

**Probabilistic Data Association of RADAR and LiDAR Measurements for Overtaking
Maneuvers in Autonomous Racing**

by

Michaela Barksdale

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 12, 2026

Keywords: Autonomous Racing, Perception Sensors, Probabilistic Data Association

Copyright 2026 by Michaela Barksdale

Approved by

David M. Bevly, Chair, Bill and Lana McNair Distinguished Professor of Mechanical Engineering
Scott M. Martin, Assistant Research Professor of Mechanical Engineering
Ehsan Taheri, Associate Professor of Aerospace Engineering

Abstract

This thesis compares the tracking performance of estimation done using cluttered RADAR velocity measurements to estimated velocity based on LiDAR position measurements alone. The lead vehicle state estimation was done with a Probabilistic Data Association (PDA) Filter. This is a modified Kalman Filter that involves calculating the probability each detection originated from the target being tracked and using a combined measurement innovation. This prevents the track divergence that often occurs when using a filter, such as the Nearest Neighbor filter, that selects one measurement to use based on mahalanobis distance. Once the lead vehicle was tracked via the PDA filter, the state estimate was used to inform an attempt at an overtake via model predictive control.

With the rise in demand for vehicles with autonomous driving capabilities, the niche of autonomous racing research has also grown. One of the main tasks to perform in a racing scenario is an overtaking maneuver. To do this successfully, the opponent's vehicle position and velocity need to be known. This tracking is done using active perception sensors on board the ego vehicle, including LiDAR and RADAR. RADAR is capable of measuring relative velocity, where LiDAR is not, but its raw detections are noisy and cluttered.

Testing of the developed autonomy pipeline was done on both simulated and real TurtleBot4s. The PDA filter implementation was tested with a Lincoln MKZ and Kia Optima as well. This testing was meant to validate the method of vehicle tracking for future implementation on an autonomous indy racecar.

LiDAR outperformed RADAR when estimating lead vehicle position in all experiments. The velocity tracking results suggest that there may be a benefit to having a direct measurement of range rate from the RADAR sensor. This difference was not significant in the case of the TurtleBot4. The result of the full-scale vehicle tracking did indicate a greater improvement in velocity tracking at slow speeds. Highway and racing speeds were not tested.

Artificial Intelligence (AI) Use Disclosure Statement

In the preparation of this thesis, the following Artificial Intelligence (AI) tools were used: Claude (Anthropic). This tool was used primarily to assist with spelling and grammar checks, as well as code debugging. The author acknowledges full responsibility for the intellectual content of this work and has ensured that all AI-assisted sections have been reviewed and revised for accuracy and appropriate academic style. All AI-generated content was reviewed and validated for relevance, appropriateness, and accuracy before incorporation into the final document to maintain scholarly integrity of this research.

Digital Accessibility Disclosure Statement

In the preparation of this thesis, no digital accessibility tools were used. The author acknowledges full responsibility for the intellectual content of this work and has made a good faith effort to comply with digital accessibility requirements in publishing, wherein the nature of the content does not significantly change in order to do so. Furthermore, all content has been reviewed and revised to meet these requirements prior to final publication.

Acknowledgments

First, I would like to thank my parents, my sister Olivia, and my brothers William and Owen for always supporting and encouraging me. My parents gave me the opportunities that led to me becoming not only a first generation college student, but also the first in my family to attend graduate school.

I would also like to thank my committee and all of the professors at Auburn University who believed in me throughout both my undergraduate and graduate years. Without their support I would not have been able to reach the level of academic success I have achieved. Completing my master's degree would have also been far more difficult without my lab mates. I am grateful to have had a group of fellow engineers to learn, grow, and make memories with over the past few years.

Finally, I would like to thank my brothers and sisters in Christ that have walked alongside me as I have grown in my faith during my time in graduate school. My Wednesday night Bible study group and entire church family at Grace Auburn have been an incredible blessing from the Lord. They were with me every step of the way as I wrestled with what it meant to shift the motivation behind my engineering career from pride and selfish gain to glorifying God in all that I do.

"Whatever you do, work heartily, as for the Lord and not for men, knowing that from the Lord you will receive the inheritance as your reward. You are serving the Lord Christ."

Colossians 3:23-24

Table of Contents

Abstract	2
Artificial Intelligence (AI) Use Disclosure Statement	3
Digital Accessibility Disclosure Statement	4
Acknowledgments	5
List of Tables	9
List of Figures	11
List of Abbreviations	13
Chapter 1 Introduction	15
1.1 Background & Motivation	15
1.2 Prior Art	18
1.2.1 Autonomous Driving	18
1.2.2 Autonomous Racing	19
1.3 Research Contributions	20
1.4 Thesis Outline	21
Chapter 2 Technical Background	22
2.1 Perception Sensors	22
2.1.1 LiDAR	22
2.1.2 RADAR	23
2.2 Vehicle Models	23
2.2.1 Kinematic Unicycle Model	24
2.3 Coordinate Frames	25
2.3.1 Sensor Frames	25
2.3.2 Global Earth Centered Earth Fixed (ECEF) Frame	26
2.4 Autonomous Driving Pipeline Structure	26
Chapter 3 RADAR Measurement Filtering for Lead Vehicle Tracking	28

