VEHICLE LOCALIZATION USING PASSIVE AND ACTIVE INFRASTRUCTURE ENABLE AUTONOMY IN GPS DENIED ENVIRONMENTS

A Thesis

by

ANDREW WEAVER

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,	Swaminathan Gopalswamy
Committee Members,	Sivakumar Rathinam
	Sue Chrysler
Head of Department,	Bryan Rasmussen

May 2021

Major Subject: Mechanical Engineering

ABSTRACT

Autonomous vehicles (AV) have the potential to provide dividends for society in areas such as alleviating congestion, reducing emissions, and increasing safety. Fundamentally, an AV makes decisions based on its surroundings and where it is located in the environment which makes localization the most critical function of an AV. The majority of autonomous transportation solutions have become highly reliant on the availability of a Global Positioning System (GPS) for localization. However, GPS is susceptible to poor signal reception in areas such as urban canyons as well as its vulnerability to malicious attacks. This research approaches solving the issue of GPS reliance by two methods — passive and active infrastructure enabled autonomy (IEA). The basis of IEA is to enable AV functions by offloading some of the computational cost to the infrastructure. The passive IEA method leverages SmartCodes (SC) in the environment where each SC has GPS information embedded in its digital signature that must be read with a camera. A radar and on-board vehicle sensors were also used and fed into an Extended Kalman Filter to localize the vehicle. The active IEA method leverages a multi-smart-sensor pack that is outfitted with road-side units that hold cameras and a radar that provide localization information to the vehicle. All data is fed into a multi-target tracking version of the linear Kalman Filter. Each method was experimentally tested and both passive and active IEA were capable of localizing a vehicle globally. The passive IEA method was able to localize a vehicle with longitudinal and lateral root-mean-square errors (RMSE) less than 0.39m and 0.18m, respectively. The active IEA method localized an average longitudinal RMSE of 4.54m and an average lateral RMSE error of 0.46m.

DEDICATION

To my mom, my dad, my sister and my grandparents:

Without whom I would not be where I am today.

ACKNOWLEDGMENTS

First and foremost, I would to extend my sincere gratitude to Dr. Swaminathan Gopalswamy for advising me and creating an environment in which I could succeed. I would also like to thank Texas A&M University for being my home away from home — Texas A&M University will always be apart of me.

I would like to thank the members of 3M: Travis Potts, Ken Smith, and Matthew Messina for providing consistent weekly feedback on the passive IEA research.

I am extremely grateful for Samin Moosavi for being an amazing research partner over the past two years. I would also like to thank all the members of the CAST lab at Texas A&M who have help me grow throughout my two years in the lab, and last but not least, a special thank you to all the friends I have made along the way that have made my time at A&M a memorable one.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Dr. Swaminathan Gopalswamy (advisor) and Dr. Sivakumar Rathinam of the Department of Mechanical Engineering and Dr. Sue Chrysler of the Department of Industrial and Systems Engineering.

The Extended Kalman Filter described in Chapter 2.3.1 was primarily written and tuned by Samin Moosavi. Additionally, 3M provided consistent weekly feedback and suggestions for the Passive Infrastructure Enabled Autonomy Experiments.

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

Funding Sources

Graduate study was supported by a fellowship from 3M Company.

NOMENCLATURE

AV	Autonomous Vehicle
IEA	Infrastructure Enabled Autonomy
GPS	Global Positioning System
IMU	Inertial Measurement Unit
RF	Radio-Frequency
RSU	Road-Side Unit
RSS	Return Signal Strength
PDA	Probabilistic Data Association
JPDA	Joint Probabilistic Data Association
NN	Nearest Neighbors
KF	Kalman Filter
EKF	Extended Kalman Filter
FMCW	Frequency Modulated Continuous Waveform
IF	Intermediate Frequency
MTT	Multi-Target Tracking
DBW	Drive-By-Wire
RMSE	Root Mean-Square Error
MSSP	Multi-Smart Sensor Pack
ML	Machine Learning
PnP	Perspective n-Point
CP	Control Point
RR	Retro-Reflector

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

CONTRIBUTORS AND FUNDING SOURCES v

NOMENCLATURE vi

TABLE OF CONTENTS vii

LIST OF FIGURES ix

LIST OF TABLES xi

1. INTRODUCTION AND LITERATURE REVIEW 1

   1.1 Introduction 1
      1.1.1 Passive Infrastructure Enable Autonomy 2
      1.1.2 Active Infrastructure Enabled Autonomy 3
   1.2 Contributions 4
   1.3 Organization 4

2. SENSOR FUSION AND KALMAN FILTERS 5

   2.1 Overview 5
   2.2 Sensor Fusion 5
      2.2.1 Synchronization 7
      2.2.2 Data Association 8
   2.3 State Estimation 9
      2.3.1 Extended Kalman Filter 10
      2.3.2 The Bicycle Model 12
      2.3.3 The Constant Acceleration Model 15

3. PASSIVE INFRASTRUCTURE ENABLED AUTONOMY 16

   3.1 Proposed Method 16
      3.1.1 Architecture 16
      3.1.2 SmartCode Technology 17

3.1.3 Radar Technology 20
3.2 Experimental Methods 23
3.3 Results 25
3.4 Further Study 28

4. ACTIVE INFRASTRUCTURE ENABLED AUTONOMY 29

4.1 Proposed Method 29
4.1.1 Architecture 29
4.2 Cameras 30
4.2.1 Machine Learning 30
4.2.2 Camera Calibration 31
4.2.2.1 Control Point Collection 32
4.2.2.2 Calibration Algorithm 32
4.3 Radar Calibration 33
4.4 Experimental Methods 35
4.5 Preliminary Results 36
4.6 Further Study 40

5. CONCLUSIONS 41

REFERENCES 43

APPENDIX A. RADAR 46

A.1 Overview 46
A.2 Radar Detection Problems 46
A.3 Sign Retro-reflectivity 48

LIST OF FIGURES

FIGURE Page

2.1 Level 1 sensor fusion pipeline 7

2.2 Synchronizer Structure 8

2.3 Nearest Neighbor with multiple observations and objects 9

2.4 Graphical Kalman filter steps 12

2.5 Kinematic Bicycle Model 13

3.1 Passive IEA architecture 17

3.2 SmartCodes as seen through an IR filter (cropped) 18

3.3 Sign angle calculation diagram 19

3.4 Sign angle approximation (image is cropped) 20

3.5 Raw radar point cloud data 21

3.6 Radar point cloud with object tracking 22

3.7 Experimental Setup 24

3.8 Experiment Sensor Array 24

3.9 Lateral error 26

3.10 Longitudinal error 27

3.11 40mph lateral error with data points 27

3.12 40mph longitudinal error with data points 28

4.1 Active infrastructure enabled autonomy architecture 30

4.2 Output of Darknet and YOLOv3 31

4.3 Radar calibration schematic 34

4.4 Active infrastructure enabled autonomy experiment setup 36

4.5 Active IEA filter data 37

4.6 Active IEA filter error 38

4.7 Active IEA sensor data 39

4.8 Active IEA sensor data (multi lane) 39

A.1 Radar detection consistency data 46

A.2 Radar accuracy data 47

A.3 Radar velocity errors 47

A.4 Sign retro-reflector experimental data 48

LIST OF TABLES

TABLE Page

3.1 Passive IEA experimental RMSEs 26

4.1 Active IEA experimental RMSEs 36

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

As of 2018 there were 6,734,000 traffic accidents across the nation with 33,654 of those being fatal [1]. Additionally, from 1982 to 2017 traffic congestion has become increasingly more severe causing billions of dollars in losses in time and fuel [2]. The transportation sector also produced 28.2 percent of greenhouse gas emissions in 2018 [3]. The three aforementioned issues are massive problems in and of themselves, and autonomous vehicles (AV) could help reduce all three problems. The largest direct impact of AVs would be safety. According to a report from the University of Texas at Austin Center For Transportation Research, at 50% AV market penetration, there would be a reduction of approximately 50% of crashes, injuries, and fatalities from motor vehicles in Texas [4]. The same paper also estimated that significant congestion reduction starts at 10% market penetration, and amazingly, nearly half of all freeway congestion would be eliminated at 90% market penetration [4]. Consequently, with congestion significantly reduced, emissions would also decrease because of the less time vehicles would spend on the road. Not to mention, the majority of AVs are electric further reducing green house gas emissions. Although AVs are not a be-all and end-all solution to these problems, they show great potential in significantly impacting these issues positively.

Autonomous vehicles need to process enormous amounts of information to make reliable and safe decisions, but all that processing power comes at a cost — power requirements and cost to the consumer. There are many areas of research into AVs that alleviate these issues, but this paper will focus on a relatively new idea — Infrastructure Enabled Autonomy (IEA) [5]. As opposed to a vehicle handling all of the autonomous functions, in IEA, the infrastructure is leveraged to aid AV functions thereby reducing the computational cost on the AV or adding redundancy to the vehicle. In this research proposal, two different facets of IEA are explored: passive and active IEA. Passive IEA can be described as infrastructure, that by itself does not process data whereas in active IEA, the infrastructure processes data to support AV functions. There are various ways in which the infrastructure could aid an AV, from controlling the vehicle entirely to indicating pedestrian and vehicle locations to the vehicle. Fundamentally, AVs make decisions based on where they are, so perhaps the most critical AV function is localization. One of the fundamental requirements for current localization techniques is GPS, but how do you localize a vehicle if GPS is unavailable or unreliable? GPS signals, especially in urban areas, can be blocked by tunnels, buildings or bridges [6]. Additionally, GPS is vulnerable to spoofing or nefarious attacks [7]. For the aforementioned reasons, it is important for vehicles to have methods to handle situations when GPS signals are unreliable. The passive IEA method leverages SmartCodes[1][8] which are embedded into the infrastructure and contain information that allows a vehicle to localize itself, and the active IEA method leverages road side cameras and a radar to telemetry localization data to the vehicle. Both methods, developed independently, could be combined for better localization and redundancy.

1.1.1 Passive Infrastructure Enable Autonomy

In the case of passive IEA, several ideas have been research, but the majority rely on costly sensors. Perhaps the simplest attempt to solve this localization issue is to use inertial navigation, also known as dead reckoning [9]. Inertial navigation refers to the use of integrated inertial measurement unit (IMU) data to extrapolate the state of a system. This method, however, increases in error overtime which constricts the maximum time GPS could be unavailable. Another common method is cooperative communication between neighboring vehicles [10]. Here each vehicle can estimate its location based on the information it receives from other vehicles and its own sensors. At any point, if information is lost or delayed between vehicles, the error could propagate throughout the connected vehicle system adversely affecting their localization. Additionally, at what density of cooperative AVs would this method show significant benefit? Some methods attempt to leverage infrastructure in the environment using visual sensors such as cameras and LiDARs. Wetzel [11] for example, used traffic signs with known GPS locations as landmarks that a camera would detect

[1] 3M product

using traffic sign detection algorithms. This method had many problems in that sign uniqueness is an issue as well as a database of signs must be created to correlate each sign with a GPS location. Schaefer uses poles as landmarks in conjunction with LiDAR data to make localization adjustments [12]. This method also requires a previous 3D-LiDAR scan of the area of interest to function. Additionally, LiDARs are not ideal for the consumer at their current price. Another interesting method used passive radio-frequency transponders embedded in lane markings that a vehicle could pick up localize itself within a lane. [13]

Most of the methods mentioned rely on GPS signals as the main means of localization while the other sensors bridge the gap in the accuracy needed to localize a vehicle effectively. This research proposes a solution to solve the reliance on GPS signal in localization. The proposed method is able to localize the vehicle precisely in absence of GPS by only relying on SmartCode technology, a low cost radar, and low resolution built-in vehicle sensors.

1.1.2 Active Infrastructure Enabled Autonomy

In principle, the active IEA method serves to solve the same localization problem as the passive IEA case, but the scope of its impact is much larger. The proposed method for the active IEA case lays the ground work to fundamentally change how AV are controlled in the future [5]. The proposed method leverages cameras and radars on the infrastructure to localize a vehicle. Previous research in IEA used independent sensor arrays connected to the vehicle wirelessly using dedicated short-range communication to telemetry localization information to the vehicle [14]. Each sensor array, mounted on traffic light poles, used a camera which fed its data into a machine learning algorithm and a lane detection algorithm to provide localization information to the vehicle. Additionally, sensors from the vehicle where also used in their localization algorithm. Other research has been done on IEA under the guise of other names, but the principles remain the same. Most research has solutions utilizing road-side units (RSU) which are structures that transmits signals to the vehicles. Gathering information from these RSUs are based on received signal strength (RSS), time-of-flight, time difference of arrival, or angle-of-attack methods. Hernandez [15] was able to achieve 6-10m accuracy by utilizing RSS from the WiFi network on a college campus. Similarly, Ang [16] achieved 30m accuracy by using RSS from WiFi in conjunction with a database of WiFi fingerprints in the area. Both methods do not provide accurate enough results for autonomous functions. Another method using RSUs was presented in [17] which has the vehicle send its GPS information to the RSU which corrects the measurement and transmits it back to the vehicle. Submeter results where achieved, but the reliance on GPS still exists. Kloeden [18] used cooperative RF landmarks which respond to a vehicles request for information. Using angle-of-attack and time-of-flight estimations he was able to achieve less than 0.3m of error on average.

Some of these methods show impressive results, however, all of these methods do not provide ways to track both a vehicle connected to the infrastructure and vehicles that are not. This research proposes a solution solve vehicle localization without GPS signals for both AV, non AVs, and other moving roadway traffic.

1.2 Contributions

This research was primarily a result of the efforts of myself and Samin Moosavi. In the passive IEA research while the main focus of my research was the radar implementation, and Samin's main focus was the extended kalman filter, our work overlapped in the sense that we provided help anytime it was needed. Additionally, 3M provided helpful feedback on the passive IEA research. For the active IEA case, the research has been a joint venture in all aspects of the research.

1.3 Organization

Chapter 2 will overview a common and reoccurring theme among both proposed methods — sensor fusion. Additionally, it will discuss the methods within sensor fusion that were used in this research. Chapters 3 and 4 will discuss the sensors, algorithms, architecture, and experiments of the proposed methods of Passive IEA and Active IEA, respectively. Chapter 5 wraps up the thesis with an overview an comparison of both research topics.

2. SENSOR FUSION AND KALMAN FILTERS

2.1 Overview

In this chapter, an overview of sensor fusion and state estimation will be discussed. The basic theory of each principle will be discussed as well as the mathematics behind them. This section precedes both proposed methods because of its relevance to each research topic. Furthermore, this section will also be heavily referenced in subsequent chapters.

