framework was proposed for the inclusion of UWB measurements. A tightly coupled GPS/UWB/INS integrated positioning equation was used in a Robust Kalman Filter. By using a Vehicular *Ad hoc* Network (VANET), vehicles are expected to make use of inter-vehicle distance and angle measurements to aid localization. This integrated system showed improvement over GPS-denied environments and has the capacity for expansion to collaborative environments.

2.3 Collaborative Localization

Given a network of vehicles, it is possible to leverage relative measurements to provide better localization accuracy as a group, than as a collection of individuals, generally referred to as collaborative localization (CL). Early work in CL was done in [18], which described a method of "Cooperative Positioning with Multiple Robots" that showed significant improvement over dead-reckoning. Improvements on their methods have produced a range of methods for using networks to improve state estimation. These methods are generally classified in two groups: centralized methods, and decentralized methods.

2.3.1 Centralized Methods

Centralized CL methods use a single or multiple fusion centers, to which every vehicle communicates measurement information. These state estimators are capable of producing optimal results when estimating linear systems, and have been used in a number of applications with success. It was shown in [19] that a network of four ground robots with relative pose measurements can be numerically optimized using maximum likelihood estimation (MLE) to determine the optimal estimate of relative state for the entire network.

A sequential motion method similar to that simulated in [18], was implemented in [20] using a centralized network of three robots. This work implemented the relative measurements using LED lights and monocular cameras. The system was capable of improving previous camera-based localization.

Additional work in CL using centralized architectures have been done using unmanned aerial vehicles (UAVs). Simultaneous localization and mapping (SLAM) was performed in [21] using a network of two UAVs that communicated monocular vision data to a central server that would manage map fusion, optimization, and distribution back to the vehicles.

Issues with centralized networks include sensitivity to failure or disconnection, as well as bandwidth and computational power. As the number of measurements made can increase on the order of $\mathcal{O}(n^2)$, centralized networks can meet constraints when large networks of nodes are implemented with complex measurements or update functions. Methods to mitigate this issue are discussed in [22].

2.3.2 Decentralized Methods

Decentralized CL methods distribute the state estimation computation across numerous or every agent in the network. Unlike centralized methods, decentralized CL methods are not susceptible to single point failures.

It was shown in [23] that the equations of a centralized Kalman estimator can be rewritten in the form of smaller decentralized communicating filters. Information exchange would only be

necessary when two robots have relative measurements, with bandwidth on order of $\mathcal{O}(n)$, but requires a fully connected and synchronous network at all times.

Approximations of the centralized method can extend the framework to decrease convergence time [24] or computational cost without loss of accuracy [25, 26, 27].

Very promising has been the work in [28] which proposed a CL framework that tracks correlations while limiting communication to the two robots that obtain a relative measurement. The algorithm is recursive, reducing memory requirements and it supports generic measurement models. This algorithm was augmented by [29] to be flexible to different measurement modalities and handle the double counting problem.

2.4 Vehicular Ad-Hoc Network

VANets are a type of wireless multi-hop network designed to have rapid changes in topology due to moving nodes [30, 31]. These networks are capable of supporting a large range of applications, but in the case of this research work, they are instrumental in the exchange of state information and covariance estimates. In the growing effort to grow and standardize such networks IEEE 802.11 was amended to include wireless access in vehicular environments which would greatly facilitate the exchange of information in V2V and V2X communication [32].

VANets have been used in numerous applications, from traffic management and routing [33, 34, 35] to aiding localization [36]. While the use of VANets poses interesting security challenges, there is work being performed to find robust solutions [37]. Most interestingly, VANets have been used with ground and air vehicles for cooperative methods [38, 39].

In this research, it will be assumed that a synchronized and connected VANet exists in order for vehicles to exchange state information and measurements.

3. RESEARCH METHODOLOGY

This section will outline the methods used to research the performance of UWB modules in landmark and collaborative localization. As stated, the goal of this thesis work to explore the application of UWB ranging technology to the localization of autonomous vehicles. This will be done by creating a flexible kinematic simulation of vehicles that incorporates the various sensors available to the unmanned systems lab. This simulation framework will be used to test various algorithms and estimate improvements from their implementation with UWB measurements using Monte Carlo simulations. These algorithms will be testing physically using experimental data collected using an vehicle in an urban environment.

3.1 Ultra-Wideband Evaluation

As the UWB modules are novel sensors, preliminary evaluation will be performed in order to better model the expected output. The first step in this process would be to ensure measurements can be reliably obtained from the sensors. To do this, the DecaRangeRTLS PC Application provided by Decawave will be used to log the ranging output of the UWB transceiver [40].

These measurements will be parsed in MatLab in order to determine the maximum reliable distance which the UWB module can measure. In order to determine the standard deviation and bias of ranging errors, two modules will be held at a fixed distance, measured by tape. The modules shall be allowed to reach steady state, and a sample of at least 100 measurements will be taken. This process will be repeated in 10 meter increments from 10 to 100 meters.

In order for a Kalman filter to be the best linear unbiased estimator, an assumption made is that the residuals of the measurements are normally distributed. As such, the sample errors will be evaluated for conformity to a normal distribution. This will be done by creating a normal probability plot. If the plot is approximately linear, a Jarque–Bera test will be performed to determine the likelihood the measurement errors do not conform to a normal distribution [41].

3.2 Simulation Design

In order to evaluate the methods used in collaborative localization with the UWB modules, it is imperative to have a repeatable and reliable simulation. The simulation proposed would facilitate the development and testing of collaborative localization algorithms, and provide initial estimates of performance. Requirements for the simulation were:

- Simulate vehicles using a kinematic model as truth
- Develop sensor models for IMU, odometry, GNSS, and UWB sensors
- Accept configurations for different scenarios, number of vehicles, and sensor availability
- Simultaneously run multiple algorithms for state estimation
- Run Monte Carlo simulations of scenarios to evaluate the distribution and errors of various estimation algorithms

3.2.1 Kinematic Model

In order to model the vehicles, the kinematics will be derived from the center of the rear wheel axis, using the methods described in [42]. These kinematics will be used to derive the differential equations of motion for the kinematic model vehicle. The simulation will implement these differential equations in a fixed step process that will approximate the motion for small steering angles.

3.2.2 Simulation Architecture

The simulation will be structured in a way to best emulate the real-world implementation of the localization algorithm. As such, each vehicle will maintain its own self estimate as well as physical and sensor characteristics. Such an architecture would follow the current state of roadways and not rely on the introduction of centralized computing and other infrastructures. This separation will be created through the use of objects. As such, each object instance will be isolated in its filtering

and private measurements, and will only be able to interact with other vehicles through relative measurements and state exchange.

The system will allow state exchange between vehicles through the use of an intermediate object that stores vehicle information and can introduce a step delay in the exchange of state. This simulated network object will handle the relative measurement models as well as the state truth and state estimates from each vehicle.

3.2.3 Monte Carlo Simulation

To explore the localization accuracy of each situation given the various random sampling errors in each sensor, a Monte Carlo (MC) method will be implemented [43]. Using random sample draws for each sensor that follow their modelled distribution, each simulation run will be different. In order to better understand performance, a MC method runs each scenario many times, collecting information on the state estimation error for each run. This information, when collected, gives a distribution of errors that approach an analytical solution of the error estimation problem. Due to the changing scenarios and complexity of the problem, a MC solution is much more attainable and can approximate otherwise incalculable values.

Therefore, a final object that acts as the simulation wrapper will be created in order to manage the simulation run, vehicle objects, and VANet object. This simulation object will collect the state estimation errors for each run and will allow simulations to be run in parallel to decrease computing time. To produce a relatively smooth distribution that approximates the true error distributions of each state estimator, each scenario will be run 5,000 times. RMS errors for each run will be collected to estimate the distribution of mean errors in the various configurations.

3.3 Collaborative Localization

UWB ranging measurements can act as a typical update to a Extended Kalman Filter (EKF) by finding a vehicle's distance to known landmarks. Additionally, by measuring distance between vehicles, it is relatively straightforward to derive the UWB update equations for a centralized localization system that receives all control and update measurements. However, as discussed in

Chapter 2, there are difficulties in scaling up a centralized network, and it is not robust against single points of failure. Additionally, decentralized methods are flexible in the introduction of nodes and require less infrastructure than centralized methods for passenger vehicles. Therefore, it is advantageous to incorporate a decentralized collaborative localization (DCL) algorithm as described in [28].

The DCL algorithm requires state exchange and relative measurements between vehicles. As such, this algorithm will be implemented into the simulation using the state exchange protocols outlined in Section 3.2.2. Here, a realistic application of DCL will be shown due to the possibility of a large network of simulated vehicles.

Additionally, the DCL algorithm will be implemented experimentally using a single moving vehicle and a simulated stationary vehicle, due to current limitations on in-person testing. Such an experiment will not fully validate the algorithm's application, but with estimations of state and uncertainty of a stationary vehicle, any improvements in collaborating with stationary vehicles can be shown.

3.4 Experimental Design

Once the performance of the UWB sensors is validated, physical experiments will be performed to show any localization accuracy improvements through the use of UWB sensors and collaborative localization.

3.4.1 UWB Sensor Package

In order to make use of the sensor data and have real-time analysis, it is necessary to make the UWB sensor measurements available within the ROS framework. In order to do this, a ROS driver package will be created to utilize the serial interface of the EVK1000 boards from Decawave. This ROS package will define custom messages for the UWB sensors to facilitate their incorporation into algorithms. Additionally, launch files and ROS nodes will be defined to decrease startup time to access sensor data. Finally, the package will be compatible with catkin to increase usability.

3.4.2 RTK GNSS Sensor Package

Additionally, estimations will be compared to a ground truth produced from RTK corrected GNSS measurements. Using a ZED-F9P from u-blox, positioning accuracy of 0.01 meter are achievable and errors are estimated in real time [44]. As the error of these corrected GNSS modules is significantly lower than the expected error of the state estimate, these RTK GNSS measurements will be used as a ground truth for means of estimating state error.