3.1	Measurement Validation	30
3.2	Nearest Neighbor Filter	30
3.3	Probabilistic Data Association Filter	30
3.4	Constant Velocity State Model	33
3.5	Measurement Models	34
3.5.1	RADAR	34
3.5.2	LiDAR	36
3.5.3	Sensor Fusion	37
Chapter 4	Overtaking via Model Predictive Control	38
4.1	Model Predictive Control Problem Formulation	38
4.1.1	System Dynamics	39
4.1.2	State Constraints	40
4.1.3	Control Constraints	40
4.1.4	Initial and Final State Conditions	42
Chapter 5	TurtleBot4 Experimentation	44
5.1	Methodology	44
5.1.1	Gazebo Simulation	44
5.1.2	Real Robots	46
5.2	Results	47
5.2.1	Static Sensor	47
5.2.2	Following	49
5.2.3	Overtaking	51
Chapter 6	Vehicle Experimentation	57
6.1	Methodology	57
6.2	Results	58
6.2.1	Static Sensor	58
6.2.2	Following	59

6.2.3	Overtaking	63
Chapter 7	Summary & Conclusions	67
7.1	Summary	67
7.2	Conclusions	67
7.3	Future Work	68
References		69
Appendix A	Tuned Filter and Controller Values	73
A.1	TurtleBot4	73
A.1.1	PDA Filter Tuning	73
A.1.2	MPC Controller Tuning	74
A.2	Vehicles	76
A.2.1	PDA Filter Tuning	76
Appendix B	Sensor Specifications	78
B.1	RADAR (Continental ARS430)	78
B.2	LiDAR (RPLIDAR A1M8)	78
B.3	LiDAR (Luminar Hydra H3)	79
Appendix C	Additional TurtleBot4 Overtaking Trial Results	81
C.1	LiDAR	81
C.1.1	Trial 1	81
C.1.2	Trial 2	83
C.1.3	Trial 3	85
C.1.4	Trial 4	87
C.2	RADAR	89
C.2.1	Trial 1	89
C.2.2	Trial 2	91
C.2.3	Trial 3	93
C.2.4	Trial 4	95

List of Tables

2.1	Kinematic Model Variables	25
3.1	Kalman Filter Variables	29
3.2	PDA Filter Variables	32
4.1	MPC State Constraint Variables	41
4.2	MPC Initial Condition Variables	42
5.1	Time to Overtake in Seconds For All TurtleBot Overtaking Trials	53
A.1	PDA Filter Process Noise Parameters	73
A.2	PDA Filter Measurement Noise Parameters	74
A.3	PDA Filter Association Parameters	74
A.4	MPC Horizon and Timestep	74
A.5	MPC State Cost Weights	75
A.6	MPC Control Cost Weights	75
A.7	MPC State Constraints	75
A.8	MPC Control Constraints	75
A.9	PDA Filter Process Noise Parameters	76
A.10	PDA Filter Measurement Noise Parameters	76
A.11	PDA Filter Association Parameters	77
B.1	RADAR Measurement Accuracy (Near Range) [42]	78
B.2	LiDAR Detection Range and Coverage [43]	78
B.3	LiDAR Measurement Resolution [43]	79
B.4	LiDAR Timing [43]	79
B.5	LiDAR Detection Range and Coverage [44]	79
B.6	LiDAR Measurement Resolution [44]	79

B.7	LiDAR Timing [44]	80
-----	-----------------------------	----

List of Figures

1.1	Autonomous Driving Levels [2]	16
1.2	Mahalanobis vs. Euclidean Distance [26]	20
2.1	Autonomous Indy Racecar	22
2.2	Unicycle Model [28]	24
2.3	Ego Vehicle Body Frame [29]	26
2.4	ECEF Coordinate Frame [29]	27
2.5	Modular Pipeline	27
3.1	Tracking Example With Measurement Origin Uncertainty [34]	33
4.1	Obstacle Avoidance Constraint Depiction	41
5.1	Standard TurtleBot4	44
5.2	Demonstration in Gazebo Simulation	45
5.3	TurtleBot4 with Continental RADAR	47
5.4	Position Estimates of Leader From Static Sensor	48
5.5	Velocity Estimates of Leader From Static Sensor	49
5.6	Position Estimates of Leader From Follower	50
5.7	Velocity Estimates of Leader From Follower	51
5.8	Position Estimates of Leader From Follower	53
5.9	Velocity Estimates of Leader From Follower	54
5.10	Position Estimates of Leader From Follower	55
5.11	Velocity Estimates of Leader From Follower	56
6.1	Autonomous Lincoln MKZ	57
6.2	X Position Estimate Errors From Static Sensor	58
6.3	Position Estimates of Kia From Static Sensor	60

6.4	Velocity Estimates of Kia From Static Sensor	61
6.5	X Position Estimate Errors From Follower Vehicle	61
6.6	Position Estimates of Kia From Following Lincoln	62
6.7	Velocity Estimates of Kia From Following Lincoln	63
6.8	X Position Estimate Errors from Overtaking Follower	64
6.9	Position Estimates of Kia From Overtaking Lincoln	65
6.10	Velocity Estimates of Kia From Overtaking Lincoln	66
C.1	Position Estimates of Leader From Follower	81
C.2	Velocity Estimates of Leader From Follower	82
C.3	Position Estimates of Leader From Follower	83
C.4	Velocity Estimates of Leader From Follower	84
C.5	Position Estimates of Leader From Follower	85
C.6	Velocity Estimates of Leader From Follower	86
C.7	Position Estimates of Leader From Follower	87
C.8	Velocity Estimates of Leader From Follower	88
C.9	Position Estimates of Leader From Follower	89
C.10	Velocity Estimates of Leader From Follower	90
C.11	Position Estimates of Leader From Follower	91
C.12	Velocity Estimates of Leader From Follower	92
C.13	Position Estimates of Leader From Follower	93
C.14	Velocity Estimates of Leader From Follower	94
C.15	Position Estimates of Leader From Follower	95
C.16	Velocity Estimates of Leader From Follower	96