2.2 Sensor Fusion

Sensor fusion, or data fusion, has many definitions, but perhaps the most specific definition for this research was defined by Hans [19] : "data fusion techniques combine data from multiple sensors and related information from associated databases to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone." In layman's terms, sensor fusion is the process of combining sources of data to produce more accurate results than using a single source of data. For both proposed methods in this paper, sensor fusion is one of the foundational algorithms that make each method function.

Sensor fusion has been researched exhaustively and there are many classifications of data that exits. The simplest example is the data classifications proposed by Durrant-Whyte [20]:

1. Competitive — Where two or more sensors measure the same state of an object

2. Complementary — Where two or more independent sensors measure states on the same object

3. Cooperative — Where one sensor relies on another to gather information

This data classification is simple and many more exist, but the classifications proposed by Durrant-Whyt fit both localization methods in this thesis well. These data classifications give insight into how the sensor fusion algorithm should work based on the data produced by the sensors used in a system. In the case of an AV, sensors such as IMUs, GPSs, and wheel speed sensors all measure the states of the vehicle. The data from these sensor can then be fused using an algorithm under the complementary classification. If an AV is being tracked by infrastructure cameras, radars, or LiDARs, these measurements can be both complementary or competitive because they provide redundant information and they complement each other if their field of view overlap.

Outside of data classification there are four important processes in sensor fusion that exist [19]:

Level 1: Object Refinement — align, associate, estimate, and classify objects

Level 2: Situation Refinement — determines relationship between objects and environment

Level 3: Threat Refinement — uses current information to determine relationship between objects and threats

Level 4: Process Refinement — uses information of previous levels to redirect and optimize processes to meet goal

Each level has significant importance to the overall goal of full autonomy. They each perform an important roll, but this thesis focuses specifically on object refinement. Within the level 1 process several areas of importance exist: data synchronization, data association, object state estimation, and object classification. A high-level representation of the level 1 sensor fusion pipeline is shown in figure 2.1. Each piece of data from sensors is sent through a synchronization process, followed by data association, and finally object state estimation. Classification was not included in the pipeline because of the methods that will be described in later chapters. In subsequent sections each process will be discussed in detail and the methods used in this thesis will be explained.

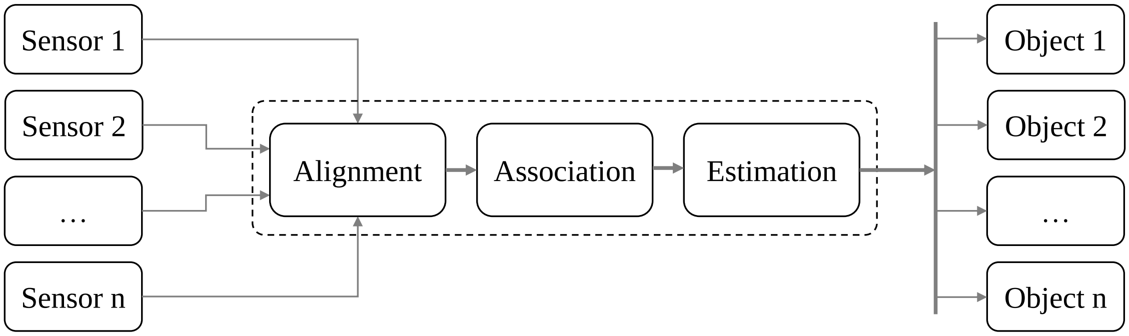


Figure 2.1: Level 1 sensor fusion pipeline

2.2.1 Synchronization

Data synchronization has the potential to have massive ramifications on system performance. Sensor data, unless analog, is discrete and arrives in discrete time intervals. In an ideal circumstance all sensor data arrives at the same time; however, in practice this is never true. When multiple sensors are running on a system there is no guarantee that every sensor will be started at the exact same time, and compounding the issue, each sensor samples data at different rates. To combat this issue, data synchronization will be performed prior to sending the information through the sensor fusion pipeline. Figure 2.2 shows a visual representation of data synchronization. By synchronizing data prior to sensor fusion, the sensor fusion algorithm does not need to be updated as frequently. As an example, if an IMU is sampled every 100Hz, and a GPS is sampled every 10Hz, the IMU data can be synchronized to the GPS sample rate. This effectively sends a packet of data to the sensor fusion algorithm with GPS data and the most up to date IMU data. When computation costs must be budgeted synchronizing data as mentioned reduces the computational load.

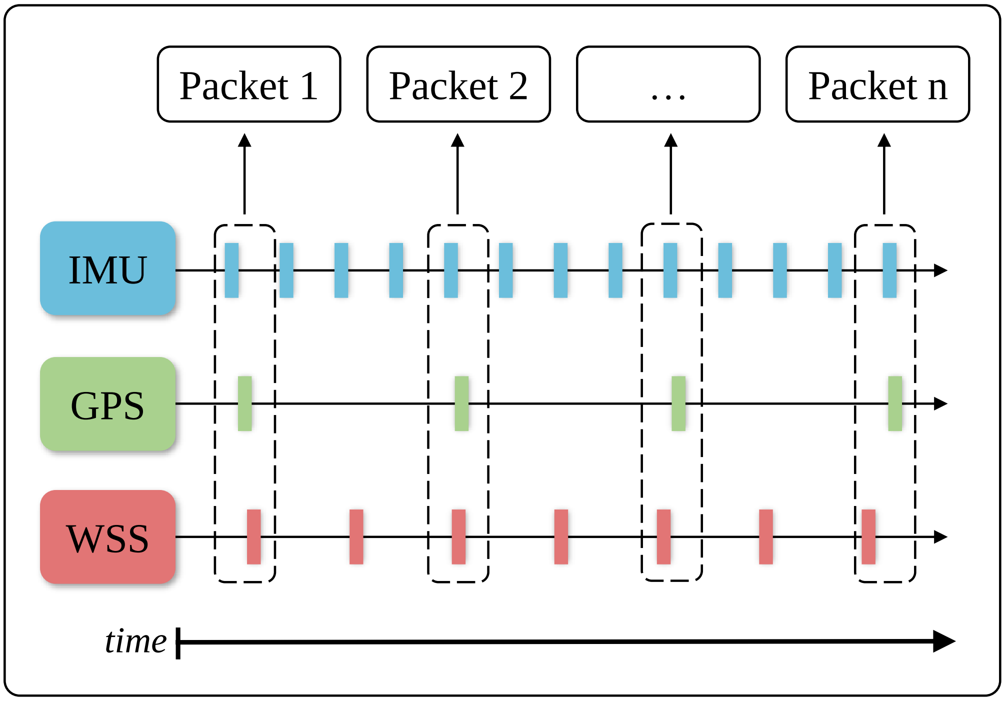


Figure 2.2: Synchronizer Structure

2.2.2 Data Association

When sensors like cameras, radars, and LiDARS get data measurements from multiple objects it becomes ambiguous which data came from what source. This issue becomes an important issue when tracking multiple objects, for example. Algorithms which pinpoint where data originates from is called data association. Various methods exist such as probabilistic data association (PDA), joint probabilistic data association (JPDA), or nearest neighbors (NN) [21]. Although the NN algorithm is the least robust, it is quick to associate data. For the proposed methods, because they are proof of concepts and noise from the environment is limited, the NN algorithm is sufficient. The NN algorithm compares an observation's distance (a metric defined by the user that is symmetric and satisfies the triangle inequality) from an object, and if the distance is within a threshold, the data is then associated with that object. For the research in this thesis, the Mahalanobis distance was chosen for increased robustness instead of a standard euclidean distance or other distance metrics. A graphical representation of a modification to the NN algorithm can seen in figure 2.3.

Consider checking the statistical distance of an observation from an objects multi-dimensional Gaussian distribution. Given a residual vector, $y_j = [y_1, y_2, ..., y_n]$ as defined in eqn 2.7 and the objects arithmetic mean, $\bar{x}_i = [\bar{x}_1, \bar{x}_2, ..., \bar{x}_n]$, and innovation covariance, S_i, as defined in eqn. 2.6, the Mahalanobis distance, d_{ij}, is defined as:

$$d_{ij} = [(y_j - \bar{x}_i)^T S_i^{-1} (y_j - \bar{x}_i)]^{\frac{1}{2}} \tag{2.1}$$

Where the subscripts i and j represent the ith object and jth observation, respectively. Eqn. 2.1 will yield an i by j matrix of Mahalanobis distances. Under the assumption that only one object can have one observation, the point associated with each object would be $min(d_i)$.

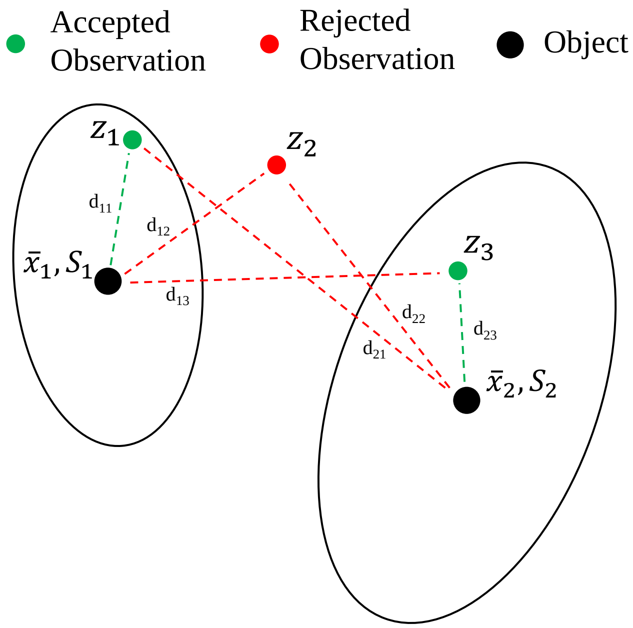


Figure 2.3: Nearest Neighbor with multiple observations and objects

2.3 State Estimation

Perhaps the most widely used sensor fusion algorithm is the Kalman Filter (KF) [22]. The KF is a stochastic linear state estimator which is minimum in the mean-square error sense. The

KF is so useful because it assumes the observations and model are imperfect with mean-zero Gaussian noise, and instead of calculating the state of the system, the KF calculates system states probabilisticly. For strictly linear systems, the standard linear KF is used, but the majority of systems are highly non-linear especially when pertaining to AVs. The KF has been extensively researched and methods have been developed to generalize the KF for non-linear systems. The most noteable non-linear KF is the extended kalman filter (EKF). Universal to all systems is the model defined generically as:

$$\dot{x} = f(x, u) + v \tag{2.2}$$

$$z = h(x) + w \tag{2.3}$$

Where the function, $f(x, u)$, is any function which relates the observable system states, x, and input, u, to the system derivatives. The system also has observations, z, ascertained from the measurement function, $h(x)$, and observable system states. The model includes mean-zero Gaussian noise in both the model and the measurements represented v, and w, respectively. The model used for passive IEA is the bicycle described in section 2.3.2 and the model used for active IEA is the constant acceleration model described in 2.3.3. In the next sections the EKF and KF will be discussed in detail.

2.3.1 Extended Kalman Filter

The basic KF algorithm can be seen in algorithm 1. Where:

- F — state transition matrix
- x — system states
- P — system covariance
- S — innovation covariance
- H — measurement transformation matrix
- K — the kalman gain
- y — sensor observation
- Q — process noise
- R — measurement variance
- NN — nearest neighbor algorithm

Algorithm 1 Linear Kalman Filter (KF)

```
• Initialization:
    i Use vehicle state assumptions and an initial measurement → $\hat{x}_{k-1}$
    ii Use assumptions based on the system to build an initial covariance matrix → $P_{k-1}$
Acquire sensor observations, $z_k$
Predict
    $\bar{x}_k = F\hat{x}_{k-1} + Bu_k$    (2.4)
    $\bar{P}_k = FP_{k-1}F^\top + Q$    (2.5)
Measurement
    $S_k = H\bar{P}_k H^\top + R$    (2.6)
    $y_k = z_k - H\bar{x}_k$    (2.7)
    $NN(\bar{x}_k, y_k, S_k)$    (2.8)
Update
    $K = \bar{P}_k H^\top S_k^{-1}$    (2.9)
    $\hat{x}_k = \bar{x}_k + Ky_k$    (2.10)
    $P_k = (I - KH)\bar{P}_k$    (2.11)
```

To generalize the linear KF to non-linear systems the model must be linearized about the current time step. This linearization can be calculated by finding the Jacobian of eqn. 2.2 followed by Eulers method:

$$F = \left.\frac{\partial f(x_{k-1}, u_k)}{\partial x}\right|_{x_{k-1}, u_k} \tag{2.12}$$

Next the predicted state, $\bar{x}_k$, can either be calculated by ODE methods like Runge-Kutta or calculated through the linearized state transition matrix. This thesis will use the latter for simplicity of implementation. After the prediction, the measurement matrix must also be linearized as follows:

$$H = \left.\frac{\partial h(\bar{x}_k)}{\partial x}\right|_{\bar{x}_k} \tag{2.13}$$

Similar to the predicted state, the predicted measurements, $H\bar{x}_k$, can be calculated using ODE solvers, but this thesis uses the straight forward matrix multiplication method. After these linearizations, the EKF algorithm does not differ from the linear KF. A graphical representation of the algorithm is shown figure 2.4. From a high level, step 1 takes the previous state estimate and projects it forward in time which also reduces the belief in the state as shown by the larger covariance. Step 2 takes the measurements and computes the residual from the predicted state and measurements. These residuals are compared to the innovation covariance using the nearest neighbor (NN) algorithm defined in section 2.2.2. Using the accepted observations from the NN algorithm, step 3 corrects the predicted state and covariance based on the residuals of the system and yields the next estimated state.