In order to interface with these modules, a similar ROS driver will be created to connect to the serial interface. This package will expand upon an existing serial package [45]. Specific contributions will be the addition of interface definitions to include the ZED-F9P module. This will allow the measurements of relative position and angle estimates from two GNSS antennae to be communicated via serial messages. Additionally, this new package will wrap the serial interface with a ROS driver. This includes the definition of nodes, messages, and launch files in order to make real-time measurements available. Finally, the package will similarly be compatible with catkin to increase usability.

3.4.3 Test Scenarios

The collaborative and landmark localization using UWB ranging will be testing physically using a autonomous Polaris GEM e6 from the Unmanned Systems Lab. This autonomous trolley has access to IMU, odometry, and GNSS measurements through ROS. To add measurements for the UWB ranging and RTK GNSS, both the UWB tags and RTK GNSS antennae will be mounted to the roof rack of the vehicle. Measurements will be taken to determine the frame transformations for both sets of sensors, and will be incorporated into the estimation and evaluation algorithms.

With the sensors mounted and the ROS drivers installed, a subset of the simulated scenarios will be created that only require a total of four UWB modules, which is a limitation on the number currently available in the lab. This subset will cover street crossing and moving parallel to a single vehicle, street crossing and moving through an intersection that has landmarks installed, and moving through a tunnel with landmarks.

Multiple data sets of each scenario will be collected and recorded to allow for playback in real time for tuning and algorithm testing. Using the resulting state estimate and transformed RTK GNSS data, accuracy evaluations will be made with and without the UWB sensor updates as the difference of the position estimate and RTK GNSS position. Using the RMS position errors in these runs, it will be possible to compare the performance gains in adding UWB modules in the simulations and in experimental environments.

4. THEORY

This section will introduce the theory behind the methods and algorithms that comprise and support this thesis work.

4.1 Tests for Normality

Incorporating measurements into a Kalman filter gives the optimal estimate of a state with the given measurement and prior information. An assumption made in the filter is the measurement residuals are drawn from a zero mean Gaussian distribution [46]. In the evaluation of the UWB modules, it is necessary to show that the ranging estimates are given by

$$z = \|p_i - p_j\| + \nu \tag{4.1}$$

where p_i and p_j are the UWB tag locations and where

$$\nu \sim \mathcal{N}(0, Q_i^t) \tag{4.2}$$

In order to verity this, samples are to be taken at a known distance and the resulting distribution will give

$$\nu = z - \|p_i - p_j\|_{true} \tag{4.3}$$

Using the calculated residuals, a normal probability plot can be used to create an visual check of normality [47]. The normal probability plot transforms a sorted distribution of samples with the following

$$z_i = \Phi^{-1}\left(\frac{i-1}{n+1-2a}\right) \tag{4.4}$$

for $i = 1, 2...2$, where $a = 1/2$ for $n > 10$, and Φ^{-1} is the standard normal quantile function.

If the data are consistent with samples from a normal distribution, they will show an approximately linear plot of z values. Any trends or deviations from the line are indications of subsequent deviations from a normal distribution.

Additionally, a Jarque–Bera goodness of fit test can be performed on the data to provide a less subjective measure of normality by determining if the sample data have the skewness and kurtosis matching a normal distribution [41]. The test is defined as

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right) \tag{4.5}$$

where the skewness S and kurtosis K are given by

$$S = \frac{\frac{1}{n}\Sigma_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{n}\Sigma_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}} \tag{4.6}$$

$$K = \frac{\frac{1}{n}\Sigma_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\Sigma_{i=1}^{n}(x_i - \bar{x})^2\right)^{2}} \tag{4.7}$$

With sufficiently large samples, the JB test statistic follows a Chi-Square distribution with 2 degrees of freedom. Therefore, an associated probability $p < 0.05$ would indicate the samples come from a normal distribution.

4.2 Kinematic Modelling

In order to simplify the kinematic modelling, a rear-axle bicycle model is used. Using the center of the rear axle as the reference point and a system defined in Figure 4.1, giving the following relations.

$$\dot{\theta} = \omega = \frac{v}{R} \tag{4.8}$$

$$\tan(\delta) = \frac{L}{R} \tag{4.9}$$

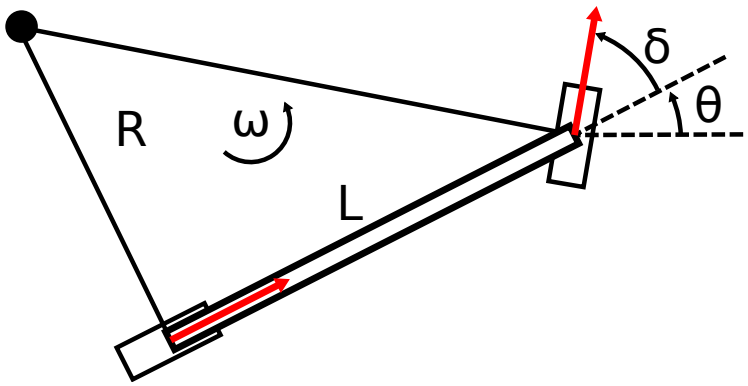


Figure 4.1: Rear Axle Bicycle Model

Therefore, the following differential equations can be derived.

$$\dot{x}_r = v\cos(\theta) \tag{4.10}$$

$$\dot{y}_r = v\sin(\theta) \tag{4.11}$$

$$\dot{\theta}_r = \omega = \frac{v\tan\delta}{L} \tag{4.12}$$

where v is the vehicle speed, δ is the angle of the front wheel, and L is the wheelbase length.

4.3 Decentralized Collaborative Localization

The decentralized collaborative localization algorithm is an approximation of a centralized kalman filter that distributes computation and reduces communication between nodes. The estimation of the entire state of a network is broken down into smaller filters where each vehicle has private controls and measurements that are used internally and relative measurements that require the sharing of state and covariance information.

For a network of N vehicles, let x_i be the pose of vehicle i. Each vehicle has an estimate of its own state $\hat{x}_i^t$ and covariance Σ_{ii}^t at time t. Let each vehicle maintain a joint state estimate of the system where $\hat{x}^t = [\hat{x}_1^t; ...; \hat{x}_N^t]$ and $\Sigma^t = [\Sigma_{ij}^t]_{1\leq i,j\leq N}$. With this network, an outline of the

collaborative update algorithm is shown in Chapter A.

4.3.1 Initialization

It is assumed that at the begging of any network, the vehicles positions are uncorrelated. Therefore, the network state can be initialized with the initial beliefs bel_i of each vehicle and the cross correlation is set as

$$\{\Sigma_{ij}^t = 0\}_{\forall i \neq j} \tag{4.13}$$

When the vehicles come into sensing range, generally the cross correlation $\Sigma_i^{t+1} j \neq 0$, and therefore decompose according to [48].

$$\Sigma_{ij}^{t+1} = \sigma_{ij}^{t+1} (\sigma_{ji}^{t+1})^T \tag{4.14}$$

where any decomposition is valid. For simplicity, the covariance is decomposed as $\sigma_{ij}^{t+1} = \Sigma_{ij}^{t+1}$ and $\sigma_{ji}^{t+1} = I$. This allows each vehicle to track contributions to the cross-correlation terms without the need for communication during each prediction and private update.

4.3.2 Control

It is assumed that the vehicles follow a motion model g where control input U is an IMU measurement. The prediction step for vehicle i is given by the standard EKF equations without updates to other vehicles

$$\hat{x}_i^{t+1} = g(\hat{x}_i^t, U_i^t) \tag{4.15}$$

$$\Sigma_{ii}^{t+1} = G_i^t \Sigma_{ii}^t (G_i^t)^T + R_i^t \tag{4.16}$$

$$\Sigma_{ij}^{t+1} = G_i \Sigma_{ij}^t \tag{4.17}$$

$$\hat{x}_j^{t+1} = \hat{x}_j^t \tag{4.18}$$

$$\Sigma_{jj}^{t+1} = \Sigma_{jj}^t \tag{4.19}$$

for all $j \neq i$ where the linearized motion model is defined as

$$G_i^t = \frac{\partial g(x, U)}{\partial x}(\hat{x}_i^t, U_i^t) \tag{4.20}$$

and R_i^t is the process noise. In the particular case of using a constant velocity prediction step, the motion model can be written as

$$\hat{x}_i^{t+1} = G^t \hat{x}_i^t + B^t U_i^t \tag{4.21}$$

where

$$G^t = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B^t = \begin{bmatrix} \frac{dt^2}{2} & 0 & 0 \\ 0 & \frac{dt^2}{2} & 0 \\ 0 & 0 & dt \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & 1 \end{bmatrix}, U_i^t = \begin{bmatrix} a_x \\ a_y \\ \omega_z \end{bmatrix} \tag{4.22}$$

To reduce communication, the cross-correlations of Equation (4.17) can be correctly updated by instead updating

$$\sigma_{ij}^{t+1} = G\sigma_{ij}^t \tag{4.23}$$

4.3.3 Private Update

It is assumed that the private update measurement to be a function of the state of a single vehicle with a Gaussian error disturbance.

$$z_i^t = h(\hat{x}_i^t) + \nu_p \tag{4.24}$$

with $\nu_p \sim \mathcal{N}(0, M_i^t)$, M_i^t being the private measurement noise, and h being the measurement

model. The private update step for the system is

$$\hat{x}_i^{t+1} = \hat{x}_i^t + K_i^t \left[z_i^t - h(\hat{x}_i^t) \right] \tag{4.25}$$

$$\hat{x}_j^{t+1} = \hat{x}_j^t + K_j^t \left[z_i^t - h(\hat{x}_i^t) \right] \tag{4.26}$$

$$\Sigma_{ii}^{t+1} = (I - K_i^t H_i^t)\Sigma_{ii}^t \tag{4.27}$$

$$\Sigma_{ij}^{t+1} = (I - K_i^t H_i^t)\Sigma_{ij}^t \tag{4.28}$$

$$\Sigma_{jj}^{t+1} = \Sigma_{jj}^t - K_j^t H_i^t \Sigma_{ij}^t \tag{4.29}$$

where

$$H_i^t = \frac{\partial h(x)}{\partial x} \left(\hat{x}_i^t \right) \tag{4.30}$$

$$K_{i/j}^t = \Sigma_{ii/jj}^t (H_i^t)^T S^{-1} \tag{4.31}$$

$$S = H_i^t \Sigma_{ii}^t (H_i^t)^T + Q_i^t \tag{4.32}$$

and the cross terms are instead correctly updated with

$$\sigma_{ij}^{t+1} = (I - K_i^t H_i^t)\sigma_{ij}^t \tag{4.33}$$

for all $i \neq j$. To avoid communication during this step, Equations (4.26) and (4.29) are replaced with

$$\hat{x}_j^{t+1} \approx \hat{x}_j^t \tag{4.34}$$

$$\Sigma_{jj}^{t+1} \approx \Sigma_{jj}^t \tag{4.35}$$

When the private update is from the odometry measurement, the measurement equation is given by

$$h(\hat{x}_i^t) = \begin{bmatrix} v_i \\ v_i \frac{\tan(\delta_i)}{b_i} \end{bmatrix} \tag{4.36}$$

which produces the following linearized measurement matrix

$$H_k = \begin{bmatrix} 0 & 0 & 0 & \frac{v_{i_x}}{v_i} & \frac{v_{i_y}}{v_i} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.37}$$

where b is the vehicle wheelbase.