List of Abbreviations

ADAS	Advanced Driver Assistance Systems
DOF	Degrees of Freedom
ECEF	Earth Centered Earth Fixed
EKF	Extended Kalman Filter
FMCW	Frequency Modulated Continuous Wave
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IAC	Indy Autonomous Challenge
IMM	Interacting Multiple Model
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
JPDA	Joint Probabilistic Data Association
LiDAR	Light Detection and Ranging
MHT	Multiple Hypothesis Tracking
MPC	Model Predictive Control
NN	Nearest Neighbor
NNF	Nearest Neighbor Filter
PDA	Probabilistic Data Association

PDAF Probabilistic Data Association Filter

RADAR Radio Detection and Ranging

RMSE Root Mean Square Error

ROS Robot Operating System

SLAM Simultaneous Localization and Mapping

ToF Time-of-Flight

UKF Unscented Kalman Filter

Chapter 1

Introduction

1.1 Background & Motivation

The field of autonomous racing has progressed significantly over the past two decades since the start of the DARPA Grand Challenge in 2004. While the approaches for achieving autonomous behavior have evolved, the main tasks to be tackled have largely remained the same. These tasks include perception, planning, and control [1]. The Society of Automotive Engineers has defined the standard for 6 different levels of autonomous driving, ranging from minimal driver assistance features, all the way up to full autonomy that doesn't require the presence of a human driver [2]. More specific comparison and details regarding these levels can be found in Figure 1.1. From a simple blind spot warning at Level 0 to full unconditional driverless capabilities at Level 5, perception sensors are utilized throughout the range of features. Planning and control steps often rely on perception in order to navigate safely through unknown and changing environmental conditions. Radio Detection and Ranging (RADAR), Light Detection and Ranging (LiDAR), and camera are the 3 most commonly used perception sensors for autonomous driving scenarios, each with its own capabilities [3]. The best way to leverage each sensor is still an active area of research.

Choosing which sensor to use for which specific task can depend on the driving scenario. One such scenario that has been studied is autonomous racing. Several different research competitions exist as an opportunity to test and compete with driverless racecars including F1Tenth, Formula Student Driverless, Roborace, and the Indy Autonomous Challenge (IAC) [4]. The IAC, successor to the DARPA Grand Challenge, provides university teams with an Indy Lights car that has been modified with hardware necessary for autonomy including sensors and a computer. This challenge has historically consisted mainly of time trials as opposed to head-to-head racing. In order to shift towards multicar competition, teams will need to determine the



SAE J3016™ LEVELS OF DRIVING AUTOMATION™

Learn more here: [sae.org/standards/content/J3016_202104](https://www.sae.org/standards/content/J3016_202104)

Copyright © 2021 SAE International. The summary table may be freely copied and distributed AS-IS provided that SAE International is acknowledged as the source of the content.

		SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?		You <u>are</u> driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You <u>are not</u> driving when these automated driving features are engaged – even if you are seated in “the driver's seat”		
		You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
Copyright © 2021 SAE International.							
What do these features do?		These are driver support features			These are automated driving features		
		These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features		• automatic emergency braking • blind spot warning • lane departure warning	• lane centering OR • adaptive cruise control	• lane centering AND • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 1.1: Autonomous Driving Levels [2]

best way to use the RADAR, LiDAR, and cameras that are installed on the cars.

Perception sensors are useful for localization and mapping, object detection, and target tracking. Localization can be done via GPS and inertial measurements and does not require the use of additional sensors. However in GPS denied environments it may be beneficial to utilize an alternative solution such as LiDAR SLAM. Since the IAC has been mainly focused on time trials, localization has been the primary concern among these perception related tasks. In a multicar scenario object detection and target tracking become much more relevant. In order to pass an opponent, an accurate estimate of the other vehicle's position, velocity, and attitude is needed.

This thesis focuses on the development of a testing demonstration to evaluate the performance of RADAR and LiDAR in opponent vehicle tracking to inform an overtaking attempt. The demonstration was implemented in its entirety with TurtleBot4s and the tracking was further tested with full-scale passenger vehicles. Ultimately this thesis is meant to support the Auburn University Indy Autonomous Challenge Team, but other testbeds were used as they were more accessible for development and testing. LiDAR has been used to a much greater extent in the Indy Autonomous Challenge thus far and this thesis investigates the degree to which incorporating RADAR would benefit the existing LiDAR work that has been done.

The cameras are the only passive sensor, do not natively measure depth, are susceptible to changes in weather conditions, and have a large data stream that is more difficult to process real-time [5]. Utilizing cameras in autonomous driving is beneficial as they are the best of the three sensors when it comes to object classification and pedestrian detection. However these advantages are far more relevant to a traditional driving scenario than autonomous racing. Typical driving involves recognizing traffic signs and signals as well as avoiding pedestrians, neither of these challenges are a concern in autonomous racing. The weaknesses associated with the camera make them the worst option when it comes to target tracking, therefore the camera is left out of the comparison presented in this thesis.