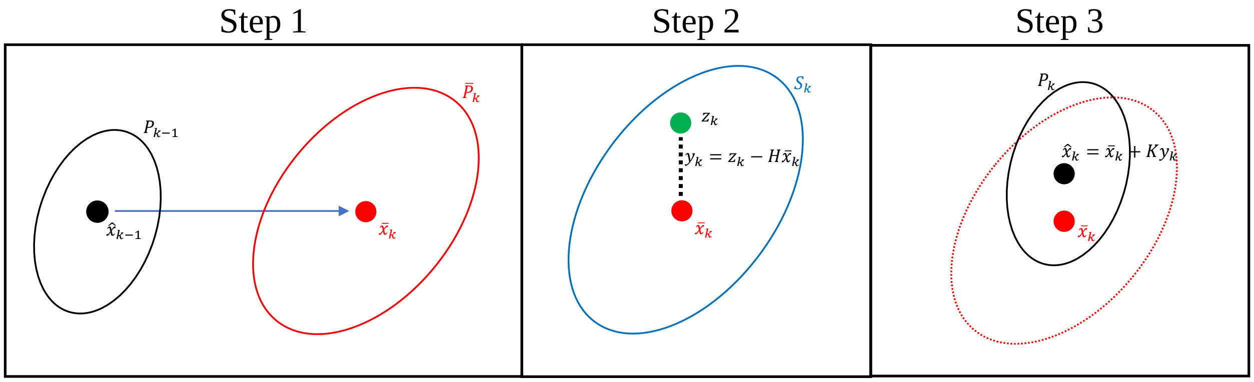


Figure 2.4: Graphical Kalman filter steps

2.3.2 The Bicycle Model

In the bicycle model, used in the passive IEA research and depicted in figure 2.5, a vehicle is modeled only with two wheels. This model holds well in circumstances with low steering angles and velocity because at small angles, both tires will have close to the same steering angle. In the experiments in this thesis, this is a valid assumption because all vehicles will have constant heading and only change lanes. In the figure each symbol is as follows:

- β — slip angle
- δ — steering angle (input)
- ψ — yaw angle
- C — center of mass (COM)
- l_r — distance from COM to rear wheel
- l_f — distance from COM to front wheel
- X, Y — inertial reference frame
- x', y' — vehicle reference frame

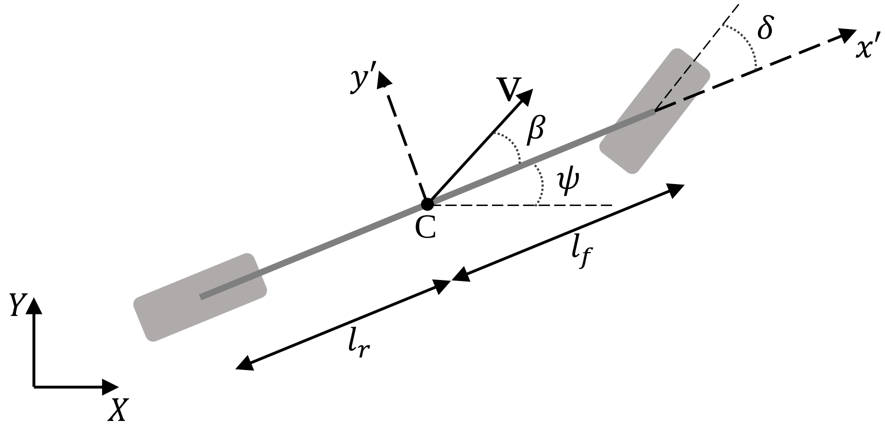


Figure 2.5: Kinematic Bicycle Model

The vehicle is modeled as the kinematic bicycle model [23]. Which is described in continuous time by the following equations:

$$\begin{aligned}
\dot{x} &= vcos(\psi + \beta) \\
\dot{y} &= vsin(\psi + \beta) \\
\dot{v} &= a \\
\dot{\psi} &= \frac{v}{l_r}sin(\beta) \\
\beta &= tan^-1(\frac{l_r}{l_f + l_r}tan(\delta)
\end{aligned} \tag{2.14}$$

Because state estimation acts in discrete time, the equations presented below are the bicycle model after applying a first order Euler's method after time Δt:

$$\begin{aligned} x_k &= x_{k-1} + v_{k-1} cos(\psi_{k-1} + \beta_k)\Delta t \\ y_k &= y_{k-1} + v_{k-1} sin(\psi_{k-1} + \beta_k)\Delta t \\ v_k &= v_{k-1} + a_{k-1}\Delta t \\ \psi_k &= \psi_{k-1} + \frac{v_{k-1}}{l_r} sin(\beta_k)\Delta t \\ \beta_k &= tan^{-1}\left(\frac{l_r}{l_f + l_r} tan(\delta_k)\right) \end{aligned} \tag{2.15}$$

Eqn. 2.15 is a non-linear function which requires the EKF. The Jacobian of eqn. 2.15 can be shown as:

$$F_k = \begin{bmatrix} 1 & 0 & cos(\psi_{k-1} + \beta_k)\Delta t & -vsin(\psi_{k-1} + \beta_k)\Delta t \\ 0 & 1 & sin(\psi_{k-1} + \beta_k)\Delta t & vcos(\psi_{k-1} + \beta_k)\Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & sin(\beta_k)\Delta t / l_r & 1 \end{bmatrix} \tag{2.16}$$

In the passive IEA chapter described later in the thesis, the process model was found to be related to the inputs more so than the state-transition function. Thus the process model can be described as:

$$Q_k = B\sigma_w B^\top \tag{2.17}$$

Where σ_w is defined as $diag(\sigma_a^2, \sigma_\delta^2)$ and B is defined as:

$$B = \frac{\partial f_k}{\partial u_k} = \begin{bmatrix} -v_{k-1} sin(\psi_{k-1} + \beta_k)\Delta t & 0 \\ v_{k-1} cos(\psi_{k-1} + \beta_k)\Delta t & 0 \\ 0 & \Delta t \\ v_{k-1} cos(\beta_k)\Delta t / l_r & 0 \end{bmatrix} \tag{2.18}$$

2.3.3 The Constant Acceleration Model

The constant acceleration model, which is used in the active IEA research, is used because the infrastructure does not have access to the vehicle states. The kinematic constant acceleration model in two dimensions can be shown as:

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \tag{2.19}$$

Where the $x = [x, \dot{x}, y, \dot{y}]^\top$. Similarly in the bicycle model section, a first order Euler method after time Δt is used to compute the equivalent discrete time system:

$$x_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} \tag{2.20}$$

In the constant acceleration model, the process noise can be easily defined by the piecewise white noise model:

$$Q = \Gamma \sigma_v^2 \Gamma^\top \tag{2.21}$$

Where σ_v is the noise of the velocity and Γ is shown below:

$$\Gamma = \begin{bmatrix} \frac{1}{2}\Delta t^2 & \Delta t & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix}^\top \tag{2.22}$$

3. PASSIVE INFRASTRUCTURE ENABLED AUTONOMY

3.1 Proposed Method

The proposed method for the passive infrastructure enabled autonomy (IEA) method leverages SmartCode (SC) technology to provide vehicles with ground truth information. SCs are invisible to the naked eye and provide no information by themselves, thus the passive distinction. In the proposed method, a camera with a IR bandpass filter and IR light source will be used to read the SCs in conjunction with a mmWave radar. The camera, radar, and SCs, in conjunction, provide a novel and accurate approach to vehicle localization in GPS denied environments. In the following sections, the passive IEA method will be discussed in detail.

3.1.1 Architecture

The architecture of passive IEA autonomy can be overviewed in figure 3.1. SCs will be placed in the environment in pre-determined locations. The camera's SC observations can be directly sent to the sensor fusion algorithm. The radar, however, has no means to objectively define SCs, so the radar's observations are sent through the NN algorithm before being fed into the KF. All other sensors used pertain to measuring the states of the vehicle. These sensors publish data at a relatively high rate, and therefore will be sent through a synchronization process before being sent to the EKF.

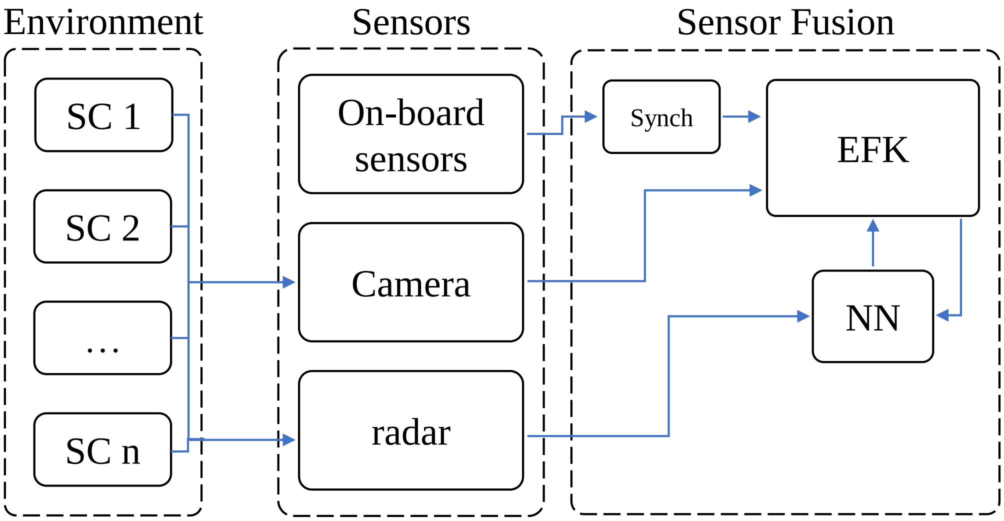


Figure 3.1: Passive IEA architecture

3.1.2 SmartCode Technology

Because the foundation of the proposed method is based SC technology, this section will describe its function. SC refers to technology where a specialized film that is retro-reflective in the IR spectrum (940nm) carries encoded information. In particular, the SC will have, among other pieces of information, the global coordinates of the SC's location. By placing SC signs throughout highways, cities, and towns, vehicles could always have a reference to ground truth regardless of their location provided that SC technology is implemented. SCs are not limited to signs, so they can be implemented on any piece of infrastructure. This research focuses on signs, so further mention of signs refers to signs with SCs applied to them.

A lot of research has been done on traffic sign detection, and they are typically done using machine learning or advanced edge detection algorithms [24][25]. The issue with these methods are the large amount of signs that exist, the vast amount of environmental conditions, and the clutter in the visible light spectrum. One unique advantage of SCs is they are designed to be visible in the IR spectrum allowing for easy detection. Thus, eliminating a large amount of visual clutter. Figure 3.2 shows SCs as seen through the camera. Additionally, SCs both classify the type of sign and provide GPS data. As a result, SCs virtually eliminate the need for computationally expensive detection algorithms while simultaneously providing significantly more information.

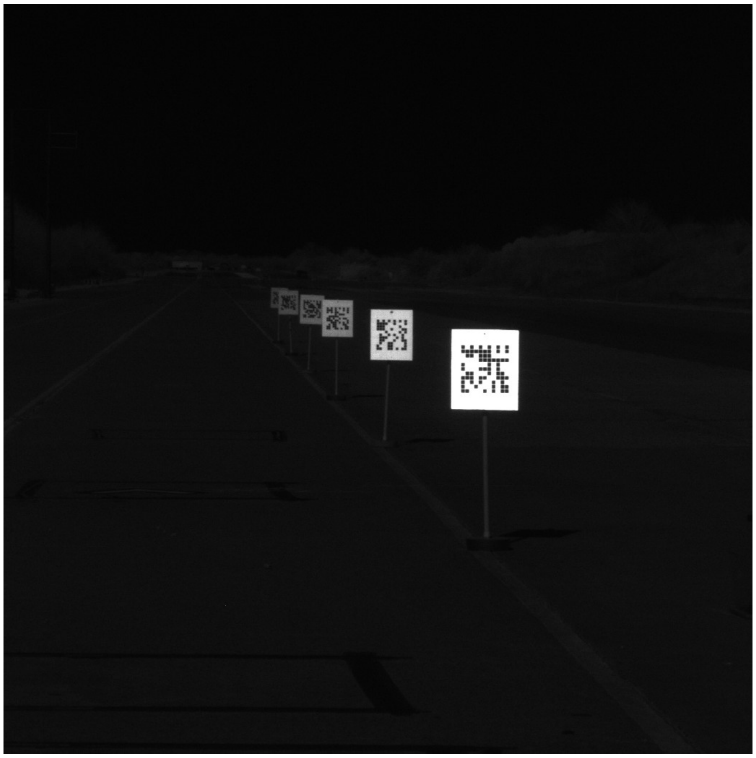


Figure 3.2: SmartCodes as seen through an IR filter (cropped)

To read a SC, a standard grayscale camera with an IR bandpass filter is used, and for the remainder of the thesis, will be referred to as the camera for the sake of brevity. For the vehicle to find SCs in the infrastructure, the vehicle first runs an edge detection algorithm on the raw images received from the camera. The detection will result in a unique hex number from the 2D bar code as well as the width and height of the SC in pixels. With the unique hex code, the vehicle can look up the signs GPS information, (X_{s_i}, Y_{s_i}), and SC width, w. The SC detection algorithm also gives the width of the SC in the pixel space, w_p. Once the vehicle reads one of the SCs, an estimate of the vehicle's relative distance from the sign is calculated using the equation below:

$$d = \frac{w}{w_p} f \tag{3.1}$$

Where f is the focal length of the camera. The focal length acts as a transformation that converts pixels into meters based on the intrinsic parameters of the camera. Figure 3.3 shows a diagram of how the position of the vehicle is estimated based on the position of the SC sign. In the figure, δ is the angle of the sign with respect to the vehicle, and ψ is the vehicle's yaw angle. Therefore, the global position of the vehicle can be found by subtracting the vehicle's relative position from the sign from the global position of the sign:

$$\begin{aligned} X_c &= X_{s_1} - dsin(\delta + \psi) \\ Y_c &= Y_{s_1} - dcos(\delta + \psi) \end{aligned} \tag{3.2}$$

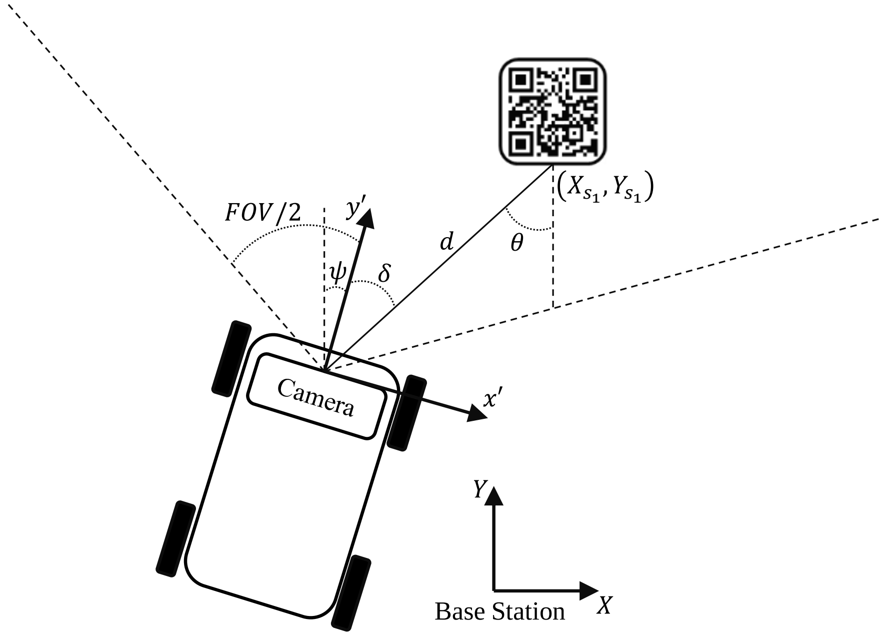


Figure 3.3: Sign angle calculation diagram

The method for calculating δ is depicted in figure 3.4. By first assigning the left most pixel in the image as $P_0 = 0$, the right most pixel as P_{image}, and knowing the field of view (FOV) of the camera, the angle of the sign with respect to the vehicle can be approximated as follows:

$$\delta = FOV\left[\frac{P_{center}}{P_{image}} - \frac{1}{2}\right] \tag{3.3}$$

Where P_{center} is the location of the center of the SC sign in pixels.