It is also possible for the UWB measurements to be used as private updates, when measuring distances to landmarks. In this case, the measurement model is given by

$$h(\hat{x}_i^t) = \begin{bmatrix} \|p_i - p_L\| \end{bmatrix} \tag{4.38}$$

where p_i is the position of the vehicle tag and p_L is the position of the landmark. This measurement equation produces the following linearized measurement matrix

$$H_k = \begin{bmatrix} \frac{\Delta X}{\sqrt{z}}, & \frac{\Delta Y}{\sqrt{z}}, & \frac{1}{\sqrt{z}} \begin{bmatrix} p_i - p_L \end{bmatrix} R[\theta] \begin{bmatrix} c_y \\ c_x \\ 0 \end{bmatrix}, & 0, & 0, & 0 \end{bmatrix} \tag{4.39}$$

where c are the anchor offsets on the vehicle, ΔX,ΔY are the differences in the position estimates of the anchor and tag, and $R[\theta]$ is a rotation matrix about the vertical axis using the current orientation of vehicle i.

4.3.4 Relative Update

It is assumed that the relative update measurement to be a function of the state of two vehicles i and j, with a Gaussian error disturbance.

$$z_i^t = f(\hat{x}_i^t) + \nu_r \tag{4.40}$$

with $\nu_r \sim \mathcal{N}(0, Q_{ij}^t)$, Q_{ij}^t being the relative measurement noise, and f being the relative measurement model. The relative update step for the system is

$$\Sigma_{ij/ji}^t = \sigma_{ij/ji}^t (\sigma_{ji/ij}^t)^T \tag{4.41}$$

$$\hat{x}_{i/j}^{t+1} = \hat{x}_{i/j}^t + K_{i/j} \left[z_{ij}^t - h(\hat{x}_i^{t+1}, \hat{x}_j^{t+1}) \right] \tag{4.42}$$

$$\Sigma_{ii/jj}^{t+1} = (I - K_{i/j} F_{i/j}) \Sigma_{ii/jj}^t - K_{i/j} F_{j/i} \Sigma_{ji/ij}^t \tag{4.43}$$

$$\Sigma_{ij}^{t+1} = (I - K_i F_i) \Sigma_{ij}^t - K_i F_j \Sigma_{jj}^t \tag{4.44}$$

$$\sigma_{ij}^{t+1} = \Sigma_{ij}^{t+1} \tag{4.45}$$

$$\sigma_{ji}^{t+1} = I \tag{4.46}$$

$$\sigma_{ik/jk}^{t+1} = \Sigma_{ii/jj}^{t+1} \left(\Sigma_{ii/jj}^{t+1} \right)^{-1} \sigma_{ik/jk}^t \tag{4.47}$$

and remaining cross terms updated with

$$\Sigma_{kk}^{t+1} \approx \Sigma_{kk}^t \tag{4.48}$$

$$\hat{x}_k^{t+1} \approx \hat{x}_k^t \tag{4.49}$$

for $k \in \{1, ..., N\} \backslash \{i, j\}$ with linearization

$$F_{i/j}^t = \frac{\partial f(x_i, x_j)}{\partial x_{i/j}} \left(\hat{x}_{i/j}^t \right) \tag{4.50}$$

and Kalman gain

$$K = \begin{bmatrix} \Sigma_{ii}^t F_i^T & \Sigma_{ij}^t F_j^T \\ \Sigma_{ji}^t F_j^T & \Sigma_{jj}^t F_i^T \end{bmatrix} S^{-1} \tag{4.51}$$

and residual covariance

$$S = \begin{bmatrix} F_i & F_j \end{bmatrix} \begin{bmatrix} \Sigma_{ii}^t & \Sigma_{ij}^t \\ \Sigma_{ji}^t & \Sigma_{jj}^t \end{bmatrix} \begin{bmatrix} F_i^T \\ F_j^T \end{bmatrix} + Q_{ij}^t \tag{4.52}$$

The cross-correlation is decomposed and distributed with

$$\Sigma_{ij}^{t+1} = \sigma_{ij}^{t+1}(\sigma_{ji}^{t+1})^T \tag{4.53}$$

where, for simplicity, we assign

$$\sigma_{ij}^{t+1} = \Sigma_{ij}^{t+1} \tag{4.54}$$

$$(\sigma_{ji}^{t+1}) = I \tag{4.55}$$

5. RESULTS

5.1 Collaborative Simulation

A collaborative simulation was created in MatLab to be a framework for testing and evaluating algorithms and filters in collaborative localization of autonomous vehicles. The simulation was designed to simulate various scenarios with a indefinite number of collaborative vehicles. Each vehicle instance is capable of handling its own sensor stack and estimations, while also sharing state estimates.

Flexibility in the estimation algorithms was built in to the simulation in order to more easily evaluate estimation performance in any given scenario. Multiple independent estimations algorithms can be run for each vehicle simultaneously. By ensuring that each vehicle maintains its own state estimates and runs independent algorithms, it was possible to predict the expected improvements from the addition of collaborative localization, landmark localization, or both.

Pre-built scenes are included in the simulator in order to quickly assess performance gains and create tests with various numbers, speeds, and directions of vehicles. The current situations include: *Parallel Motion*, *Street Crossings*, and *Underground Tunnels*.

Finally, test scripts were generated in order to run single instances of the scenes, or a Monte Carlo simulation of the scene. The Monte Carlo test script runs thousands of simulations in parallel and collects the distribution of state errors. This distribution can be used to create an approximation of the state error probability density function. Not only does this provide expected values for the state error, but also gives a greater understanding to the possible distribution of errors.

5.1.1 Simulation Objects

The ability to define objects within MatLab was utilized to increase the flexibility of the simulation. A *KinematicCar* object was defined to handle each instance within the simulation. These instances independently would handle their own state estimation and measurement models.

Ultra-Wideband ranging between vehicles naturally requires inter-vehicle information sharing.

To handle this and ensure availability of data, a separate *UltraWideband* object was defined that would handle the storage and update of UWB tag information. Additionally, this object would handle the measurement models of the UWB sensors.

Lastly, in order to streamline the simulation process, a wrapper *Simulation* object was created that would create the necessary *KinematicCar* and *UltraWideband* objects and initialize based on a configuration file. This allows for a quick evaluation of various experiments. The *Simulation* instance manages MC simulations following the algorithm outlined in Chapter A. Additionally, the *Simulation* instance handles the parallel computing of the experimental runs such that MC simulation results are quickly realizable even with large numbers of vehicles to simulate.

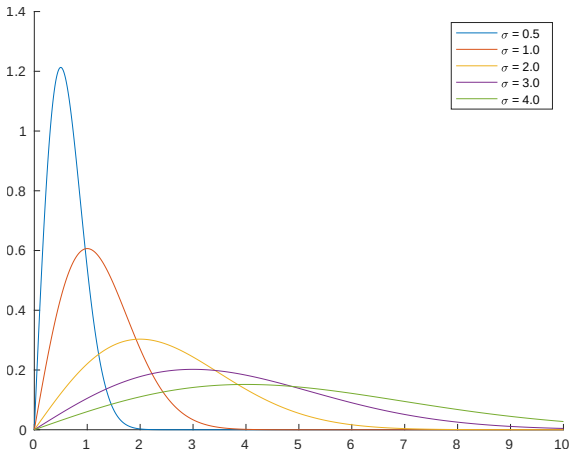
5.1.2 GNSS Sensing Model

The GNSS sensing model was created with the VN-300 in mind. Because the specifications give GNSS accuracy in Circular Error Probable (CEP), it was desirable to create a model that took that value as a parameter. In order to simulate the GNSS with errors, the CEP value is transformed into a distance root mean square (DRMS) value, which is approximately 84.93% of CEP [49]. This DRMS value is the sigma of Rayleigh distribution exemplified in Figure 5.1a. Using a Rayleigh random number generator, the error distance is generated, and a uniform random number generator is used to determine the angle. Combined, this gives a two-dimensional offset value for the GNSS error model given by Equations (5.1) and (5.2).

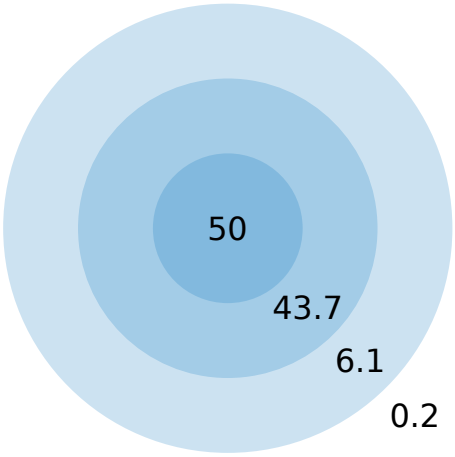
$$r \sim Rayleigh(0.8493 \cdot CEP) \tag{5.1}$$

$$\theta \sim U(0, 2\pi) \tag{5.2}$$

Additionally, the VN-300 is capable of estimating both heading and vehicle speed in UTM coordinates. Errors on these measurements are assumed to be normal, and the standard deviations were given by the product specifications.