1.2 Prior Art

1.2.1 Autonomous Driving

A significant amount of existing work with perception sensors has been specific to autonomous driving as opposed to racing as this is an older and better-funded field of research. RADAR has proven to remain functional in various weather conditions which, paired with its direct measurement of radial velocity, makes it an ideal sensor choice for Advanced Driver Assistance Systems (ADAS) [6, 7]. The tasks that RADAR is typically used for include adaptive cruise control, park assist, blind spot detection, and collision warnings [8]. These tasks all require the use of an object detection algorithm, an example that focuses mainly on the use of RADAR is RODNet [9]. While a piece of this network involves fusion with camera, the object detection portion is done solely with RADAR. Where vision based detection would begin to degrade, RODNet does not due to RADAR's ability to operate even in low-visibility conditions. While RADAR is commonly used in vehicles for detection it is not the best option for classification. While this can be done with RADAR as previously demonstrated by evaluating the performance of various clustering algorithms for recognition of pedestrians, cyclists and other vehicles with RADAR alone [10], sensor fusion is often necessary in order to improve the results. The LiDAR sensor has higher resolution and precise range measurements which allow for better estimation of an object's orientation and shape than with RADAR [11]. However, when tested in adverse weather it did not perform as well [12]. PointPillars is an object detection and classification network that was developed using only LiDAR and performed well when compared to various other methods [13]. This demonstrates that LiDAR is a reasonable single sensor choice for identifying vehicles, pedestrians, and cyclists on the road.

Fusing LiDAR and RADAR with each other, or with a camera, allows for sensor strengths to be leveraged while supplementing their weaknesses. Pairing RADAR's weather agnostic function and radial velocity measurement, LiDAR's high measurement resolution, and camera's color differentiation often leads to higher performance capabilities of any given algorithm as a whole than if only one sensor was being used [14]. A dominant combination for object detection has

been LiDAR and camera. One example is Liu et al.'s real-time algorithm that performed better than many state of the art methods including PointPillars [15]. Another instance of this specific sensor combination is BEVFusion which not only does object detection, but also segmentation [16]. Although LiDAR and camera are more commonly fused, work has been done with RADAR and camera fusion as well. Introducing RADAR improves on detection done with camera alone, but this sensor combination is really enhanced in object tracking due to RADAR's radial velocity measurement [17]. Similarly, fusing RADAR with LiDAR improves upon object detection done with LiDAR alone. Wang et al. found that aiding LiDAR with RADAR fares better than PointPillars as well [18]. Ultimately though, as with camera fusion, the main benefit of introducing RADAR to a system that already has LiDAR is for the purpose of target tracking which also lends itself to predicting future behavior of moving targets as seen with LiRaNet [19].

1.2.2 Autonomous Racing

Although the research done on sensor fusion in autonomous driving is relevant to the research area of autonomous racing, the findings can not be directly translated. While the tasks to be accomplished are similar, there are several differences in the assumptions that can be made. Detection of pedestrians and cyclists is a worthwhile task to research for driving, but is irrelevant to autonomous racing. Additionally it is safe to assume that any object moving at a similar speed is another racecar, which simplifies the object classification task. In many cases, such as in the IAC, autonomous racing is only done in good weather, eliminating the concerns related to rain or fog.

IAC teams have already done work with detection and tracking of an opponent vehicle. Most of the existing work leans predominantly on the use of the LiDAR over the other perception sensors, with one team even using PointPillars [20]. Two teams, TUM and PoliMOVE, both utilized an extended Kalman filter for state estimation with slightly different nearest neighbor data association [21–23]. The former used the euclidean distance while the latter used mahalanobis distance. Euclidean distance does not account for the distribution of the data in its calculation,

but mahalanobis does [24]. As seen in Figure 1.2 this can cause certain measurements to be considered outliers even if they are not. Both methods use the measurement with the least distance from the predicted measurement to correct the state and covariance predictions in the filter. These two teams fuse RADAR with LiDAR to leverage RADAR's ability to measure range rate. KAIST used a more sophisticated estimation filter, IMM-UKF-PDAF, and fused LiDAR with RADAR [25]. None of these prior works utilize the RADAR's velocity measurement or offer an in depth comparison between the performance of different sensors as will be done in this thesis.

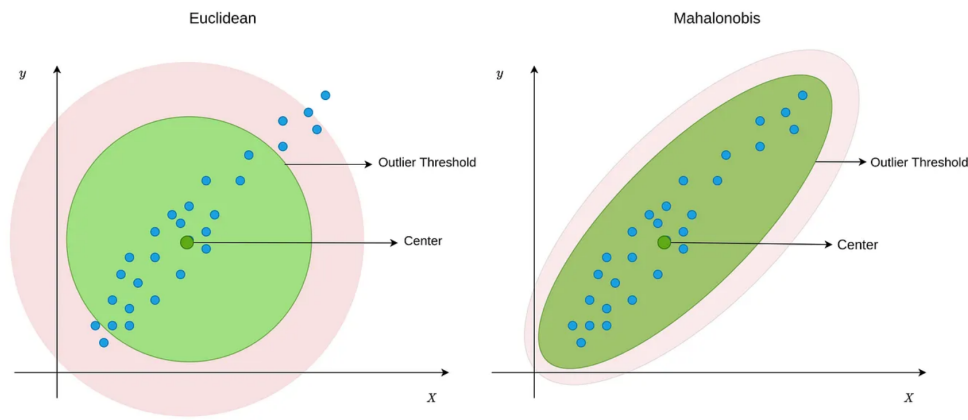


Figure 1.2: Mahalanobis vs. Euclidean Distance [26]

1.3 Research Contributions

This thesis offers a quantitative comparison of the performance of active perception sensors in state estimation of a lead vehicle in autonomous racing scenarios for the purpose of informing an attempt to overtake. Existing work has fused LiDAR and RADAR to perform target tracking tasks for the IAC, and brief comparisons have been presented, but none have been done on the same testbeds or to the same extent as this thesis. The main contributions of this thesis are listed below.