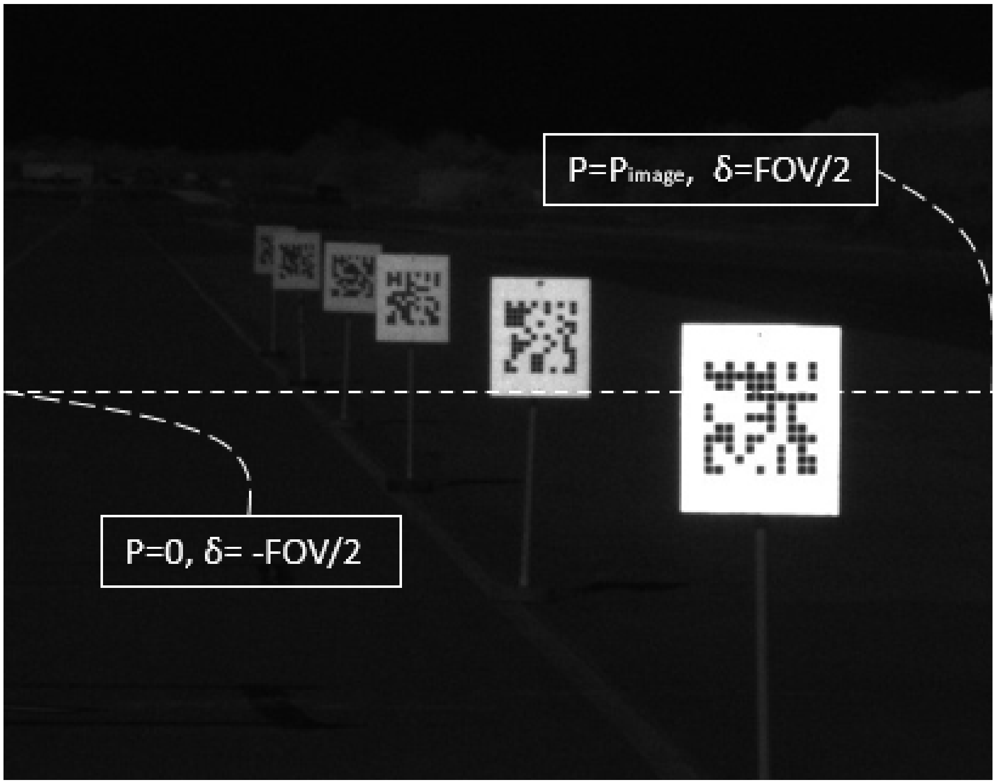
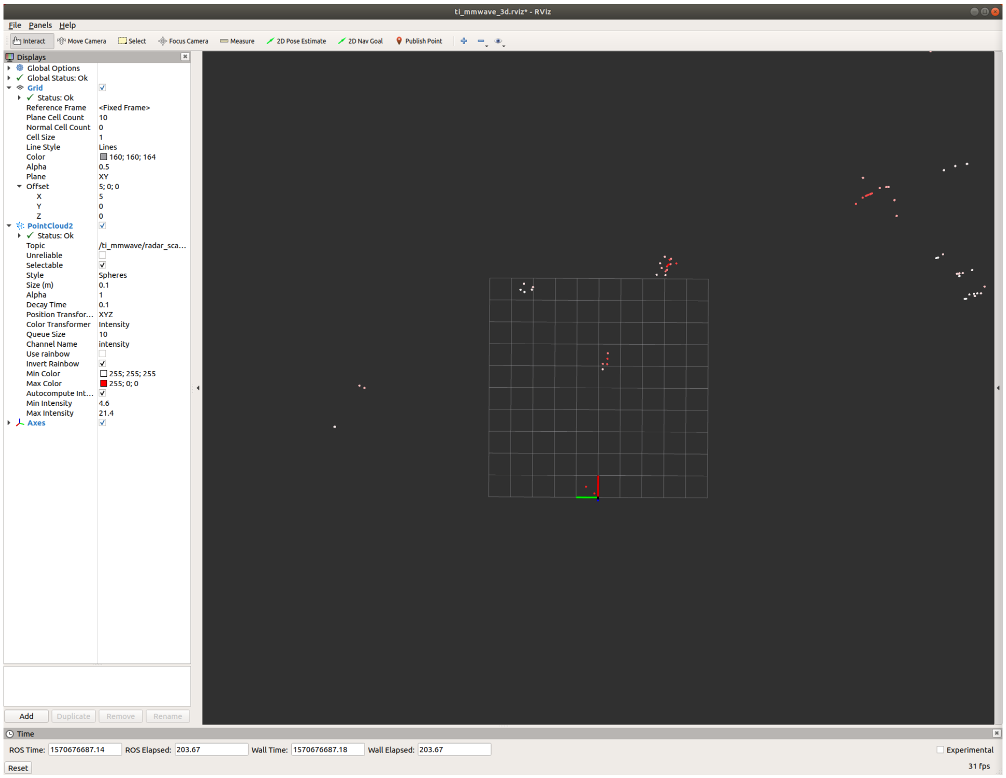


Figure 3.4: Sign angle approximation (image is cropped)

3.1.3 Radar Technology

Previous experiments showed that the longitudinal estimates provided by the camera were unreliable with errors exceeding several meters on occasion. Consequently, a low-cost frequency modulated continuous waveform (FMCW) mmWave radar was added into the sensor array to more accurately measure the distance of a SC sign from the vehicle. The underlying principle of the radar exploits the transmitted chirp or waveform of the signal. Because the frequency of the signal is increased linearly with time, a detected object will be proportional to the difference (using a signal mixer) between the transmitted and received signal called the intermediate frequency (IF). Each object detected will correspond to a unique IF. Through a series of fast-fourier transforms on the IFs the radar is able to ascertain the range, angle-of-attack, velocity, and reflected signal intensity of a detected object. The output of these values results in a point cloud of Cartesian points associated with objects that reflect the radar signal back with a certain intensity. A more detailed description of the exact theory behind FMCW radars can be found in [26]. Figure 3.5 shows a figure of the raw radar data. Each detected point is plotted in Cartesian space, and the color of each point corresponds to the reflected intensity of the point. The color is a gradient ranging from white as the lowest intensity value to red as the highest intensity value.



Global Options	
✓ Global Status: Ok	
Grid	☑
✓ Status: Ok	
Reference Frame	<Fixed Frame>
Plane Cell Count	10
Normal Cell Count	0
Cell Size	1
Line Style	Lines
Color	▨ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	5; 0; 0
X	5
Y	0
Z	0
PointCloud2	☑
✓ Status: Ok	
Topic	/ti_mmwave/radar_sca…
Unreliable	☐
Selectable	☑
Style	Spheres
Size (m)	0.1
Alpha	1
Decay Time	0.1
Position Transfor…	XYZ
Color Transformer	Intensity
Queue Size	10
Channel Name	intensity
Use rainbow	☐
Invert Rainbow	☑
Min Color	□ 255; 255; 255
Max Color	🟥 255; 0; 0
Autocompute Int…	☑
Min Intensity	4.6
Max Intensity	21.4
Axes	☑

ROS Time: 1570676687.14 ROS Elapsed: 203.67 Wall Time: 1570676687.18 Wall Elapsed: 203.67 ☐ Experimental

31 fps

Figure 3.5: Raw radar point cloud data

The raw point cloud data is not useful by itself, so post processing on the data must be done. In particular, objects must be objectively defined and tracked from the point cloud data. To separate noise and objects from the environment, the radar was configured to run a multi-target tracking (MTT) algorithm. The algorithm is based on the PDA method using a constant acceleration model. First, the radar runs a clustering algorithm called DBSCAN which averages points of data that are similar based on various criteria (position, velocity, intensity, number of points) into a single point. After clustering, the observations are assigned to each tracked object based on maximizing an assignment table of data probabilities based on NN and the Gaussian likelihood function. Lastly, each object can be updated to the next state. The algorithm also uses a maintenance step to handle object initialization and deletion. More on this algorithm can be found in [27]. Figure 3.6 shows the radar with object tracking implemented. The green squares represent objects that the radar is tracking and the other points are the same as in figure 3.5. From the tracked objects the radar outputs, they can be filtered further to limit unnecessary computation. Exploiting the fact that SCs are stationary, they will have exact velocities negative of the vehicle. Objects that do not pass a velocity threshold are determined to be moving objects and will be rejected as a potential SC observation.

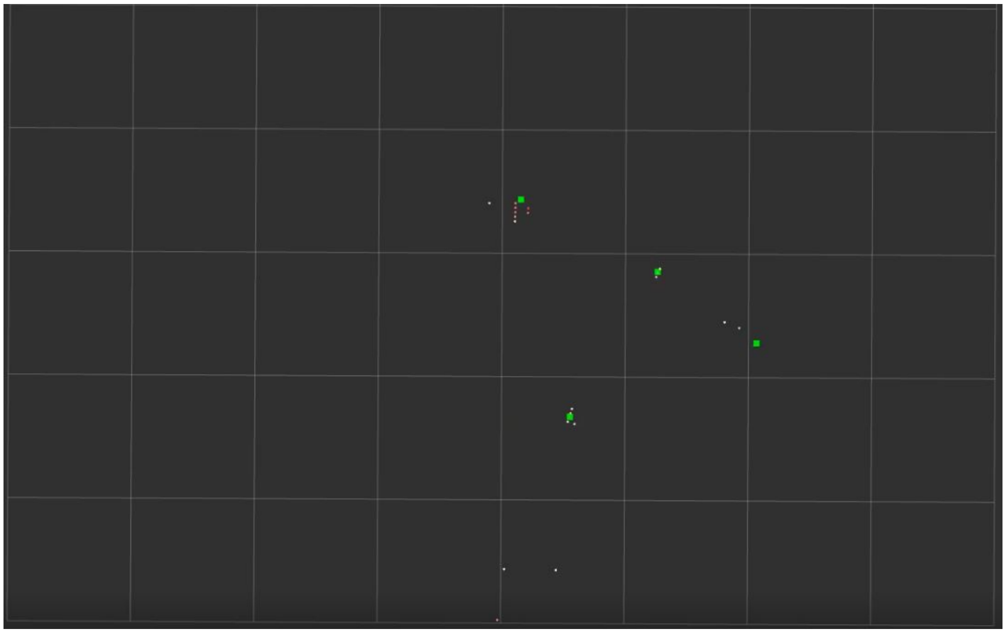


Figure 3.6: Radar point cloud with object tracking

The radar does not have the ability to objectively classify a SC as a sign, so the NN algorithm as defined in section 2.2.2 is used to associate radar observations with camera observations. In addition to the MTT algorithm configured on the radar, the radar parameters were specifically tuned to limit noise especially in close proximity to the radar without sacrificing too much performance. Additionally, the detection range of the radar was tuned such that a SC sign could be detected theoretical 80m away. A consequence of increasing the detection distance of the radar was a decrease in angular resolution and range resolution. More information into the radar's pitfalls and performance can be found in appendix A.

3.2 Experimental Methods

The set up for the experiment testing the passive IEA method were performed with a Lincoln MKZ, an industrial computer using the ROS ecosystem, a TI AWR1843BOOST radar, a PointGrey camera with an IR bandpass filter, and a SwiftNav Piksi for estimating ground truth. The vehicle is drive-by-wire (DBW), and by nature, has wheel speed sensors, steering wheel sensors, brake sensors, and throttle sensors to name few. Every DBW sensor communicates over the CAN protocol, but all the sensor data is passed through a serial converter. The vehicle also has an IMU and GPS equipped. The two sensors, which are the work-horses of this research, are the camera and radar. The radar and camera are mounted on the front, right side of the roof rack on the MKZ shown in figure 3.8. The radar is the small red PCB on the right, the camera is in the enclosure, and on top of the camera, is a Piksi. Each SC sign used in this experiment is made from a form of plastic and is backed with aluminum foil to emulate a real sign's retro-reflectivity, and they are mounted 63.5in (top of sign) from the ground. Another Piksi, which acts as the base station, is arbitrarily placed in close proximity to the experiment. Lastly, 8 SC signs are placed 42m apart from each other such that the left edge of the sign just outside the outer edge of the lane marking and perpendicular to the road.

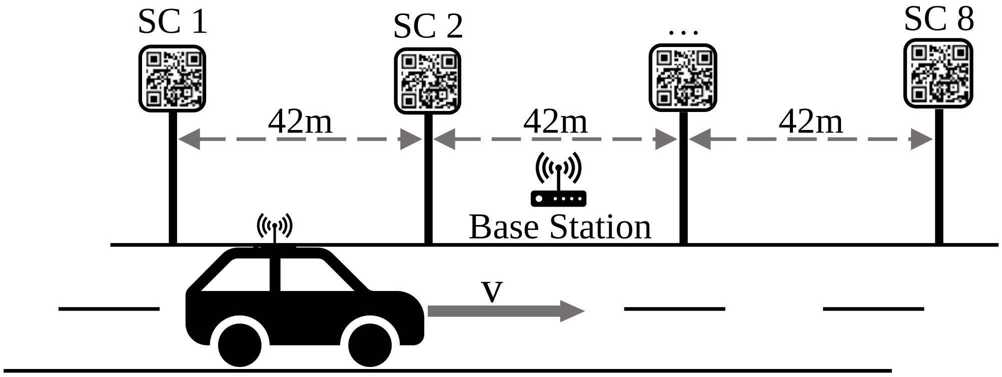


Figure 3.7: Experimental Setup

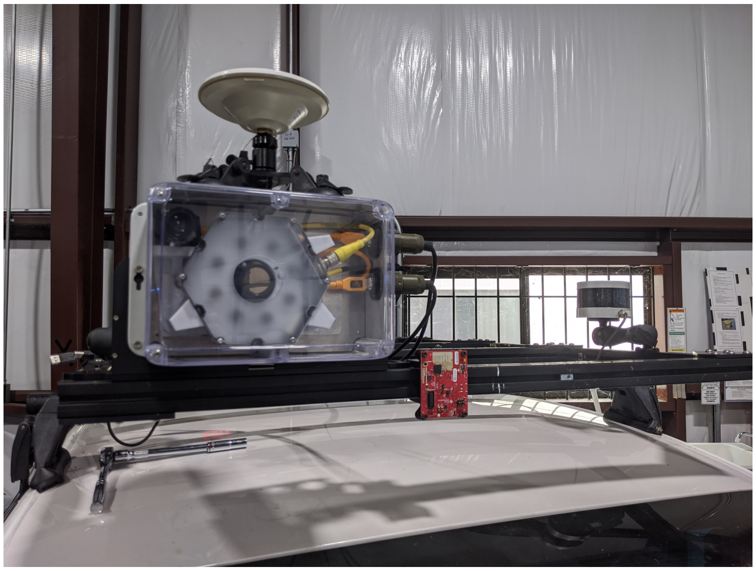


Figure 3.8: Experiment Sensor Array

After all signs are set up, their locations are recorded using the Piksi, and the SC sign database is updated with their respective locations. At the beginning of each test, the vehicle is stationary and all SC signs are outside of the detection range of the sensors. The vehicle is then driven at constant speed from 10 to 40mph until SC signs are no longer detectable.