(a) Example of various Rayleigh distributions with different sigma values



(b) Probability of landing in each CEP ring

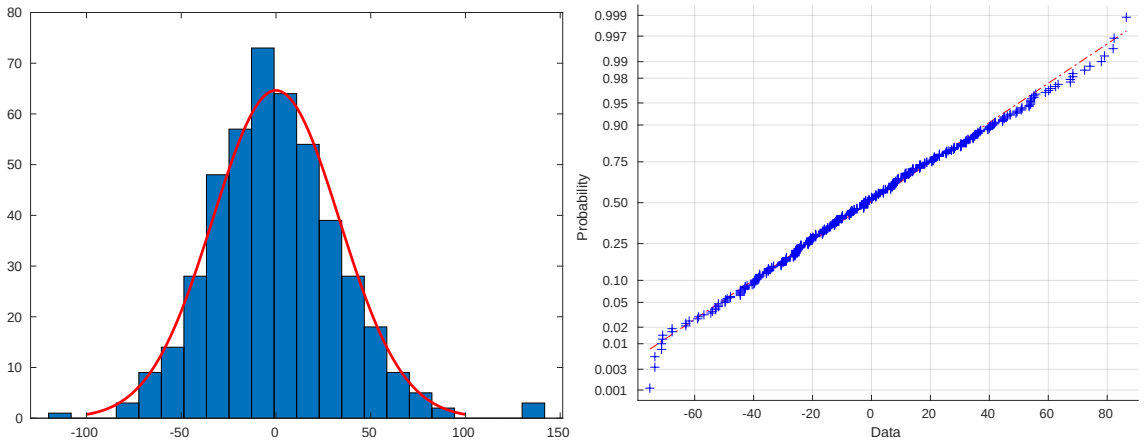
Figure 5.1: Properties of the Rayleigh distribution and CEP

5.1.3 UWB Sensing Model

Since the Kalman Filter assumes Gaussian measurement errors, it was desirable to validate the distribution errors of these UWB ranging modules. Measurements were taken in increments of 10 meters from 10 to 100 meters. The distribution of errors are summarized in Figure 5.2a which appears to be approximately normal. The normal probability plot shown in Figure 5.2b is approximately linear, which indicates Gaussian errors. The tails shown symmetric deviations, which could indicate outliers that deviate from the normal distribution. A more rigorous Jarque–Bera test was performed which failed to reject the null hypothesis with a probability $p < 0.05$ that the sample errors come from a Gaussian distribution. Therefore, the UWB sensing model was implemented with Gaussian normal randomly distributed errors with a measured standard deviation of 0.31 meters, modelled as the difference of tag positions at time time

$$r_{ij}^t \sim \mathcal{N}(\left\|p_i^t - p_j^t\right\|, \sigma^2) \tag{5.3}$$

Additionally, the signal was shown to have an effective maximum in the exploratory tests of 110 meters. Because measurements were not shown to be reliable over 100 meters, a limit of 100



(a) Normal Distribution Fit

(b) Normal Probability Plot

Figure 5.2: Visual methods for testing normality of UWB measurement residuals

meters was implemented in the UWB sensing model. Finally, the model simulates the measurement availability of the UWB sensors, which give the entire network of measurements at once.

5.1.4 Simulation Output

The resulting output of this software package is a visual animation of the simulation exemplified in Figure 5.3, position error summaries for each vehicle, and the final state and covariance. When compiled in Monte-Carlo simulations, these data can be used to evaluate filter performance improvements. Additionally, the software is configurable to automatically create videos from the animation output of the simulation.

5.2 Simulation Results

Using the simulation framework, it was possible to test a wide variety of vehicle environments and sensor configurations. These configurations included vehicles moving parallel to each other, vehicles moving perpendicular in street crossings, and moving in groups in tunnels without GNSS data. Additionally, UWB landmarks could be added to each of these scenarios.

In order to run realistic simulations, error parameters were drawn from sensor data sheets or calibration procedures. A summary of the parameters used is shown in Table 5.1. Simulations were allowed to run for 20 seconds over 10,000 iterations.

Error Parameter	Error	Units
GPS CEP	1.00	m
PACMod Velocity	0.05	$\frac{m}{s}$
PACMod Steering	0.05	$^\circ$
UWB Ranging	0.30	m

Table 5.1: Error Parameters used in Simulations

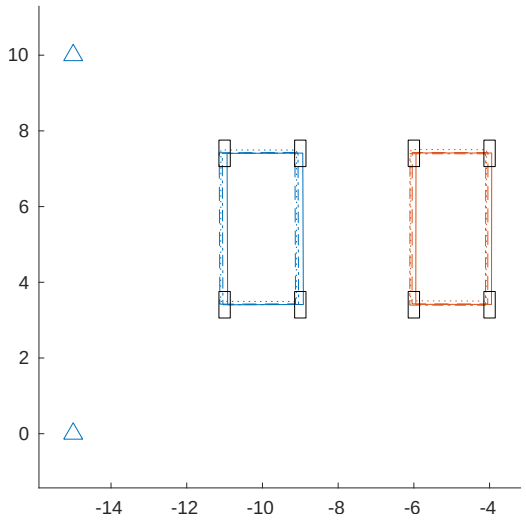


Figure 5.3: Scene with two vehicles moving in parallel

5.2.1 Parallel Vehicles Environment

This experiment tested the motion of two vehicles moving in parallel with full sensor access. Figure 5.3 outlines the environment with both cars and the optional landmarks. Figure 5.4 shows the RMS position error through the Monte-Carlo simulation of this environment using the four different localization algorithms.

Because the vehicles were moving in parallel, reductions in horizontal position errors were seen, which is to be expected, as this is the direction down range of the UWB ranging measurement. Incorporating landmarks as additional points of measurement continued to improve the localization accuracy of the vehicles when using UWB measurements. This also was expected as increasing

the number of measurements should improve the localization estimate. The addition of landmarks significantly decreased both the mean and deviations of the position errors.

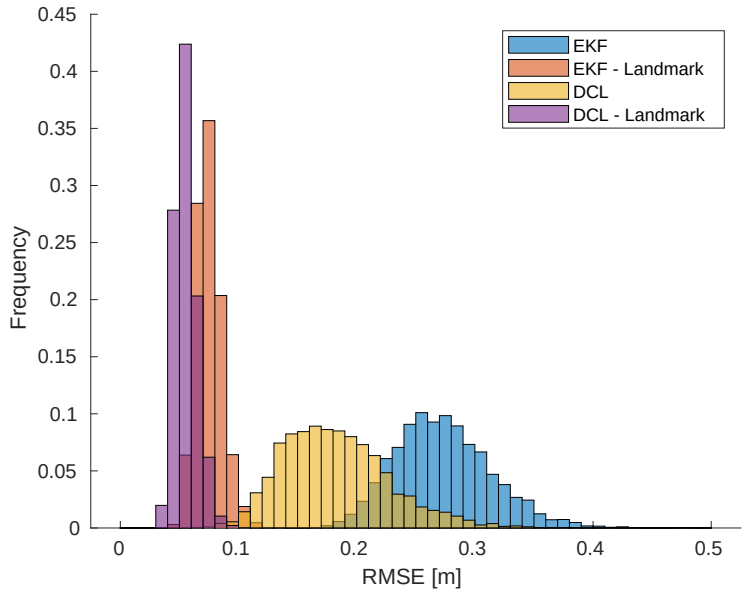


Figure 5.4: Distribution of position errors of two collaborative vehicles moving in parallel

5.2.2 Four-Way Intersection Environment

This experiment tested the motion of two vehicles moving in parallel yet opposite directions through a four way stop, while two other vehicles are waiting to proceed. Figure 5.5 outlines the environment with all four cars and the optional landmarks. Figure 5.6 shows the RMS position error through the Monte-Carlo simulation of this environment using the four different localization algorithms.

As expected, increasing the number of vehicles (and therefore the number of measurements) increased the localization accuracy of the vehicles. This is a promising result for the continued expansion of intelligent vehicle networks as it suggests more connected vehicles could lead to more accurately located vehicles. However, it is expected that increases in the number of measurements was not the sole factor in improving the vehicles' localization accuracy. As this environment had

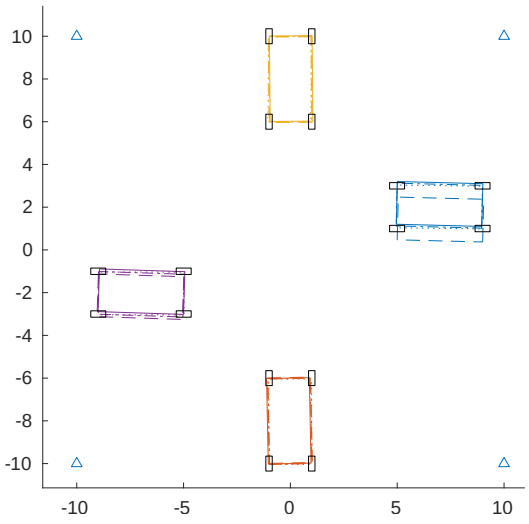


Figure 5.5: Scene with four vehicles moving through a four-way stop

measurements in both the lateral and longitudinal directions of the vehicle, a significant reduction in error was observed in both directions. This adds on to the improvement seen in the parallel test.