- Develop a modular autonomy pipeline involving a Probabilistic Data Association Filter for target tracking and Model Predictive Control (MPC) for overtaking as a method for testing perception sensor performance

- Implement the developed autonomy on TurtleBot4s both in simulation and with hardware
- Implement the target tracking filter real-time on full-scale vehicles
- Compare LiDAR and RADAR results from all testing scenarios to determine the extent of the benefit of RADAR's range rate measurement

1.4 Thesis Outline

The following six chapters dive deeper into the development pieces of the autonomy pipeline, present the various experimentation methods and results, then outlines the conclusions to be drawn and possible future work. Chapter 2 offers more background information including descriptions of the function of the sensors, explanations of typical autonomy pipelines, and presentation of the different coordinate frames used for this thesis. Chapter 3 describes the methods of measurement filtering used to accomplish target tracking. Chapter 4 outlines the planning and control involved in overtaking, which was done using MPC. Chapter 5 presents the experimental set up and results of the implementation on the simulated and physical TurtleBots. Chapter 6 is similar, but for the target tracking implementation on the passenger vehicles. The final chapter concludes and summarizes the results of the comparison performed in the thesis and suggests further work that could be done to investigate how to best utilize different perception sensors in autonomous racing.

Chapter 2

Technical Background

2.1 Perception Sensors

The AV24 currently being used in the Indy Autonomous Challenge comes equipped with three different types of perception sensors; 6 Mako cameras, 3 Luminar LiDARs, and 2 Continental RADARs. The Auburn University AV24 autonomous indy racecar is shown in Figure 2.1. Since this thesis is focused on active perception sensors, the following sections will cover the ways in which the LiDAR and RADAR function.

2.1.1 LiDAR

Traditionally, LiDAR sensors emit a laser and measure the time it takes for it to return to the sensor, t , then convert this time to a range measurement, R , by equation (2.1) using the speed of light, c .

$$R = \frac{c \Delta t}{2}, \quad c = 3 \times 10^8 \text{ m/s} \quad (2.1)$$

The value is divided by two to account for the light traveling to the target and back to the sensor. This is known as time-of-flight (ToF) LiDAR and often results in millimeter level accuracy, and therefore strong relative position solutions. ToF LiDARs, which are the most commonly used in



Figure 2.1: Autonomous Indy Racecar

automotive applications, do not measure the frequency of light, meaning that they cannot detect doppler shift, which would be necessary for directly measuring velocity. It is worth mentioning that emerging technology allows for frequency modulated continuous wave (FMCW) LiDAR, which would address this shortcoming of LiDAR and possibly eliminate the need for RADAR fusion [27]. However, these sensors are more expensive and currently not present in most vehicles, including IAC racecars.

2.1.2 RADAR

The technology for FMCW RADAR is more mature and accessible than FMCW LiDAR. FMCW involves transmitting a radio wave with frequency that increases linearly, also known as chirp. The frequency difference between signal transmission and return can be used to determine the difference in time, which can be converted to range just like with a ToF sensor. The Doppler shift of the wave allows for the relative velocity measurement that ToF sensors do not provide. This is a simplified explanation of how FMCW technology works; Patole et al. dive deeper into its functionality, as well as other types of automotive RADAR sensors [8]. Typical ADAS systems, and the IAC vehicles, are equipped with FMCW RADARs. The Continental RADARs report range, range rate, azimuth, and elevation directly. All of the FMCW signal processing is done by the sensor itself. However these raw RADAR measurements need further processing as they are cluttered and noisy. This was handled with the use of a Probabilistic Data Association (PDA) filter which is further explained in Chapter 3.

2.2 Vehicle Models

In addition to measurements, estimation of the opponent vehicles state using a bayesian filter requires a model of the dynamics. The TurtleBots can be represented with a unicycle model. A constant velocity version of the model is used for this work.

2.2.1 Kinematic Unicycle Model

A unicycle model is characterized by having two degrees of freedom and no side-slip, that is, the vehicle always travels in the direction it is facing. This is true for a TurtleBot, which is why this model was used for the MPC implementation done in Chapter 5. A visual depiction of this model is shown in the diagram in Figure 2.2. The nonlinear discrete state update in Equation (2.2) is a function of the previous states and control inputs which are given in Equation (2.3). Table 2.1 describes the variables present in the unicycle model figure and equations.

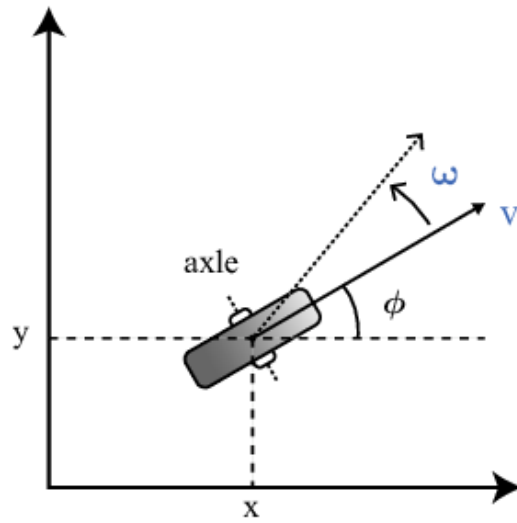


Figure 2.2: Unicycle Model [28]

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} x_k + v_k \cos(\psi_k) \Delta t \\ y_k + v_k \sin(\psi_k) \Delta t \\ \psi_k + \omega_k \Delta t \end{bmatrix} \quad (2.2)$$

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \quad (2.3)$$

Table 2.1: Kinematic Model Variables

Variable	Description
k	Current Time Step
$\mathbf{x}$	State Vector
$\mathbf{u}$	Control Inputs
x	X Position
y	Y Position
ϕ	Yaw
v	Forward Velocity
ω	Yaw Rate
Δt	Time Difference

2.3 Coordinate Frames

There are two main types of reference frames used in this thesis for pose estimates. Those include sensor and global coordinate frames. Both frames are described in more detail below, and example diagrams of each from Paul Groves’ Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems [29] are provided.