3.3 Results

In this section, the results of the passive IEA experiments are presented. An overview of all experimental tests can be viewed in table 3.1 which tabulates the root-mean-square error (RMSE) of all tests. Figures 3.9 and 3.10 shows the lateral and longitudinal error of the filtered position of the vehicle as a function of distance traveled. The blue colored line corresponds to the 10mph test while the green, red, and cyan lines show the 20mph, 30mph, and 40mph test respectively. Figures 3.11 and 3.12 are included in the data to show the data points associated with the 40mph test case. In these figures, each data point is either a square or a circle depending on if the data corresponds to the radar or camera, respectively. All signs also have their own unique color to differentiate which data corresponds to each sign.

The data shows that the vehicle can localize itself without GPS signal within 0.39m longitudinally and 0.18m laterally in the sense of RMSE. In all tests the lateral error remained under ±0.5m which is also true in the longitudinal case with the exception of the 40mph test.

The filter's longitudinal accuracy relies heavily on the radars performance which can easily be seen by comparing the results of the 10mph and 20mph tests to the 30mph and 40mph tests. At lower speeds the radar's detection accuracy and detection consistency are superior when compared to higher speeds. When the vehicle was tested at 30mph and 40mph, for unknown reasons, the radar's detection suffered greatly. As examples, in figure 3.12 at 4 seconds the radar struggles to detect sign 2 which leads to error growth until sign 3 gets detected at the 7 second mark. The filter maintains a relatively stable error until the 16 second mark where the radar looses detection of sign 7. Error growth occurs again with sign 8 after the 19 second mark. All things considered, with a more optimized radar, the 30mph and 40mph cases could gain significant improvements. Appendix A shows radar specific graphs which show its data in more detail.

The lateral error through all 4 tests showed stable results regardless of speed. Contrary to the longitudinal error, the lateral error decreased as speed increased. However, figure 3.11 shows an over confidence in the camera's lateral observations as the filter closely follows and reacts to new camera observations. Overall, the vehicle's filter performed more consistently and had lower error laterally when all errors are averaged. With increases in the radar's performance, however, the longitudinal error would likely fall inline with the errors seen in the 10mph and 20mph longitudinal cases.

Table 3.1: Passive IEA experimental RMSEs

	10mph	20mph	30mph	40mph
Longitudinal error (m)	0.02	0.02	0.24	0.39
Lateral error (m)	0.18	0.17	0.13	0.13

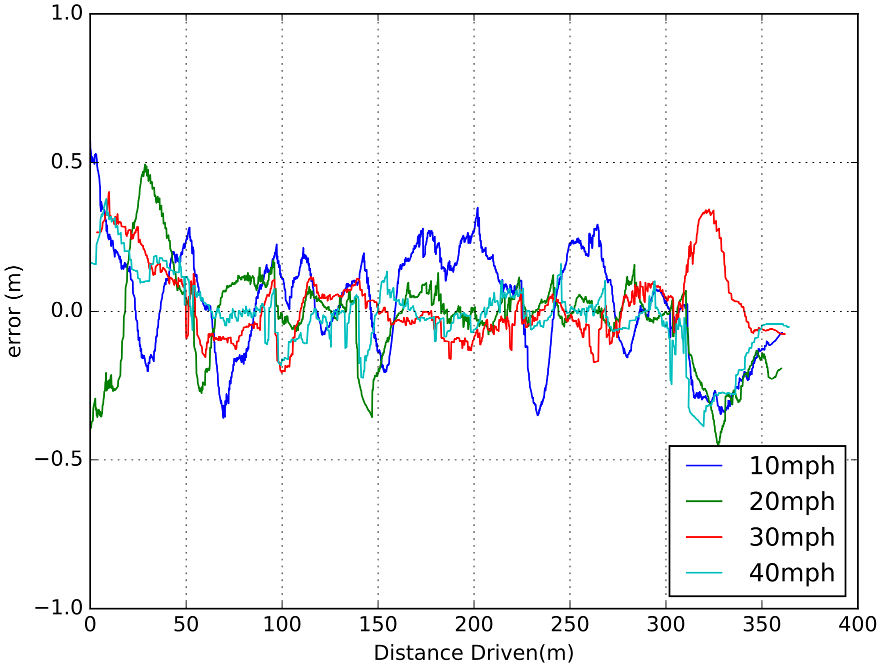


Figure 3.9: Lateral error

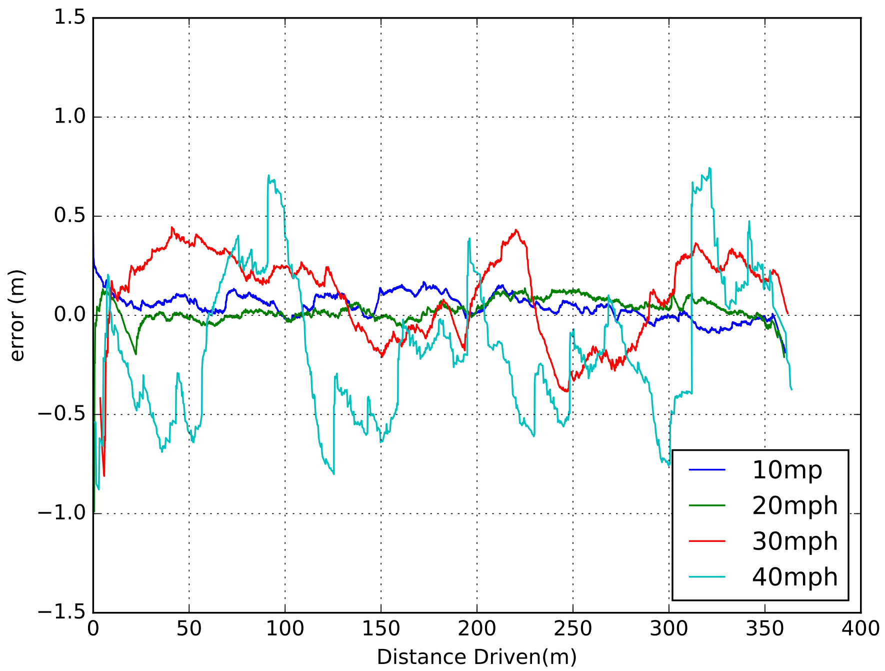


Figure 3.10: Longitudinal error

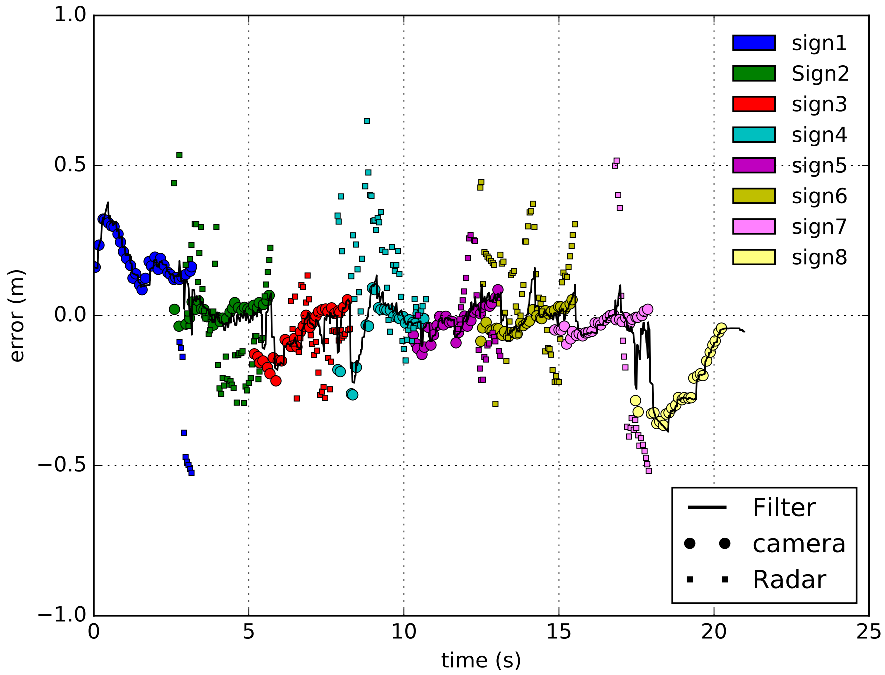


Figure 3.11: 40mph lateral error with data points

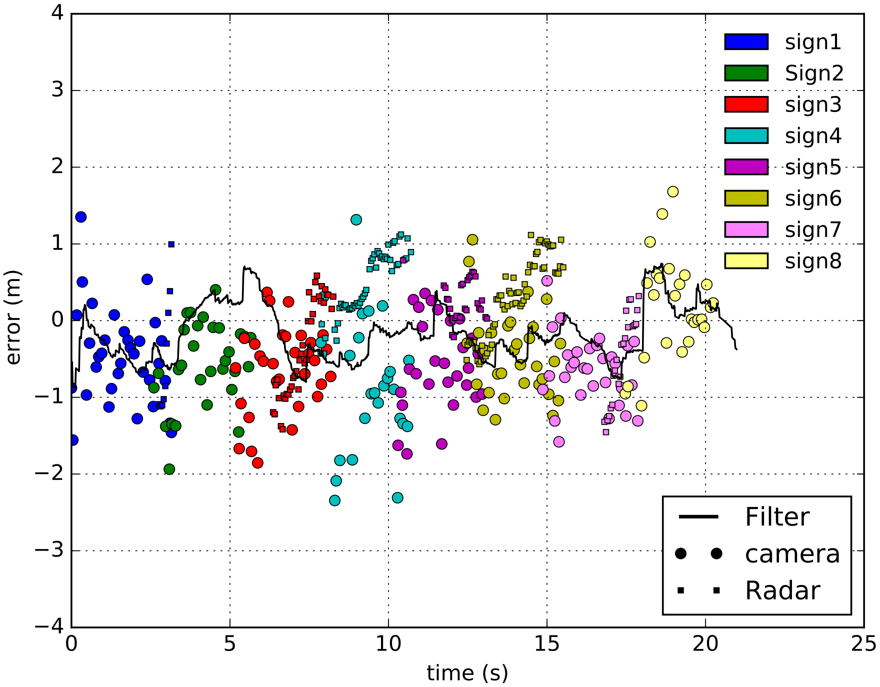


Figure 3.12: 40mph longitudinal error with data points

3.4 Further Study

Although this project has concluded by the direction of 3M, there are immediate areas of research to further improve the localization using SCs. Experiments which are discussed in appendix A.3 show the radar had increased detection consistency both in distance and larger incident angles. Specifically, research in materials to increase the retro-reflective properties of the SC signs could greatly increase the performance of the radar. The SC detection system also has room for improvement. Developing a more robust and accurate SC detection algorithm could directly lead to improved system performance. In addition, the use of two cameras in a stereo setup has the potentially to decrease the longitudinal error of the camera observations. Lastly, a standard EKF algorithm was used for this experiment, but a more robust state estimation algorithm could be used to increase the filters performance.

4. ACTIVE INFRASTRUCTURE ENABLED AUTONOMY

4.1 Proposed Method

In the case of active infrastructure enabled autonomy (IEA), the proposed method tracks multiple vehicles on the road by using infrastructure based radars and cameras. This method proposes the use of 4 cameras and a traffic monitoring radar to provide localization information. Machine learning based vehicle detection will be leveraged on each camera feed to provide classification information as well as position estimations. The radar will provide high accuracy position estimations that will be fused with the camera detections to provide a novel approach for vehicle tracking. The following section will discuss the active IEA method in detail.

4.1.1 Architecture

The architecture of active IEA autonomy can be overviewed in figure 4.1 and is a modification to the multi-smart sensor packs (MSSP) initially proposed in [5]. Each MSSP uses a set of 4 cameras which are all powered via power-over-ethernet (POE). Additionally, the MSSP is rigged with a traffic monitoring radar which can detect vehicles over the effective distance of the camera system. MSSP are by nature modular which allows daisy-chaining of other MSSPs that allow an entire stretch of road to track vehicles where MSSPs are implemented. While sensors such as radars and LiDARS are accurate, they lack the ability to easily classify objects – they each require extensive processing of data. In areas where both vehicles, pedestrians, cyclists, and other modes of transportation need to be tracked LiDAR and radar fall short. For the aforementioned reasons, cameras are leveraged to objectively classify objects and simultaneously provide position estimates. In essence, using this two-factor position estimation provides a robust localization method for object tracking.