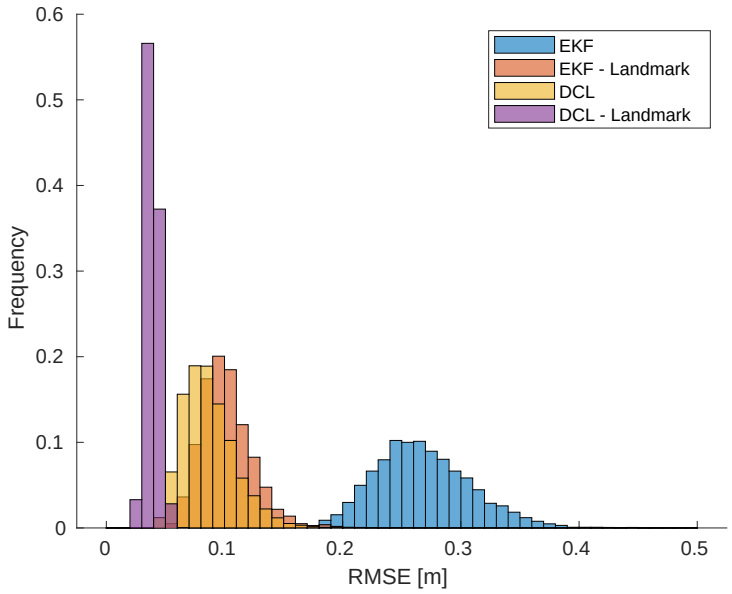


Figure 5.6: Distribution of position errors of two collaborative vehicles moving through a four-way stop

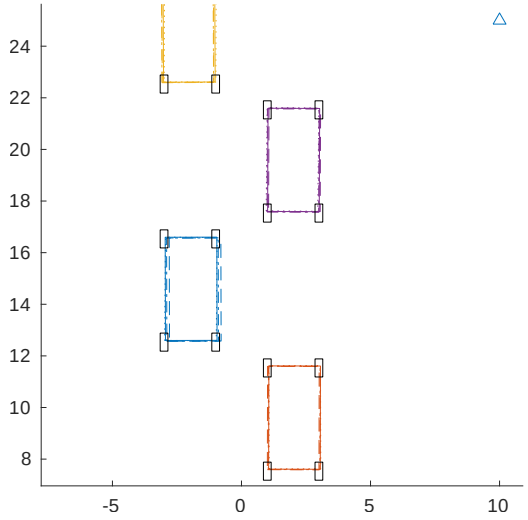


Figure 5.7: Scene with six vehicles moving through a tunnel

5.2.3 Tunnel Environment

This experiment tested the motion of six vehicles moving in parallel through a tunnel environment that obscures GNSS measurements. Figure 5.7 outlines the environment with all six cars and the optional landmarks. Figure 5.8 shows the RMS position error through the Monte-Carlo simulation of this environment using the EKF and DCL algorithms without landmark ranging. Figure 5.8 shows the RMS position error through the Monte-Carlo simulation of this environment using the EKF and DCL algorithms with additional ranging to landmarks every 25 meters.

As expected, an environment where GNSS measurements were obstructed showed the greatest improvement from the addition of UWB measurements. Using these measurements in a purely collaborative framework only showed minor improvements to localization accuracy. Even though the UWB modules offer additional measurements to the system, they are relative measurements between similar vehicles and therefore are still subject to drift. A larger number of vehicles would be expected to offer greater improvements as the RMS drift would approach zero.

With the addition of landmarks in the environment, however, significant improvements in localization accuracy were seen. This was expected as UWB ranging measurements to landmarks

would be the only exteroceptive measurement made by the vehicles. This indicates the additional framework of adding UWB ranging to landmarks in GNSS-denied environments could expand the working range of autonomous vehicles.

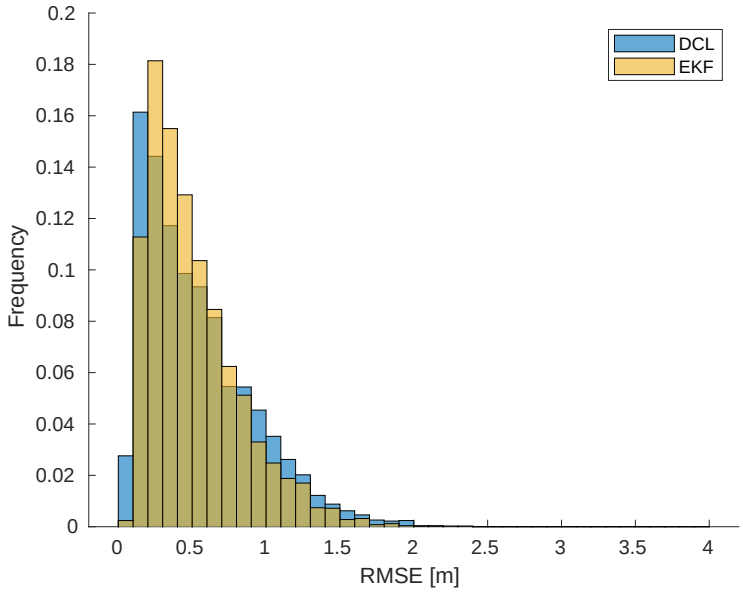


Figure 5.8: Distribution of position errors of two collaborative vehicles moving through a tunnel without landmarks

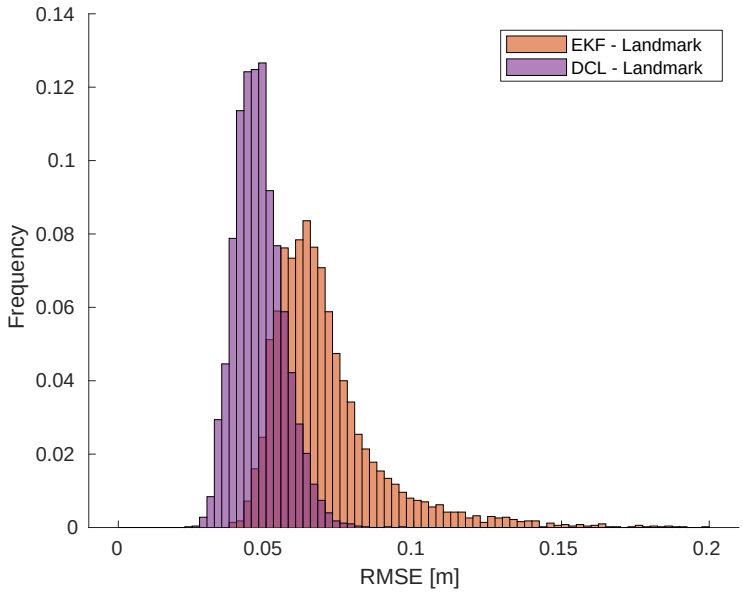


Figure 5.9: Distribution of position errors of two collaborative vehicles moving through a tunnel with landmarks

5.2.4 Simulation Results Summary

Across the simulations performed, many improvements were seen. Additionally, this implementation and subsequent tests validated the architecture and algorithms used. A numerical summary of the simulation results is shown in Table 5.2. Additionally, the simulations showed the following trends

- UWB measurements offered improvements to localization accuracy across all scenarios especially in cases GPS-deprivation
- While still being subject to drift, using more vehicles generally improved the accuracy of localization and decreased overall deviation in the error means.
- Much like GPS dilution of precision, improved UWB sensing geometry improved the localization accuracy of vehicles.
- The addition of landmarks improved the localization accuracy of all environments

Experiment	GNSS	EKF	EKF LMK	DCL	DCL LMK
Parallel	Yes	0.273	0.075	0.182	0.056
Crossing	Yes	0.263	0.102	0.087	0.039
Tunnel	No	0.537	0.073	0.514	0.048

Table 5.2: Summary of simulation RMS position errors, in meters

5.3 Decawave ROS Driver

In order to interface with the Decawave EVK1000 UWB ranging modules, a ROS package was developed using python3 using serial commands. The decawave_ros package includes an installation script and can be built with catkin [50]. This package contains the source code to connect to the serial interface of the EKK1000 boards and parse the resultant messages that are output when ranging data is available.

This serial interface and parsing is wrapped by a ROS node that takes the data strings output by the parser and publishes the measured distances along with the respective anchor and tag numbers. This data becomes very useful in collaborative localization because the tag information is key in making use of decentralized algorithms. As such, the package greatly facilitated the data collection used in later sections of this work.

5.4 UBX ROS Driver

In order to compare to a high-accuracy ground truth, it was important that the system could use Real-Time Kinematic (RTK) GNSS correction, which provides up to 2 cm of accuracy. In order to use the ArduSimple RTK GNSS module, the UBXtranslator package was expanded to include different serial message definitions [51]. This package has a similar structure to the decawave_ros package where the serial message parsing is wrapped by a ROS node driver. This package is able to access the high-precision lattitude, longitude, and height data given by the module, as well as a relative heading measurement. As these measurements were used as a ground truth to the system, the development of this package was also very useful in facilitating the evaluation of the proposed collaborative localization algorithm.

5.5 Experimental Results

To further the results shown in simulation, the collaborative and landmark localization framework was tested in downtown Bryan, Texas using the Unmanned Systems Lab autonomous trolley exemplified in Figure 5.10. The UWB modules were mounted alongside a VectorNav VN-300 and ArduSimple RTK GNSS [52, 53]. Further information about the sensor offsets is outlined in Chapter B. The trolley itself includes PACMod, a by-wire kit prepared by Autonomous Stuff, which gives access to wheel speed and steering angle data through a provided ROS driver [54].