2.3.1 Sensor Frames

A body frame is a right-handed coordinate frame centered on the ego vehicle or a given sensor. An example of a body frame is depicted in Figure 2.3. Since perception sensors measure relative to the sensor itself, the frame used for the filter solutions in this thesis is the RADAR’s sensor frame. In the case of the RADAR the x axis points straight forward out of the sensor, the y axis is to the left, and the z axis is orthogonal to the other two and pointing straight up. For the experiments performed in this thesis the lead vehicle’s position and velocity are estimated relative to this sensor frame. In the case of the physical TurtleBots the LiDAR was treated as the reference for evaluating the filter’s performance. The RADAR and LiDAR were mounted close enough together on the TurtleBots that the lever arms were able to be disregarded, and both were considered to coincide with the center of the robot. The same was the case on the Lincoln MKZ in Chapter 6. The LiDAR frame on the TurtleBot is 90 degrees offset from that of the RADAR, with the positive y axis pointing forward and the positive x axis to the right. These measurements

were rotated to the RADAR frame before being used to update the filter.

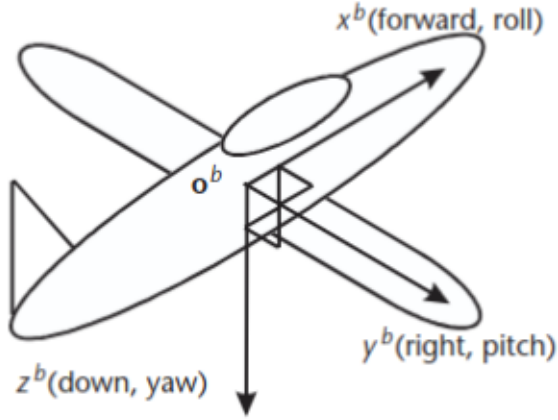


Figure 2.3: Ego Vehicle Body Frame [29]

2.3.2 Global Earth Centered Earth Fixed (ECEF) Frame

The ECEF frame is used for GPS localization of the vehicles in Chapter 6, which is used as a truth reference comparison for the PDA filter's results in this case. This frame's origin is at the center of the earth. The x and y axes both intersect with the equator and align with 0 degrees and 90 degrees east longitude respectively. The z axis goes through the north pole. This is shown in Figure 2.4. The ECEF measurements were converted to the sensor frame for the purpose of direct comparison with filter results.

2.4 Autonomous Driving Pipeline Structure

Traditionally, each task in an autonomous driving framework is performed separately, and the result of one task is fed into the next. These tasks typically include perception, path planning, and control. Recently, with the onset of artificial intelligence and machine learning, it has become possible to go straight from sensor measurements to control outputs. The former is known as a modular approach, and the latter is called end-to-end control[30]. Although end-to-end pipelines are the state-of-the-art, this thesis uses a modular approach, depicted in Figure 2.5,

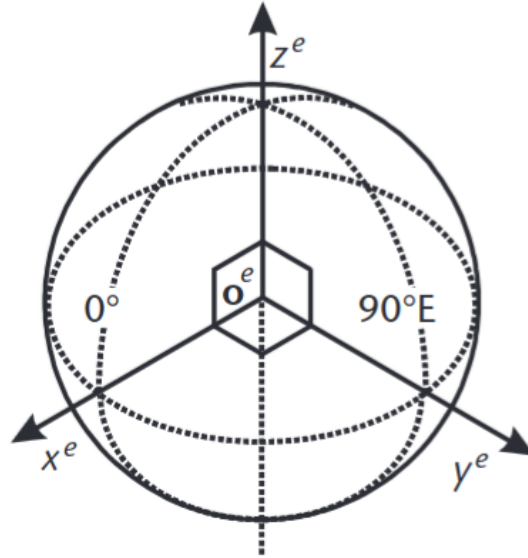


Figure 2.4: ECEF Coordinate Frame [29]

in order to evaluate the effectiveness of the PDA filter with RADAR measurements. The estimate from the PDA filter is then fed into MPC which handles both the path planning and control. The following two chapters cover the PDA filter and MPC respectively.

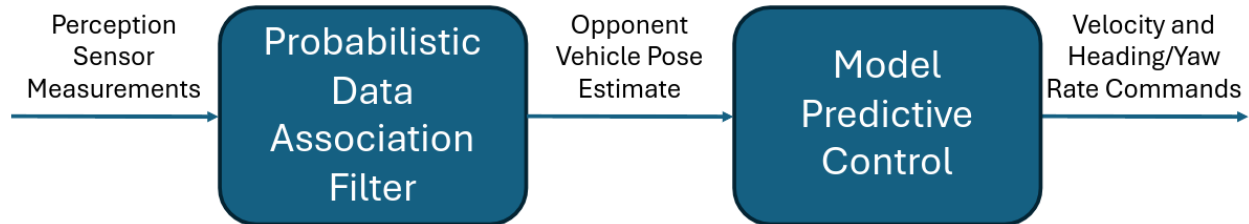


Figure 2.5: Modular Pipeline

Chapter 3

RADAR Measurement Filtering for Lead Vehicle Tracking

In order to successfully follow and pass another racecar it must first be identified and tracked by the ego vehicle. This requires the use of perception sensors to detect objects present in the environment. Measurements from these sensors can be filtered in order to produce an estimate of the lead vehicle's pose. The work presented in this thesis focuses on the completion of the target tracking step. Therefore, the filters are initialized with the known truth state of the lead vehicle, assuming that the object detection piece has already been completed.