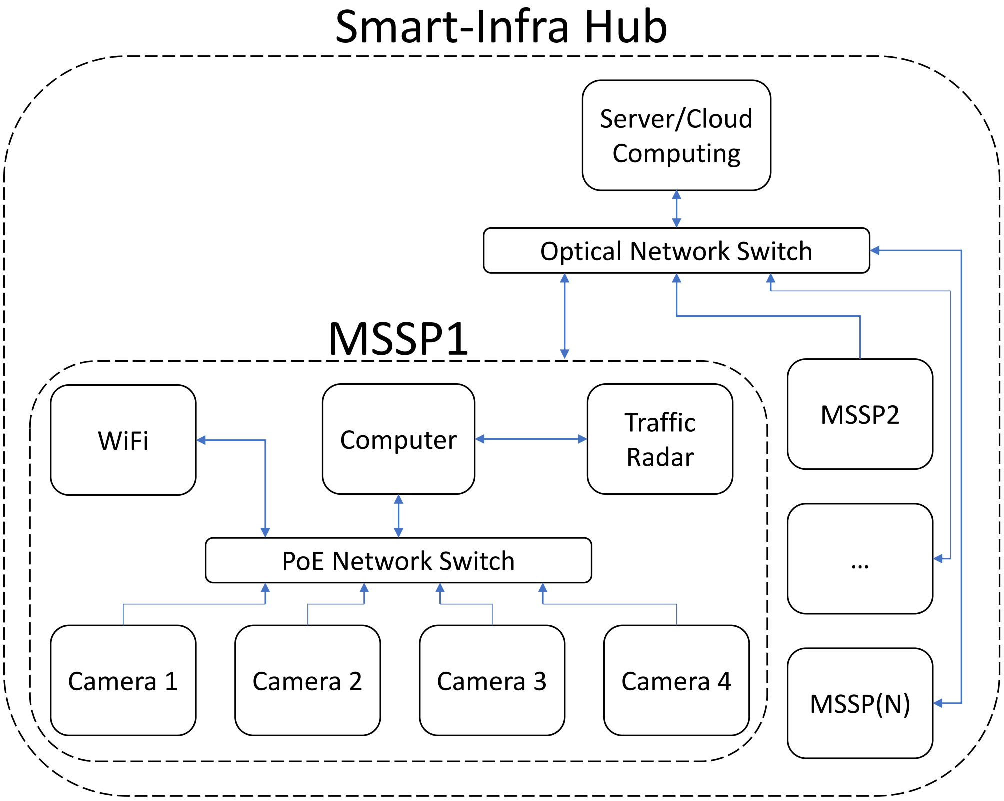


Figure 4.1: Active infrastructure enabled autonomy architecture

4.2 Cameras

Unlike LiDARs and radars, cameras by nature do not provide position information. Only after processing an image can position be ascertained from an image. The process for calculating position data from a camera image relies on camera calibration and machine learning which is detailed in the subsequent sections.

4.2.1 Machine Learning

To extract pixel coordinates from the cameras a machine learning (ML) algorithm called YOLOv3 [28] will be run on the neural network called Darknet [29]. The output of the ML algorithm is a bounding box which contains the silhouette of a detected object. YOLOv3 itself is a detection algorithm which only uses a single image to detect objects. It is also computationally fast compared to other detection algorithms. YOLOv3 is capable of running at over 100Hz in certain configurations allowing for real-time object detection and tracking hence its use in this research. The downside of running these ML algorithms, however, is their immense computational cost. On a CPU, YOLOv3 can barely run a single camera at 1Hz, but ML can leverage parallel computing so GPUs are almost exclusively used in machine-vision detection. This research required an Nvidia RTX 2080 to run 4 cameras at 10Hz as an example. This should not deter further research though because the computational throughput of GPUs has been drastically increasing over the past decade. Figure 4.2 shows an example of the output of the machine learning algorithm.

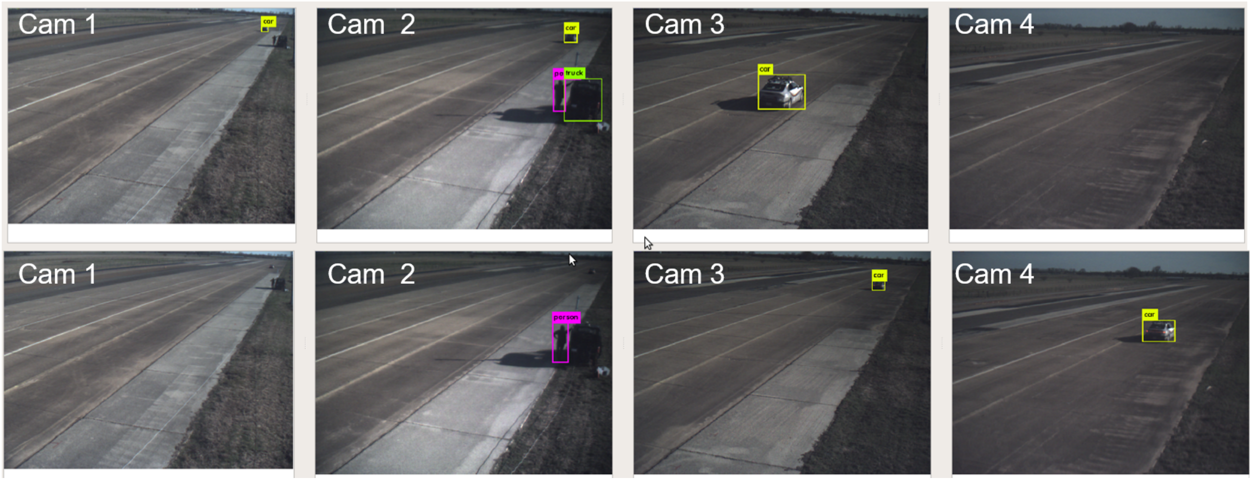


Figure 4.2: Output of Darknet and YOLOv3

4.2.2 Camera Calibration

To get 3D world coordinates, X, from 2D pixel coordinates, x, a transformation must be known called the camera projection matrix ,**P**:

$$x = \mathbf{P}X \tag{4.1}$$

The camera projection matrix can be expanded into two matrix properties — intrinsic and extrinsic camera parameters:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|t] \tag{4.2}$$

Where **K** represents the intrinsic camera parameters, and $[\mathbf{R}|t]$ is the extrinsic parameters. The intrinsic parameters consist of the focal length, f_x and f_y, and the principal point, c_x and c_y, which represents the center of the image:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

The extrinsic parameters are represented as an augmented matrix of the orientation of the camera, **R**, with respect to the global coordinate system and the position of the camera, t, with respect to the camera's reference frame.

Camera calibration techniques have been study extensively, and perhaps the most well known are the direct-linear transform (DLT) and perspective n-point (PnP) methods. These methods both have open-source implementations which are ideal for this research. Therefore, these methods were the primary candidates for this research. Each method requires control points (CP) which are known 3D locations from the environment with corresponding pixel locations from an image.

4.2.2.1 Control Point Collection

In this research, the method for collecting CPs leveraged the existing MSSP, two Piksis in RTK mode, and the test vehicle. One Piksi is mounted on the center of the vehicle and the vehicle is driven through all lanes covered by the cameras. Meanwhile, the ML vehicle detection algorithm running on each camera and the Piksi store the pixel and world coordinates, respectively. The CP collection method used is not perfect and has a fair amount of noise because the bounding boxes drawn around the vehicle do not always align with the Piksi.

4.2.2.2 Calibration Algorithm

With the control points collected, the intrinsic and extrinsic camera parameters can be calculated. One well known method which calculates both parameters simultaneously is the DLT. However, DLT requires CPs that are not co-planar, and for good results, the CPs must span as much of the volume in the FOV of the camera. The CP collection described in section 4.2.2.1 restricts all CPs to a plane which results in CPs that do not span any volume. Consequently, the DLT method can not be used. The camera calibration parameters need not be computed simultaneously, however. The cameras can easily have its intrinsic parameters computed before being mounted on the infrastructure using a standard chess board technique presented by Zhang [30]. With the intrinsic parameters calculated, only the location and orientation of the camera is unknown. Given the intrinsic parameters, the PnP algorithm can be used. PnP requires only 3 CP, but for better results the PnP problem can be expanded into a least-squares solution. Shown in [31], however, least squares based PnP methods are sensitive to CP outliers, and with the control CP collection method devised, there can be numerous CP outliers. To deal with outliers, the PnP method can first be combined with RANSAC [32] to remove outliers and then followed by least-squares methods.

4.3 Radar Calibration

The radar has a similar issue to the cameras in that the location and orientation with respect to the global frame is unknown. The output of the radar mapped to world frame coordinates can be shown as follows:

$$x = P_r + R(\theta)c_i^r \tag{4.4}$$

Where x is the global coordinates of a detected object, $P_r = [p_x, p_y]^\top$ is the location of the radar, $R(\theta)$ is the rotation which transforms the radar points to the global frame, and c_i^r represents a detected radar point. A schematic of the radar calibration can be seen in figure 4.3. Similar to the camera calibration method, CPs are required. The exact same method in section 4.2.2.1 can be used for collecting CPs with the modification to collect radar points instead of pixel coordinates. The radar's coordinate frame is shown as $\mathcal{F}_r = (X^r, Y^r)$, and CPs are represented as either green (c_i) or red (c_i^r) for the points collected via the Piksi and radar, respectively. Although visually each CP is represented as two different colored dots, they are a single instant of time showing the difference between the radar and ground truth measurement of the same point.

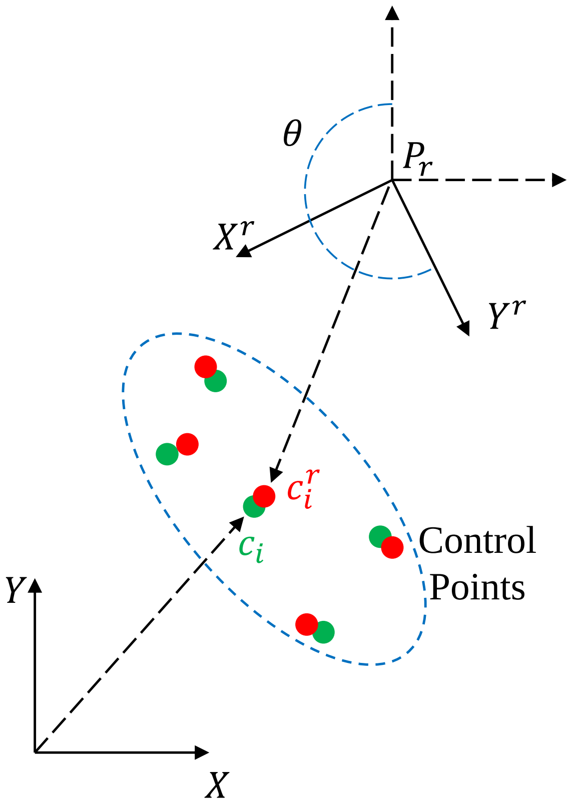


Figure 4.3: Radar calibration schematic

To calculate the extrinsic parameters, first we define the extrinsic parameters of the radar as $\beta = [p_x, p_y, \theta]^\top$. Then we create a cost function:

$$f(\beta) = \frac{1}{2}\sum_{i=1}^{n} r_i(\beta)^2 \tag{4.5}$$

Where r_i is a residual from the i^{th} control point:

$$r_i = \|c_i - (P_r + Rc_i^r)\| \tag{4.6}$$

The optimal parameters, β^*, can be found by minimizing eqn. 4.5:

$$\beta^* = \underset{\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} r_i(\beta)^2 \right) \tag{4.7}$$

The problem defined above is a non-linear least-squares problem and can be solved iteratively using the Gauss-Newton method shown below:

$$\beta(k+1) = \beta(k) - (J_r^\top J_r)^{-1} J_r^\top r(\beta(k)) \tag{4.8}$$

Where β^* is considered converged after an arbitrary threshold, $\|\beta(k+1) - \beta(k)\| < \varepsilon$, is met from k iterations, and J_r is the Jacobian of the residuals defined as follows:

$$J_r = \begin{bmatrix} \frac{\partial r_i}{\partial \beta_1(k)} & \frac{\partial r_i}{\partial \beta_2(k)} & \frac{\partial r_i}{\partial \beta_3(k)} \\ \vdots & \vdots & \vdots \\ \frac{\partial r_n}{\partial \beta_1(k)} & \frac{\partial r_n}{\partial \beta_2(k)} & \frac{\partial r_n}{\partial \beta_3(k)} \end{bmatrix} \tag{4.9}$$

4.4 Experimental Methods

In this research, a single MSSP was tested due to infrastructure and equipment limitations; however, the setup can easily be expanded to include more MSSPs. The MSSP has four cameras which are mounted on individual RSUs that are each 50m apart and 4.5m from the road. On the first RSU a radar is also mounted just under the camera, and all cameras are mounted approximately 5.5m from the ground. In the middle of the MSSP is the computer, network switch, WiFi router, and Piksi base station. The MSSP cameras are all connected into the network switch where they are also are powered via POE. The MSSP computer is connected to the ethernet switch as well so that it can communicate with the cameras, and the radar is connected to the computer via a RS-485 serial connection. Figure 4.4 shows a schematic of the experimental setup. The network switch chosen also has an optical SFP port to further expand the number of MSSPs for further research. To keep track of the vehicle's ground truth, two Piksis are set up in RTK mode with one being the base station and the other acting as the rover. Both the MSSP and the vehicle are connected to the WiFi router where information can be exchanged between the two systems.

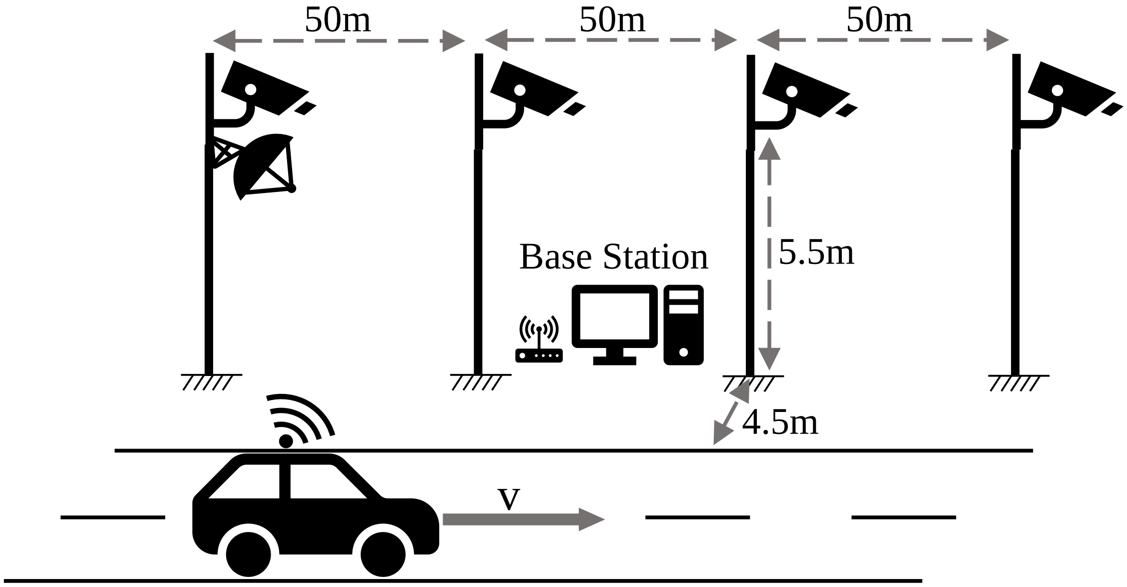


Figure 4.4: Active infrastructure enabled autonomy experiment setup

4.5 Preliminary Results

In this section, the preliminary results of the active IEA experiments are presented. Figure 4.5 displays the results of two experiments testing the active IEA system. Each experiment was performed with the vehicle moving at approximately 10mph. The first test (left graph) had the vehicle driven in a single lane while the second test (right graph) had the vehicle perform a lane change. Each graph shows the ground truth as a black line and the estimated vehicle track as a blue line. All other colors are spurious tracks determined by the system. The errors associated with each experiment are shown in figure 4.6 where the top and bottom graphs are for the single and multi lane tests, respectively. The lateral results are shown in blue while the longitudinal error is shown in red. The overall error results are tabulated in table 4.1.