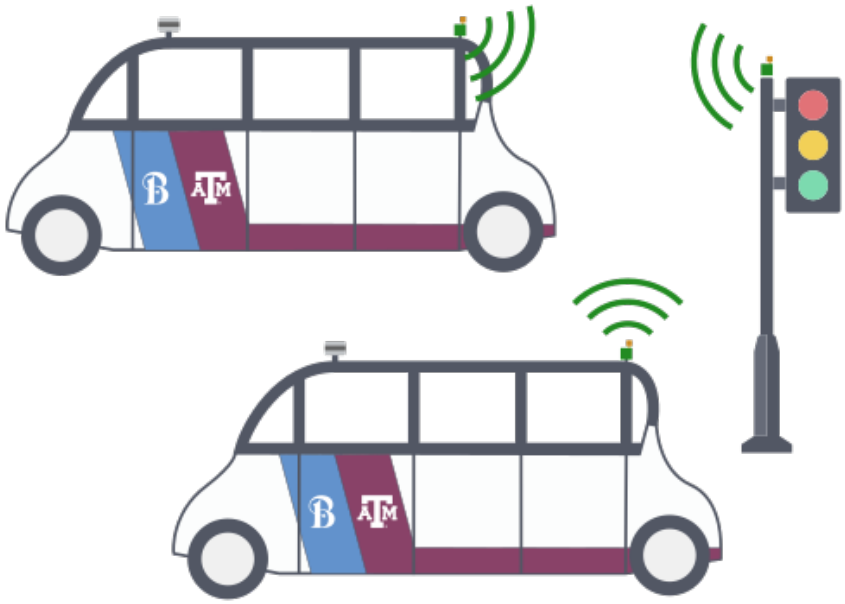


Figure 5.10: Example experimental setup with Trolleys and UWB sensing

Because of limitations on in-person testing and the number of golf carts in Bryan, only a single vehicle was present for testing. As such, the collaborative localization algorithm was only able to be tested with a simulated stationary vehicle. This did not affect the testing of landmark setups. Using tripods, the second set of UWB ranging modules were placed in various experimental setups to either represent a stationary second vehicle or road landmarks. These offsets are also outlined

in Chapter B. The experiments selected to experimentally test were a subset of those performed in Section 5.2 due to a limitation on the number of tags available. However, the current design of the tags allows for up to four anchors that range to up to eight tags at a time. With this in mind, a much larger variety of tests can be performed.

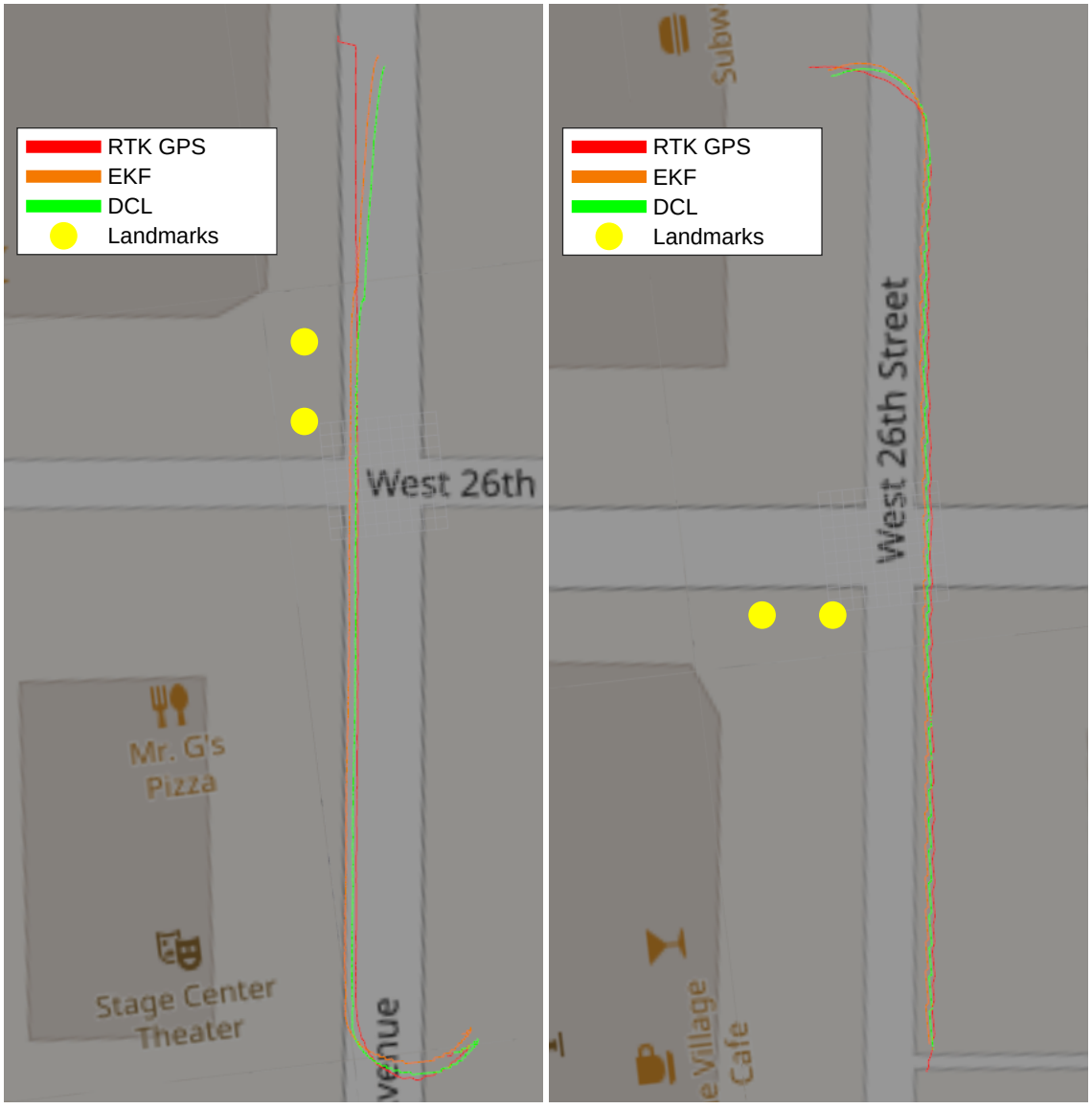
Additionally, experimental data showed a significantly reduced range of measurements taken, with few measurements being received over 60 meters. As this is an observation made after collecting data, additional more rigorous experimentation is necessary to determine the accuracy of this observation and what is a reasonable ranging measurement when UWB modules are mounted on vehicles and not just in open space.

5.5.1 Parallel Vehicles Environment

The results of implementing the collaborative localization algorithm in the parallel motion experiment increased the localization accuracy as expected with motion and estimates shown in Figure 5.11a. The most pronounced improvement came from the lateral correction of the position estimate. As this is the direction the measurements were generally taken, this is an expected result. As the measurements became further, and the baseline distance decreased with respect to the measurement distance, lateral sensitivity became large. As such, the relative updates became decreasingly effective. As such, near the end of the runs, even when receiving correction updates, it appears as if the vehicle is dead reckoning in the cross range direction.

5.5.2 Collaborative Street Crossing Environment

The results of implementing the collaborative localization algorithm in the perpendicular street crossing experiment increased the localization accuracy, as expected, with motion and estimates shown in Figure 5.11b. As with the parallel experiment, the most pronounced improvement came from the lateral correction of the position estimate. As this is the down range direction the measurements were generally taken, this is an expected result. Similar to the parallel motion experiment, UWB measurements made at increasing distances had decreasing effects on the state estimate. However, because the measurements were less colinear due to the orientation of the stationary ve-



(a) Parallel Motion with Stationary Vehicle (b) Street Crossing with Stationary Vehicle

Figure 5.11: Experimental results of the DCL with UWB measurements in different scenarios

(a) Street Crossing with Landmarks (b) Tunnel with Landmarks

Figure 5.12: Experimental results of the EKF with landmark measurements in different scenarios

hicle, measurements seemed to suffer less geometric dilution of precision. This could help explain the significantly better performance of the DCL in this scenario.

5.5.3 Landmark Street Crossing Environment

The results of incorporating the private UWB measurements into the EKF in the street crossing environment increased the localization accuracy, as expected, with motion and estimates shown in Figure 5.12a. As with the collaborative test in this environment, lateral accuracy showed the largest improvement in the state estimate. As the geometry of this sensing environment was better than the collaborative environment, it is expected that a greater improvement in localization accuracy would be seen.

5.5.4 Landmark Tunnel Environment

The results of incorporating the private UWB measurements into the EKF in the tunnel environment significantly increased the localization accuracy, with motion and estimates shown in Figure 5.12b. Given that in this environment, GNSS corrections were denied, UWB measurements gave the only exteroceptive measurement, whose addition should regulate possible drift in the system. Therefore, just as in the simulation, adding these measurements greatly increased the localization accuracy of the vehicle while driving in the corridor. There does appear to be large oscillations in tracking the true position, which indicates that further tuning of the EKF may be necessary.

5.5.5 Experimental Results Summary

Across the experiments performed, improvements were seen in all when UWB measurements were added in either DCL or landmark localization in an EKF. These experiments have validated the simulation data and suggest a need to continue testing with multiple moving vehicles and more UWB tags. A numerical summary of the experimental results is shown in Table 5.3. Very similar to the simulations, the experimental results showed the following trends

- UWB measurements offered improvements to localization accuracy across all scenarios especially in cases of landmark usage and GPS-deprivation

- Much like GPS dilution of precision, improved UWB sensing geometry improved the localization accuracy of vehicles.
- Landmarks offered greater improvements to localization accuracy than collaboration alone

The key trend lacking from the simulations is a reduction in drift when multiple vehicles moving in parallel can collaboratively localize. When multiple vehicles and drivers are available with more tags, this will be a next step experiment in the exploration of using UWB ranging.

Experiment	GNSS	EKF	EKF LMK	DCL	DCL LMK
Parallel	Yes	1.40	–	1.23	–
Crossing	Yes	2.32	1.61	1.81	–
Tunnel	No	7.78	1.30	–	–

Table 5.3: Summary of experimental RMS position errors, in meters

The realized improvements from adding the UWB ranging measurements was lower than predicted in the simulations. It is expected that this is related to the certainty of data coming from the VN-300 and PACMod being underestimated, leading to much higher location accuracy than simulated. As such, there percent improvement from adding collaborative measurements was lower than in the simulations, with the smallest percent improvement being 9.2% and the largest being 12.1%. Similarly when adding landmarks, the improvement was lower being only 30.6%. However, removing the GNSS updates showed results much more in line with the simulation, with the addition of UWB measurements in into the EKF showing a 83.3% improvement in localization accuracy in a tunnel environment.

5.6 Future work

As discussed in previous sections, there are a number of possible areas of exploration introduced by this work. The largest area of future work is the expansion of the number of moving vehicles and UWB ranging modules used in the experiments. This would allow for a more realistic

evaluation of the DCL equations and experiments that utilize both modes of update measurements. Additionally, this would allow experiments to occur over larger areas for more realistic scenarios.