A major drawback to utilizing RADAR is that the raw measurements are not usable as they are cluttered with noise, detections of other objects in the area, and ghost detections. A filter is needed to eliminate noise and associate measurements with the desired target.

Typical RADAR filters include the Nearest Neighbor (NN), Probabilistic Data Association (PDA), Joint Probabilistic Data Association (JPDA), and Multiple Hypothesis Tracking (MHT). The NN filter is the simplest of these and works best in low clutter environments with one target. The PDA filter is more complex than the NN filter but is able to track targets even in the presence of clutter. The JPDA Filter and MHT are far more computationally expensive and are often needed when there is more than one target that needs to be tracked.

The nearest neighbor filter and probabilistic data association filter, are both modified versions of a typical Kalman filter, which consists of the equations below.

Time Update:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (3.1)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.2)$$

Measurement Update:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (3.3)$$

$$\hat{z}_k = h(\hat{x}_k^-) \quad (3.4)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{z}_k) \quad (3.5)$$

$$P_k = (I - K_k H) P_k^- \quad (3.6)$$

The variables present in these equations are described in Table 3.1. The PDA filter was selected for the work done in this thesis, since there is only one target being tracked. It was chosen over the NN filter as it performs better with high measurement clutter.

Table 3.1: Kalman Filter Variables

Variable	Description
k	Current Time Step
$\hat{x}$	State Estimate Vector
A	State Transition Matrix
B	Control Matrix
u	Control Input Vector
P	State Covariance Matrix
Q	Process Noise Covariance Matrix
K	Kalman Gain
H	Measurement Matrix
R	Measurement Noise Covariance Matrix
z	Measurement Vector
I	Identity Matrix

The time update remains identical across all three filters, with the modifications being present in the measurement update portion. With the presence of origin uncertainty in RADAR measurements, choosing which measurement to use in the Kalman filter is not straight-forward, and this issue is addressed by both the NNF and PDAF.

3.1 Measurement Validation

An important step when tracking targets in clutter is measurement validation. This involves comparing each mahalanobis distance, calculated as

$$d_{k_i} = \sqrt{(z_{k_i} - \hat{z}_k)^T (HP_k^- H^T + R)^{-1} (z_{k_i} - \hat{z}_k)} \quad (3.7)$$

with a predetermined validation gate value, γ . Any measurement with too large of a mahalanobis distance is not considered as it is unlikely to have originated from the target of interest.

3.2 Nearest Neighbor Filter

After the measurement validation step, the nearest neighbor filter consists of the exact same equations as a typical Kalman filter. Measurement selection for Equation (3.5) is done by calculating the mahalanobis distance for each RADAR detection and using the one with the minimum value of d_k . An assumption is made that the measurement corresponding with the minimum mahalanobis distance, also known as the nearest neighbor, is from the correct target. This filter can be effective, especially in low clutter environments, but if the nearest neighbor measurement happens to originate from an incorrect target, track divergence can occur, i.e. the filter tracks something other than the actual target [31].

3.3 Probabilistic Data Association Filter

The probabilistic data association filter is more complex and has two components that are not present in a traditional Kalman filter or NNF. These are calculation of association probabilities and updating the covariance matrix to account for measurement origin uncertainty. The purpose of these additional steps is to consider all validated measurements when updating the state estimate vector and covariance Matrix as opposed to selecting just one. This helps to prevent the issue seen in the NNF where an incorrect measurement is selected and the RADAR track diverges from the truth [32]. This filter was developed by Bar-Shalom et al. [33].

For each measurement an association probability, β , is calculated as

$$\beta_{k_i} = \frac{L_{k_i}}{1 - P_D P_G + \sum_{j=1}^{m_k} L_{k_j}} \quad (3.8)$$

which is the likelihood that any given measurement corresponds with the target of interest. In the above equation, L is the ratio of the likelihood that the measurement originated from the target to the likelihood that is originated from clutter, P_D is the probability the target is detected, and P_G is the probability that the measurement falls within the validation region that corresponds with the chosen value of γ . This is based on a 2 dimensional chi-squared distribution. The probability that none of the measurements can be associated with the desired target is also calculated as

$$\beta_{k_0} = \frac{1 - P_D P_G}{1 - P_D P_G + \sum_{j=1}^{m_k} L_{k_j}} \quad (3.9)$$

and this value is used in the update of the covariance matrix.

A likelihood ratio, L , is calculated for each measurement at each time step via the following equations,

$$L_{k_i} = \frac{\mathbf{N}_{\mathbf{k}_i} P_D}{\lambda} \quad (3.10)$$

$$\mathbf{N}_{\mathbf{k}_i} = \frac{1}{2\pi\sqrt{|S_k|}} \exp\left(-\frac{1}{2}(z_{k_i} - \hat{z}_k)^T S_k^{-1} (z_{k_i} - \hat{z}_k)\right) \quad (3.11)$$

$$S_k = H P_k^- H^T + R \quad (3.12)$$

where $\mathbf{N}$ is the probability density function of a multivariate normal distribution of the measurement innovation and S is the measurement innovation covariance. The clutter is modeled as a Poisson distribution and λ represents the density of the clutter.