Table 4.1: Active IEA experimental RMSEs

	Single Lane	Multi Change
Lateral error (m)	0.48	0.44
Longitudinal error (m)	2.25	6.84

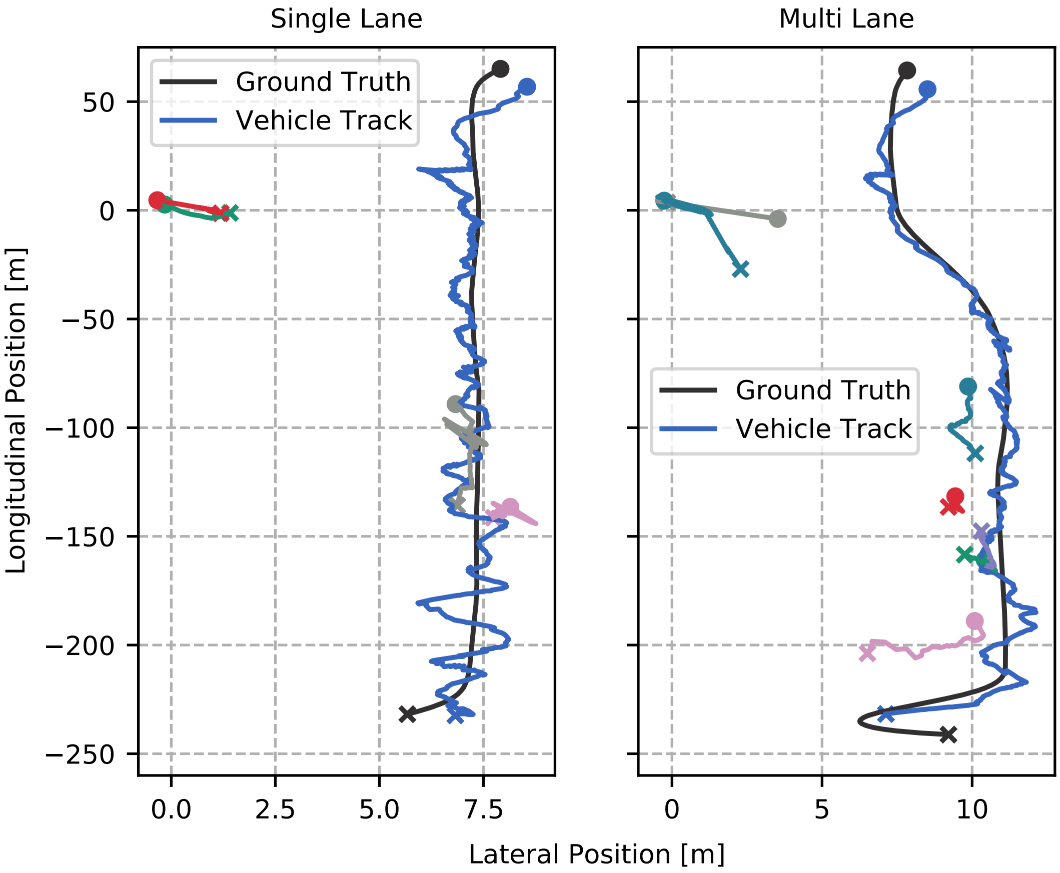


Figure 4.5: Active IEA filter data

Overall, the lateral errors were 0.48m and 0.44m for the single lane and multi lane tests, respectively. For the longitudinal errors, the single lane test had a 2.25m error and the multi lane test had a 6.84m error. By a large margin, the lateral estimates outperformed the longitudinal estimates. Large spikes in the longitudinal errors can be attributed primarily to camera observations. Figures 4.7 and 4.8 show the sensor data associated with each test. During the first several observations of each camera, the camera's observation errors are large because the vehicle is not completely in frame, and the bounding boxes do not describe the entire vehicle. These initial camera observations cause the filter to lag behind ground truth until the entire vehicle is within frame. The errors in the beginning of the vehicle track are due to the radar's observations. Radars by nature have poor accuracy at the extremes of their FOV which can be seen in the radar calibration data shown in figures 4.7 and 4.8. Consequently, the vehicle track gets initialized with a couple meters of error, and the error eventually lowers as the vehicle travels out of the extremes of the radar's FOV.

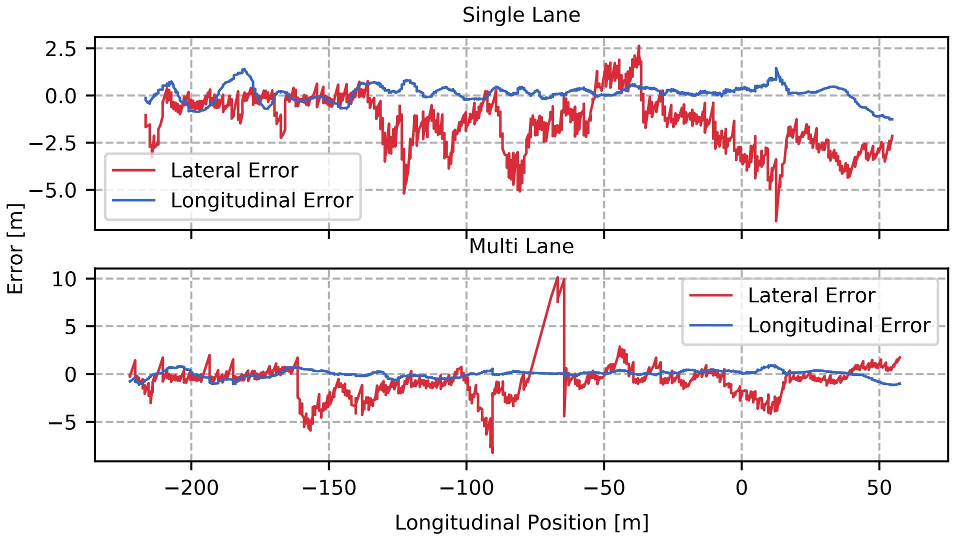


Figure 4.6: Active IEA filter error

The spurious tracks along the ground truth path are caused by inaccurate camera observations. To shed light on the camera observation issue, refer back to figures 4.7 and 4.8 for the calibration data. In both tests, only cameras 2 and 3 had acceptable calibrations. In fact, cameras 1 and 4 had to be excluded from the filter to prevent excessive target generation and accuracy degradation of the vehicle track. A similar result can be seen in the multi lane case. Visually it seems as though camera 1 and 2 have similar errors and orientations, but camera 1 has significant error in the longitudinal direction. Camera 4 was excluded because of the error in orientation as shown by the large skew in both figures. Figure 4.5 shows two areas of track generation — the origin and tracks in the vehicles path. The tracks generated near the origin are the base station and are therefore a valid track. All other tracks are caused by the camera's longitudinal observations lagging behind the vehicles track. The farther the vehicle is from each camera the worse their longitudinal estimates, and when the observation's error exceed the data association threshold, a new track is generated.

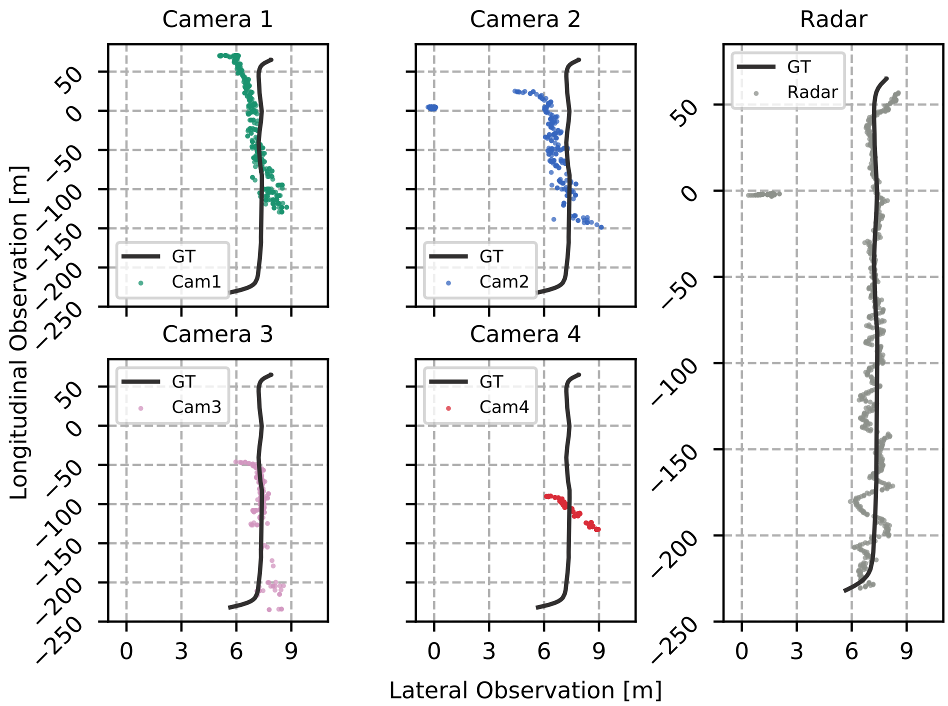


Figure 4.7: Active IEA sensor data

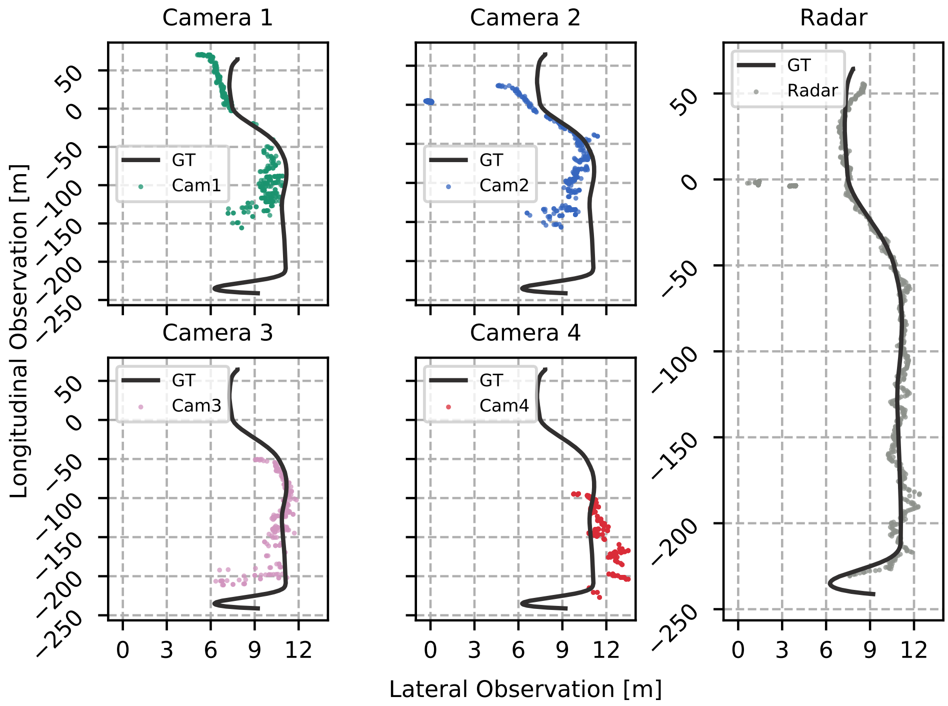


Figure 4.8: Active IEA sensor data (multi lane)

4.6 Further Study

There are three major areas that the active IEA system would benefit from further research and development. The most impactful improvement would be further development in the camera calibration process. It was initially theorized that the PnP+RANSAC could handle the noise in the CP collection process; however, it is clear from the preliminary data collected, a more robust method for collecting less noisy and more accurate CPs is needed. With an improved camera calibration process, the active IEA MTT filter would see immediate performance benefits. Additionally, the MTT algorithm used in the active IEA experiments was a naive approach to tracking multiple targets, but it works well in low noise environments; however, as active IEA is further researched, more advance MTT algorithms like JPDA can be implemented to increase tracking robustness. Lastly there was no vehicle-to-infrastructure communication meaning that no vehicle dynamics could be included in the MTT filter. Further improvements to performance would be to enable vehicle-to-infrastructure communication, thus enabling more complex and descriptive models to be used in the filter.

5. CONCLUSIONS

The two main objectives of this research were to develop a means to localize a vehicle in the absence of GPS signals. Each topic was developed independently, but this research lays the groundwork for the passive and active IEA methods to be combined for even more robust localization. Due to delays from COVID and Texas weather the active IEA research remains to be finished; however, the majority of the research has been done, and preliminary experiments have been run. In summary, for the passive IEA research:

- Vehicle localization was enabled in GPS denied environments by leveraging passive elements (signs) in the infrastructure called SmartCodes.
- A sensor fusion algorithm (EKF, NN) was developed to localize a vehicle globally utilizing SmartCodes, an IR camera, a radar, and on-board vehicle sensors.
- Experimental results showed an overall average RMSE of 0.20m when considering both longitudinal and lateral errors.

To summarize the active IEA research:

- An architecture for active IEA has been developed, implemented, and tested on real hardware.
- Calibration methods for both the infrastructure cameras and radars have been developed to facilitate sensor fusion algorithms.
- A multi-target tracking linear Kalman Filter has developed and tested (still under development) showing global localization with an overall average RMSE of 3.45m when considering both longitudinal and lateral errors.

The two localization methods presented in this research demonstrate the capabilities of IEA in providing stable localization information, redundancy to systems, and offloading computational cost from vehicles. It is not hard to extrapolate that the fusion of these two systems could decrease the computational load on connected vehicles, and provide stable localization information in urban canyons. Additionally, the two systems researched were developed in a modular approach — the scope and complexity of these systems can be easily expanded. In the case of passive IEA, SCs can be implemented on a large scale (RELLIS Campus as an example) and the true potential of SC would be realized. The active IEA case poses a larger issue with scalablity because the land and infrastructure requirements, but the architecture was developed such that the system could be scaled easily. Unique to the active IEA case is the scalability to track other moving objects such as pedestrians and cyclist enabling object threat detection to vehicles as well.

All things considered, further development in both passive and active IEA localization methods could pay dividends for autonmous vehicles of the future.

REFERENCES

[1] “Traffic safety facts annual report tables,” tech. rep., National Highway Traffic Safety Administration, Jun 2020.

[2] D. Shrank, B. Eisele, and T. Lomax, “2019 urban mobility report,” tech. rep., Texas AM Transportation Institute, 2019.