Additionally, it is expected that improvements in the EKF tuning could lead to greater performance gains in the position accuracy of the vehicle. As such, larger sample sets will be collected to aid in the finer tuning of the EKF equations.

In order to expand the operating area of autonomous vehicles, it is desirable to testing transitional spaces where GNSS signals are lost to further evaluate the benefits of using collaborative and landmark localization.

Finally, as the UWB measurements while mounted on the vehicle did not match the performance seen in open environments, it is necessary to re-characterize the modules *in situ* in order to verify the residuals are samples of a normal distribution. Additionally tests on the limits of measurement and standard deviation of the residuals should be performed while the sensors are mounted on the vehicles to ensure the numbers used in simulation are accurate.

6. SUMMARY AND CONCLUSIONS

In conclusion this thesis summarizes the work performed in the development and testing a collaborative localization algorithm simulation environment with experimental validations. In the developed simulation environment, multi-vehicle scenarios are testable with various sensor combinations and configurations. These scenarios provide repeatable and realistic environments to test various algorithms for collaborative localization and filtering. Through the use of various objects, the simulation emulates the networking required for collaborative localization and serves as a platform for evaluating algorithm performance using Monte Carlo analysis. Additionally, animations of the environments, vehicles, and estimates are producible in real time for aided evaluation.

Sensing models were created for both the GNSS and UWB modules. The GNSS sensor model was developed to accept CEP values, which is a common performance metric for receivers, and transform the value using a circularly symmetric Rayleigh distribution. The UWB ranging residuals were shown to follow a normal distribution using a normal probability plot and a Jarque-Bera test. This satisfies the assumptions made in using Kalman filters.

Monte-Carlo simulations were run using a number of situations and vehicles to test the efficacy of UWB sensors in decentralized collaborative localization as well as landmark measurements within an extended Kalman filter. By using 5,000 runs for each environment, it was possible to evaluate the expected distribution of errors in each environment with various frameworks and filters. Improvements were shown in all simulated environments, with landmarks offering additional improvements to collaborative localization, and with the most significant accuracy improvements seen in GNSS-denied environments.

To validate the simulation results, physical experiments were run using a by-wire GEM e6 from Autonomous Stuff in an urban environment in both collaborative and landmark setups. Due to higher than expected INS certainty, adding UWB measurements showed smaller improvements than simulations. Improvements of 9.2 to 12.1% were shown using UWB in collaborative localization environments. Improvements of 30.6 to 83.3% were shown in using UWB ranging

measurements to landmarks in an extended kalman filter for street crossing and tunnel environments respectively. These results are similar to the simulated data, and are promising in showing the efficacy of adding UWB ranging sensors to cars for collaborative and landmark localization, especially in GNSS-denied environments.

In the future, additional moving vehicles with additional tags will be tested in a greater number of environments and for larger distances. Transitional spaces from GNSS to GNSS-denied will be used to further test the algorithms. Lastly, further evaluations of the UWB ranging modules will be performed *in situ* to ensure earlier characteristics were valid.

REFERENCES

[1] K. Bimbraw, "Autonomous cars: Past, present and future - a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings*, vol. 1, pp. 191–198, 01 2015.

[2] G. Balamurugan, V. Jayaraman, and D. V. Naidu, "Survey on uav navigation in gps denied environments," in *2016 International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 198–204, 10 2016.

[3] Federal Communication Commission, *Characteristics of ultra-wideband technology*, 2006.

[4] M. Aftanas, J. Rovnakova, M. Drutarovsky, and D. Kocur, "Efficient method of toa estimation for through wall imaging by uwb radar," in *2008 IEEE International Conference on Ultra-Wideband*, vol. 2, pp. 101–104, 2008.

[5] Y. Zhou, C. Law, and J. Xia, "Ultra low-power uwb-rfid system for precise location-aware applications," in *Wireless Communications and Networking Conference Workshops (WCNCW)*, 04 2012.

[6] DecaWave, *EVK1000 user Manual*, 2016.

[7] G. Fan and Y. Sun, "An improved ins/pdr/uwb integrated positioning method for indoor foot-mounted pedestrians," *Sensor Review*, vol. 39, 11 2018.

[8] S. Sun, J. Hu, J. Li, R. Liu, M. Shu, and Y. Yang, "An ins-uwb based collision avoidance system for agv," *Algorithms*, vol. 12, p. 40, 02 2019.

[9] L. Yao, Y. A. Wu, L. Yao, and Z. Z. Liao, "An integrated imu and uwb sensor based indoor positioning system," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2017.

[10] J. Liu, J. Pu, L. Sun, and Z. He, "An approach to robust ins/uwb integrated positioning for autonomous indoor mobile robots," *Sensors*, vol. 19, p. 950, 02 2019.

[11] A. Benini, A. Mancini, and S. Longhi, "An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15.4a wireless sensor network," *Journal of Intelligent & Robotic Systems*, vol. 70, 04 2013.

[12] L. Qiang, W. Heng, L. Huican, and Z. Ying, "Design and implementation of multi robot research platform based on uwb," in *29th Chinese Control And Decision Conference (CCDC)*, pp. 7246–7251, 05 2017.

[13] W. Guosheng, Q. Shuqi, L. Qiang, W. Heng, L. Huican, and L. Bing, "Uwb and imu system fusion for indoor navigation," in *37th Chinese Control Conference (CCC)*, pp. 4946–4950, 07 2018.

[14] K. Dierenbach, S. Ostrowski, G. Jóźków, C. Toth, D. Grejner-Brzezinska, and Z. Koppanyi, "Uwb for navigation in gnss compromised environments," in *28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*, 08 2015.

[15] E. Ghanem, K. O'Keefe, and R. Klukas, "Estimating vehicle-to-vehicle relative position and attitude using multiple uwb ranges," *International Conference on Indoor Positioning and Indoor Navigation - Work-in-Progress Papers*, 2019.

[16] J. S. Martín, A. Cortés, L. Zamora-Cadenas, and B. J. Svensson, "Precise positioning of autonomous vehicles combining uwb ranging estimations with on-board sensors," *Journal of Intelligent & Robotic Systems*, vol. 9, p. 1238, 09 2020.

[17] J. Wang, Y. Gao, Z. Li, X. Meng, and C. Hancock, "A tightly-coupled gps/ins/uwb cooperative positioning sensors system supported by v2i communication," *Sensors*, vol. 16, 06 2016.

[18] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1250–1257, 1994.

[19] A. Howard, M. J. Matark, and G. S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 434–439 vol.1, 2002.

[20] B. E. Nemsick, A. D. Buchan, A. Nagabandi, R. S. Fearing, and A. Zakhor, "Cooperative inchworm localization with a low cost team," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6323–6330, 2017.

[21] A. I. Mourikis and S. I. Roumeliotis, "Optimal sensor scheduling for resource-constrained localization of mobile robot formations," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 917–931, 2006.

[22] P. Schmuck and M. Chli, "Multi-uav collaborative monocular slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3863–3870, 2017.

[23] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[24] F. R. Fabresse, F. Caballero, and A. Ollero, "Decentralized simultaneous localization and mapping for multiple aerial vehicles using range-only sensors," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6408–6414, 2015.

[25] T. R. Wanasinghe, G. K. I. Mann, and R. G. Gosine, "Decentralized cooperative localization for heterogeneous multi-robot system using split covariance intersection filter," in *2014 Canadian Conference on Computer and Robot Vision*, pp. 167–174, 2014.

[26] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *2009 IEEE International Conference on Robotics and Automation*, pp. 1402–1409, 2009.

[27] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Putting the 'i' in 'team': an ego-centric approach to cooperative localization," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 1, pp. 868–874 vol.1, 2003.

[28] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018.

[29] S. Tanwar and G. Gao, "Decentralized collaborative localization with deep gps coupling for uavs," in *2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 - Proceedings*, 2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 - Proceedings, (United States), pp. 767–774, Institute of Electrical and Electronics Engineers Inc., June 2018. 2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 ; Conference date: 23-04-2018 Through 26-04-2018.

[30] H. Badis and A. Rachedi, *Modeling and Simulation of Computer Networks and Systems*, pp. 653–682. London: Bantam, 2015.

[31] S. u. Rehman, M. A. Khan, T. A. Zia, and L. Zheng, "Vehicular ad-hoc networks (vanets) - an overview and challenges," *Journal of Wireless Networking and Communications*, vol. 3, no. 3, pp. 29–38, 2013.

[32] IEEE Standards Association, "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," Standard IEEE 802.11p-2010, Institute of Electrical and Electronics Engineers, 2010.

[33] D. L. Guidoni, G. Maia, F. S. H. Souza, L. A. Villas, and A. A. F. Loureiro, "Vehicular traffic management based on traffic engineering for vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 45167–45183, 2020.

[34] N. C. C. Roque, A. M. D. Guico, E. F. C. Mondia, A. L. P. Garing, J. M. B. Rocamora, and E. A. Roxas, "Analysis of traffic information dissemination using vehicular ad hoc network simulations," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 1035–1038, 2016.

[35] A. A. P. and S. J., "Dynamic route scheduler in vehicular ad hoc network for smart crowd control," *Personal and Ubiquitous Computing*, vol. 24, no. 4, p. 441–449, 2020.

[36] F. B. Günay, E. Öztürk, T. Çavdar, Y. S. Hanay, and A. u. R. Khan, "Vehicular ad hoc network (vanet) localization techniques: A survey," *Archives of Computational Methods in Engineering*, 2020.

[37] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on vanet security challenges and possible cryptographic solutions," *Vehicular Communications*, 2014.

[38] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, "Uav-assisted supporting services connectivity in urban vanets," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3944–3951, 2019.

[39] S. Bhamidipati and G. X. Gao, "Distributed cooperative slam-based integrity monitoring via a network of receivers," *ION GNSS+*, pp. 2023 – 2034, 2019.