[3] “Sources of greenhouse gas emissions,” Oct. 2018.

[4] K. Kockelman *et al.*, “An assessment of autonomous vehicles : traffic impacts and infrastructure needs : final report.,” tech. rep., Mar 2017.

[5] S. Gopalswamy and S. Rathinam, “Infrastructure enabled autonomy: A distributed intelligence architecture for autonomous vehicles,” *CoRR*, vol. abs/1802.04112, 2018.

[6] “Gps accuracy,” 2020.

[7] “Vulnerability assessment of the transportation infrastructure relying on global positioning system,” Aug 2001. Tech Report.

[8] J. Snyder, D. Dunn, J. Howard, T. Potts, and K. Hansen, “"invisible" 2d bar cose to enable machine readability of road signs - material and software solutions,” tech. rep., 3M Company, 2018.

[9] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.

[10] Y. Han, C. Wei, R. Li, J. Wang, and H. Yu, “A novel cooperative localization method based on imu and uwb,” *Sensors*, vol. 20, no. 2, 2020.

[11] A. Welzel, P. Reisdorf, and G. Wanielik, “Improving urban vehicle localization with traffic sign recognition,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2728–2732, 2015.

[12] A. Schaefer, D. Büscher, J. Vertens, L. Luft, and W. Burgard, "Long-term urban vehicle localization using pole landmarks extracted from 3-d lidar scans," in *2019 European Conference on Mobile Robots (ECMR)*, pp. 1–7, 2019.

[13] N. Houdali, T. Ditchi, E. Géron, J. Lucas, and S. Holé, "Rf infrastructure cooperative system for in lane vehicle localization," *Electronics*, vol. 3, no. 4, pp. 598–608, 2014.

[14] D. Ravipati, K. Chour, A. Nayak, T. Marr, S. Dey, A. Gautam, S. Rathinam, and G. Swaminathan, "Vision based localization for infrastructure enabled autonomy," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1638–1643, 2019.

[15] N. Hernández, A. Hussein, D. Cruzado, I. Parra, and J. M. Armingol, "Applying low cost wifi-based localization to in-campus autonomous vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2017.

[16] C. Ang, "Vehicle positioning using wifi fingerprinting in urban environment," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 652–657, 2018.

[17] F. A. Santos, A. T. Akabane, R. S. Yokoyama, A. A. F. Loureiro, and L. A. Villas, "A roadside unit-based localization scheme to improve positioning for vehicular networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, 2016.

[18] H. Kloeden, D. Schwarz, E. M. Biebl, and R. H. Rasshofer, "Vehicle localization using cooperative rf-based landmarks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 387–392, 2011.

[19] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[20] H. F. Durrant-Whyte, "Sensor models and multisensor integration," *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 97–113, 1988.

[21] Y. Bar-Shalom, *Multitarget-multisensor tracking. Yaakov Bar-Shalom, editor.* Artech House, 1990.

[22] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[23] R. Rajamani, *Vehicle Dynamics and Control*, p. 26. Mechanical Engineering Series, Springer US, 2012.

[24] P. S. Zaki, M. M. William, B. K. Soliman, K. G. Alexsan, K. Khalil, and M. El-Moursy, “Traffic signs detection and recognition system using deep learning,” 2020.

[25] N. Mohd Ali, S. Karis, A. F. Zainal Abidin, B. Bakri, E. F. Shair, and N. Razif, “Traffic sign detection and recognition: Review and analysis,” *Jurnal Teknologi*, vol. 77, 12 2015.

[26] C. Iovescu and S. Rao, “The fundamentals of millimeter wave sensors,” tech. rep., Texas Instruments, 2020.

[27] M. Livshitz, “Tracking radar targets with multiple reflection points,” tech. rep., Texas Instruments, 2018.

[28] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.

[29] J. Redmon, “Darknet: Open source neural networks in c.” `http://pjreddie.com/darknet/`, 2013–2016.

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

[31] H. Zhou, T. Zhang, and J. Jagadeesan, “Re-weighting and 1-point ransac-based p$n$np solution to handle outliers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 3022–3033, 2019.

[32] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” vol. 24, p. 381–395, June 1981.

APPENDIX A

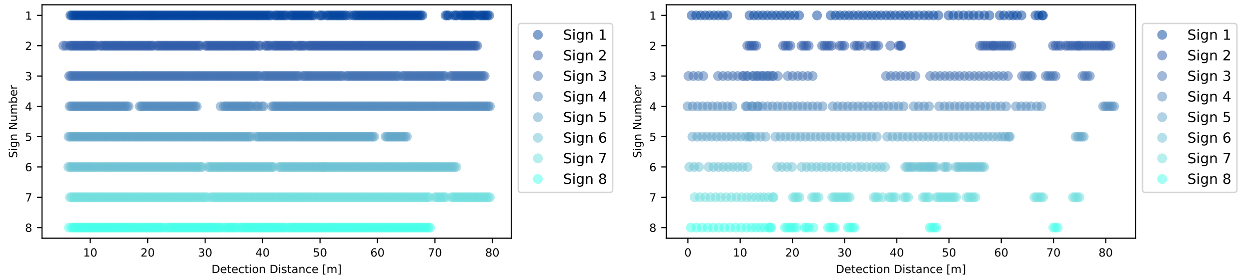
RADAR

A.1 Overview

This appendix section will be broken into two sections—Radar Detection Problems and Sign Retro-reflectivity. The detection problems section will specifically discuss issues the radar had during the passive IEA experiements as described in section 3.3, and the sign retro-reflectivity section will discuss experiments done to expand on the potential of the radar.

A.2 Radar Detection Problems

The main issues with the radar were only apparent at speeds at or above 30mph. Both the 30mph and 40mph cases had the same issues, so only the 40mph data will be shown. Because the KF heavily relied on the radar to provide accurate longitudinal data the detection consistency of the radar is important. In figures A.1a and A.1b the consistency of detection is shown. Each data point signifies a detection, and it is clear that the consistency of detection degraded substantially in the 40mph test. This detection inconsistency was a large portion of the performance decrease in the KF.

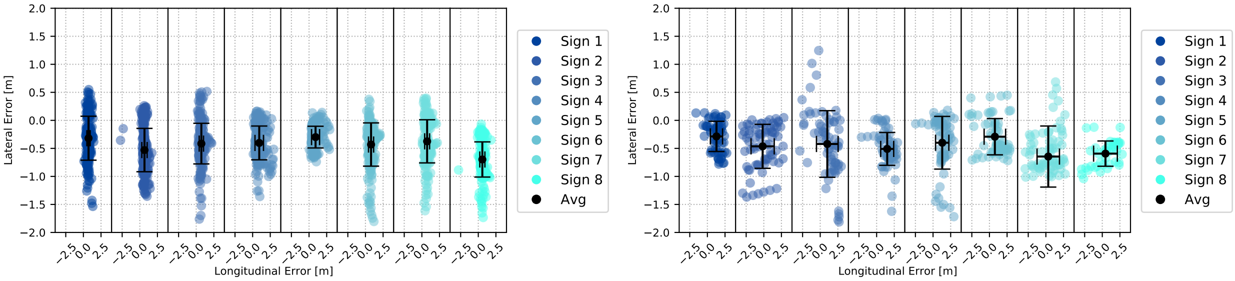


(a) 10 mph detection distance

(b) 40 mph detection distance

Figure A.1: Radar detection consistency data

During data analysis it was also compelling to see if the accuracy of the radar decreased. Figures A.2a and A.2b show the data points and error bars associated with each sign. While the variance of the radar detect ions increased, the accuracy of the detections do not appear to be significantly different.

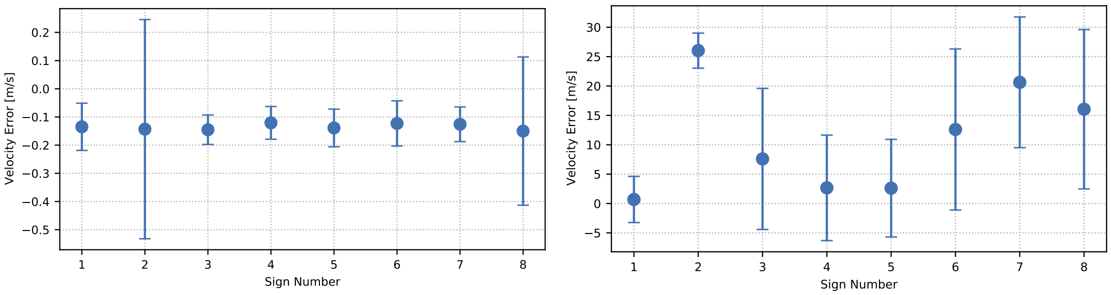


(a) 10 mph detection distance

(b) 40 mph detection distance

Figure A.2: Radar accuracy data

The detection consistency alone did not logically lead to the drastic performance degradation of the KF, so through further investigation, it was discovered that the main source of error was coming from the radar's MTT algorithm. As mentioned in section 3.1.3 a velocity filter was applied to the list of tracked objects to filter out non-stationary objects. Figures A.3a and A.3b shows the velocity error of the signs for the 10mph and 40mph test, respectively. During the 40mph test, the velocity error caused the majority of the radar's observations to be rejected causing the KF to rely primarily on the camera observations.



(a) 10 mph detection distance

(b) 40 mph detection distance

Figure A.3: Radar velocity errors

A.3 Sign Retro-reflectivity

It was well known that the performance of the radar decreased as the velocity of the vehicle increased passed 30mph, and through literature review, it was discovered that flat plates (signs in this case) only have good retro-reflective properties at incident angles under $\approx 10^\circ$ and the retro-reflectivity degrades drastically past that threshold. An experiment was performed with a 2.5in trihedral retro-reflector (RR) that was placed on top of one sign and two other signs were used as a control to see what performance increase can be expected with increased retro-reflectivity. The experiment tested speeds from 10 to 40mph, and to check the performance at higher incident angles, the test was also performed in two different lanes. The results of this experiment are summarized in figure A.4. Each line represents three tests that were averaged to get a single data point.

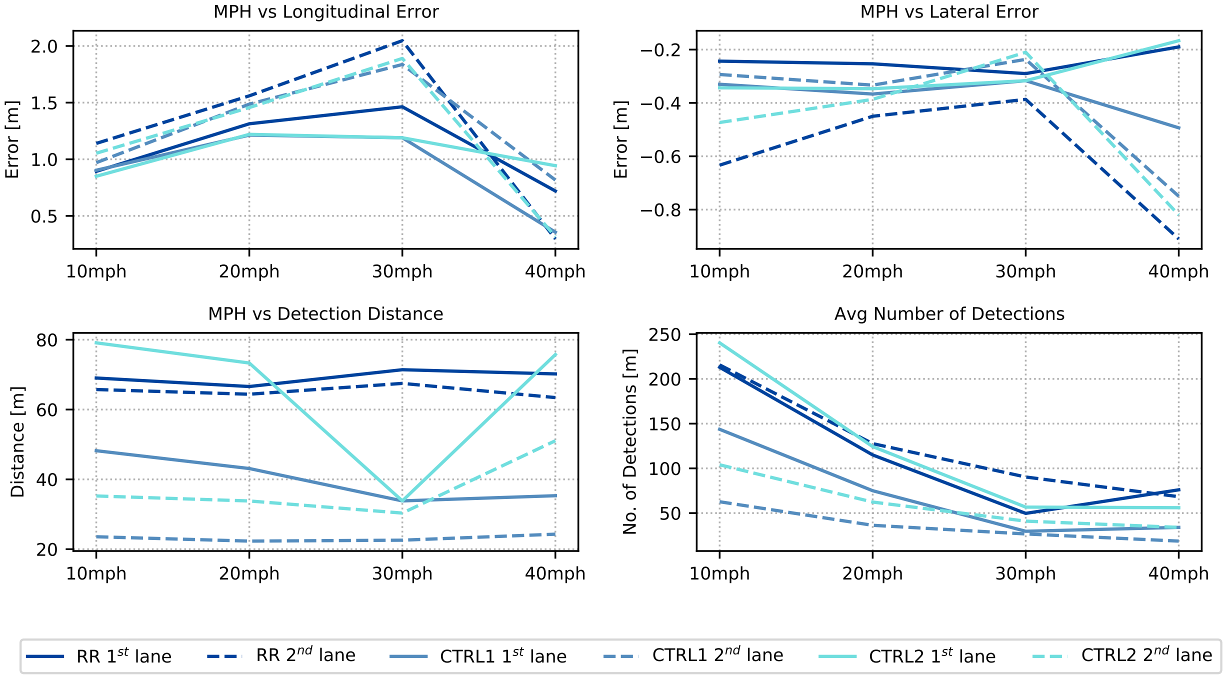


Figure A.4: Sign retro-reflector experimental data

The solid and dotted lines correspond to the first and second lane experiments, respectively.

The dark blue lines correspond to the sign with the radar RR and the other lines are the controls. The top left graph displays the longitudinal errors, the top right graph shows the lateral errors, the bottom left graph presents the maximum detection distance of the sign, and the bottom right graph exhibits the average number of times each sign was detected.

Starting with the longitudinal errors, the errors of the first and second lane for each sign had similar performance in longitudinal accuracy. The errors between each sign during the first lane and second lane test lie within the variance of each other and can be considered statistically similar. The longitudinal errors were lower when comparing the same signs under different lanes however. When it comes to lateral errors, the sign with the RR shows better performance in the first lane overall and worse accuracy in the second lane. Looking at detection distance, it is clear that the RR has an impressive consistency of detection compared to the controls. With the exception of control 2 in the first lane, the RR outperforms in all other cases. The significant drop in retro-reflectivity is directly shown during larger incident angles when comparing the first and second lane data of control 2. An interesting take away from this result, however, is that under optimal conditions, a normal sign can have similar performance to a sign with a retr-oreflector in the first lane. Lastly, looking at the number of detections, the sign with the radar RR had more detections than all other signs especially in the second lane. Control 2 and the sign with the RR showed similar performance in the first lane. An key take away to these experiments, despite control 2 having impressive results, is the discrepancy between control 1 and 2. Considering that there will very likely be signs that will not be in ideal placements causing poor detection or no detection at all, a radar RR will alleviate that problem. There can be no guarantee that each SmartCode sign will be perfectly perpendicular to the road. In fact, it should be assumed that the signs will have some tilt and rotation to them from installation issues. By having a radar retro-reflective material on the sign, the significance of proper sign placement is greatly reduced, and the consistency of sign detection increases substantially. Additionally, the RR installed was only a small 2.5in reflector. The SC signs have approximately a $1.5ft^2$ of reflective area, so if a micro-scale RR is developed to place on entirety of the SC area, a massive increase in retro-reflectivity would result.