[40] Decawave, "Trek1000 software package: Decarangertls pc application software." https://www.decawave.com/software/, 2020. ver. 3.8.

[41] C. M. Jarque and A. K. Bera, "A test for normality of observations and regression residuals," *International Statistical Review / Revue Internationale De Statistique*, vol. 55, no. 2, pp. 163–172, 1987.

[42] K. M. Lynch and F. C. Park, "Modern robotics: Mechanics, planning, and control," *Cambridge University Press*, 2017.

[43] N. Metropolis and S. Ulam, "The monte carlo method," *The Monte Carlo Method*, vol. 44, no. 247, pp. 335–341, 1949.

[44] u blox, "Zed-f9p module: u-blox f9 high precision gnss module." https://www.u-blox.com/en/product/zed-f9p-module, 2020.

[45] dalymople, "ubxtranslator." https://github.com/dalymople/ubxtranslator, 2020. v0.1-beta.

[46] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[47] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey, *Graphical methods for data analysis*. Belmont, Calif. : Wadsworth International Group ; Boston : Duxbury Press, 1983.

[48] S. I. Roumeliotis and G. A. Bekey, “Distributed multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[49] F. van Diggelen, “Gps accuracy: Lies, damn lies, and statistics,” *GPS World*, vol. 9, January 1998.

[50] J. Hartzer, “decawave_ros: a ros python package for interfacing with the decawave trek1000 modules.” https://github.com/unmannedlab/decawave_ros, 2020. v0.1-beta.

[51] J. Hartzer, “ubxtranslator: a ros python package for interfacing with ubx gps modules.” https://github.com/unmannedlab/ubxtranslator, 2020. v0.1-beta.

[52] VectorNav, “Vn-300 dual gnss/ins.” https://www.vectornav.com/products/vn-300, 2006.

[53] ArduSimple, “simplertk2b.” https://www.ardusimple.com/product/simplertk2b/, 2020.

[54] Autonomous Stuff, “Astuff pacmod.” https://autonomoustuff.com/product/pacmod/, 2020.

APPENDIX A

ALGORITHMS

Algorithm 1: Motion Update for Robot *i*

input : $\hat{X}_i^t,\ \Sigma_{ii}^t,\ \{\sigma_{ij}^t\}_{j\in 1,\ldots,N},\ U$
output: $\hat{X}_i^{t+1},\ \Sigma_{ii}^{t+1},\ \{\sigma_{ij}^{t+1}\}_{j\in 1,\ldots,N}$
$G = \frac{\partial g(X,U)}{\partial X}\left(\hat{X}_i^t, U\right)$
$\hat{X}_i^{t+1} = g(\hat{X}_i^t, U)$
$\Sigma_{ii}^{t+1} = G\Sigma_{ii}^t G^T + R$
for $1 \leq j \leq N, j \neq i$ **do**
  $\sigma_{ij}^{t+1} = G_i^t(x_i^t)\sigma_{ij}^t$;

Algorithm 2: Private Measurement Update for Robot *i*

input : $\hat{X}_i^t,\ \Sigma_{ii}^t,\ \{\sigma_{ij}^t\}_{j\in 1,\ldots,N},\ z$
output: $\hat{X}_i^{t+1},\ \Sigma_{ii}^{t+1},\ \{\sigma_{ij}^{t+1}\}_{j\in 1,\ldots,N}$
$H = \frac{\partial h(X)}{\partial X}\left(\hat{X}_i^t\right)$
$S = H\Sigma_{ii}^t H^T + Q\ K_i = \Sigma_{ii}^t H^T S^{-1}$
$\hat{X}_i^{t+1} = \hat{X}_i^t + K_i\left[z - h\left(\hat{X}_i^t\right)\right]$
$\Sigma_{ii}^{t+1} = (I - K_i H)\,\Sigma_{ii}^t$
for $1 \leq j \leq N, j \neq i$ **do**
  $\sigma_{ij}^{t+1} = (I - K_i H)\,\sigma_{ij}^t$

Algorithm 3: Relative Measurement Update for Robot *i* measuring Robot *j*

input : $\hat{X}_i^t,\ \Sigma_{ii}^t,\ \{\sigma_{ij}^t\}_{j\in 1,\dots,N},\ r$

output: $\hat{X}_i^{t+1},\ \Sigma_{ii}^{t+1},\ \{\sigma_{ij}^{t+1}\}_{j\in 1,\dots,N}$

$\Sigma_{ij}^t = \sigma_{ij}^t \begin{pmatrix} t \\ ji \end{pmatrix}^T$

$\Sigma_{aa}^t = \begin{bmatrix} \Sigma_{ii}^t & \Sigma_{ij}^t \\ \left(\Sigma_{ij}^t\right)^t & \Sigma_{jj}^t \end{bmatrix}$

$F_a = \left[\frac{\partial f(X_i,X_j)}{\partial X_i}\left(\hat{X}_i^t, \hat{X}_j^t\right), \frac{\partial f(X_i,X_j)}{\partial X_j}\left(\hat{X}_i^t, \hat{X}_j^t\right),\right]$

$S_a = F_a \Sigma_{aa}^t F_a^T + Q$

$K_a = \Sigma_{aa}^t F_a^T S_a^{-1}$

$\begin{bmatrix} \hat{X}_i^{t+1} \\ \hat{X}_j^{t+1} \end{bmatrix} = \begin{bmatrix} \hat{X}_i^t \\ \hat{X}_j^t \end{bmatrix} + K_a \left[r - f\left(\hat{X}_i^t, \hat{X}_j^t\right)\right]$

$\begin{bmatrix} \Sigma_{ii}^{t+1} & \Sigma_{ij}^{t+1} \\ \left(\Sigma_{ji}^{t+1}\right)^t & \Sigma_{jj}^{t+1} \end{bmatrix} = (I - K_a F_a)\Sigma_{aa}^t$

$\sigma_{ij}^{t+1} = \Sigma_{ij}^{t+1}\, \sigma_{ji}^{t+1} = I$

for $1 \le k \le N, k \ne \{i, j\}$ **do**

  $\sigma_{ik}^{t+1} = \Sigma_{ii}^{t+1} \left(\Sigma_{ii}^t\right)^{-1} \sigma_{ik}^t$

Algorithm 4: Monte Carlo Simulation Algorithm

Result: Matrix of simulated estimation errors for cars

for $j \in \{1, ..., nRuns\}$ **do**

  initialize cars without UWB;

  while *t < nSteps* **do**

    for $i \in \{1, ..., nCars\}$ **do**

      $x_i^t \leftarrow$ carStep(x_i^{t-1});

      $\hat{x}_i^t \leftarrow$ carFilter($\hat{x}_i^{t-1}, Z$);

    calculate RMSE;

    t++;

  initialize cars and landmarks with UWB;

  while *t < nSteps* **do**

    for $i \in \{1, ..., nCars\}$ **do**

      $x_i^t \leftarrow$ carStep(x_i^{t-1});

      $\hat{x}_i^t \leftarrow$ carFilter($\hat{x}_i^{t-1}, Z$);

    calculate RMSE;

    t++;

APPENDIX B

EXPERIMENTAL OFFSETS

In order to properly use the various sensor data, coordinate frame transformations were used to keep all data in UTM zone 14S and with respect to the VectorNav center of the vehicle. The offsets for the RTK GNSS antennae and the UWB tags are geometrically shown with respect of the center of rotation of the vehicle in Figure B.1.

The locations of each non-moving tag during the experiments are shown in Table B.1. These coordinates are in UTM, which allows for easier processing of data.

Experiment	Tag	Easting	Northing	Z
Parallel	0	751574.23	3396401.47	1.27
	1	751574.53	3396405.02	1.17
Crossing	0	751591.54	3396403.54	1.27
	1	751572.08	3396386.66	1.17
Tunnel	0	751574.23	3396401.47	1.27
	1	751574.23	3396356.33	1.17

Table B.1: UTM coordinates in zone 47H of tags in each experiment

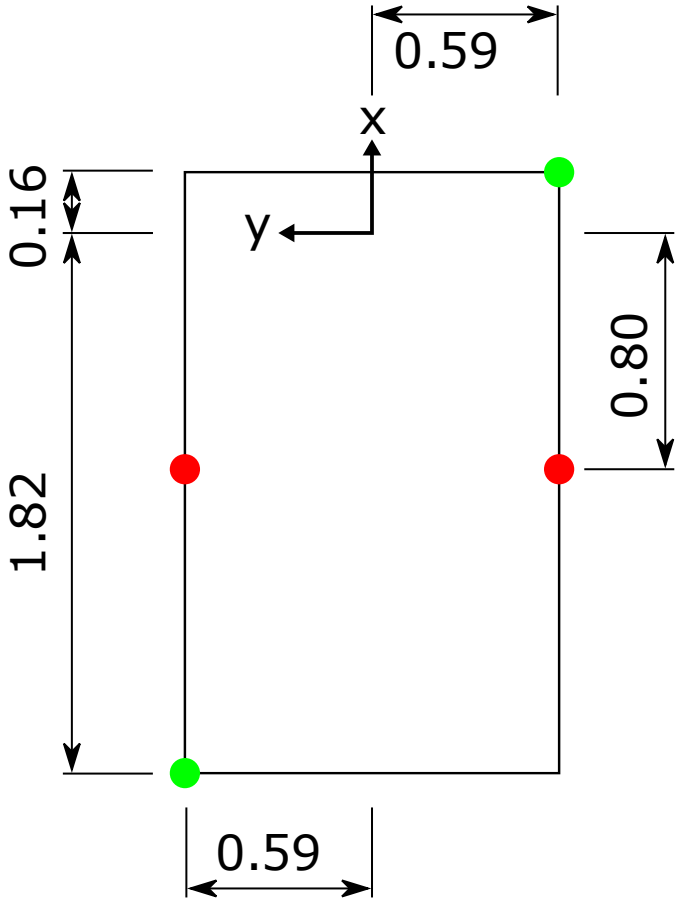


Figure B.1: Sensor Coordinate Offsets- UWB tags are green, RTK GNSS receivers are red