After the association probabilities are calculated, the measurement update is performed. Equations (3.3 - 3.4) are done the same as with a standard Kalman filter. The association proba-

bilities are used to calculate a combined measurement innovation, $\mathbf{z}$.

$$\mathbf{z}_k = \sum_{i=1}^m \beta_{k_i} (z_{k_i} - \hat{z}_k) \quad (3.13)$$

This combined measurement innovation is used in Equation (3.5). The state covariance matrix for the PDA filter is calculated as shown below.

$$P_k = \beta_{k_0} P_k^- + (1 - \beta_{k_0}) P_k^c + \tilde{P}_k \quad (3.14)$$

$$P_k^c = P_k^- - K_k S_k K_k^T \quad (3.15)$$

$$\tilde{P}_k = K_k [\sum_{i=1}^m \beta_{k_i} (z_{k_i} - \hat{z}_k)(z_{k_i} - \hat{z}_k)^T - \mathbf{z}_k \mathbf{z}_k^T] K_k^T \quad (3.16)$$

Finally, a summary of the variables used in the PDA filter are given in Table 3.2.

Table 3.2: PDA Filter Variables

Variable	Description
β	Association Probability
L	Likelihood Ratio
P_D	Detection Probability
P_G	Gate Probability
m	Number of Detections
N	PDF of Normal Distribution
λ	Spatial Density of Clutter
S	Measurement Innovation Covariance Matrix
$\mathbf{z}$	Combined Measurement Innovation
P_k	Updated Covariance Matrix

The possibility that none of the measurements are from the correct target has to be accounted for. This is done with the first term in Equation (3.14). The update to the covariance matrix if one of the measurements does originate from the desired target is represented by P_k^c . The measurement origin uncertainty is accounted for with $\tilde{P}$.

Figure 3.1 depicts the benefit of the PDA filter. The blue line represents an existing track of a state being estimated by a given filter. The image on the left shows three possible measurements represented by the green, yellow, and red dots. Along with the measurements, the resulting track,

covariance, and association probability for each are shown by the lines, ellipses, and numbers respectively. Instead of just relying on the measurement with the minimum mahalanobis distance as the NN filter would, in this case the one shown in yellow, all three possible measurements are taken into account in the PDA filter. This results in the updated blue track shown on the right, as opposed to tracking the yellow line. The three example association probabilities do not sum to one; this is because the probability that none of these measurements originated from the target being tracked is also accounted for in this example. Since the perception sensors used in this thesis produce cluttered measurements, the PDA filter is a better choice than the NN filter.



Figure 3.1: Tracking Example With Measurement Origin Uncertainty [34]

3.4 Constant Velocity State Model

In the time update of the filter a constant velocity model was used for the state transition matrix. The states for the model are shown in Equation (3.17).

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y \end{bmatrix}^T \quad (3.17)$$

The corresponding state transition matrix (A) and process noise covariance (Q) are shown in Equations (3.18-3.19).

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

$$Q = \begin{bmatrix} q_x \frac{\Delta t^3}{3} & q_x \frac{\Delta t^2}{2} & 0 & 0 \\ q_x \frac{\Delta t^2}{2} & q_x \Delta t & 0 & 0 \\ 0 & 0 & q_y \frac{\Delta t^3}{3} & q_y \frac{\Delta t^2}{2} \\ 0 & 0 & q_y \frac{\Delta t^2}{2} & q_y \Delta t \end{bmatrix} \quad (3.19)$$

3.5 Measurement Models

3.5.1 RADAR

The Continental ARS430 RADAR was used for experimentation with both the TurtleBots and the test vehicles. The range, range rate, and azimuth measurements were used to track x and y positions and velocities. This resulted in a nonlinear measurement model, meaning a Jacobian would be needed to be used for each measurement update step of the PDA Filter, making it an EKF/PDAF instead of a standard linear Kalman filter adaptation. The measurement state vector (z) and measurement observation model are shown in Equation (3.20) and Equation (3.21) respectively.

$$z = \begin{bmatrix} r & \dot{r} & \theta \end{bmatrix}^T \quad (3.20)$$

$$h(\mathbf{x}) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \frac{xv_x + yv_y}{\sqrt{x^2 + y^2}} \\ \text{atan2}(y, x) \end{bmatrix} \quad (3.21)$$

Bar-Shalom et al. found that converting polar position measurements to cartesian coordinates before tracking leads to better results [35]. This same conversion can not be done for velocity. The resulting converted state vector(z') and measurement observation model are shown in Equation (3.22) and Equation (3.23) respectively.

$$z' = \begin{bmatrix} x & y & \dot{r} \end{bmatrix}^T \quad (3.22)$$

$$h'(\mathbf{x}) = \begin{bmatrix} x \\ y \\ \frac{xv_x + yv_y}{\sqrt{x^2 + y^2}} \end{bmatrix} \quad (3.23)$$

The first two rows of the observation matrix (H) become linear by nature but the last row still needs to be linearized. The resulting linearized observation matrix is shown in Equation (3.24).

$$H = \frac{\partial h}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \left(\frac{v_x}{r} - \frac{x\dot{r}}{r^2}\right) & \frac{x}{r} & \left(\frac{v_y}{r} - \frac{y\dot{r}}{r^2}\right) & \frac{y}{r} \end{bmatrix} \quad (3.24)$$

A conversion also needs to be applied to the measurement covariance matrix(R) to properly reflect the converted states. The conversion from polar to cartesian coordinates is given by Equations (3.25 - 3.26).

$$x = r\cos(\theta) \quad (3.25)$$

$$y = r\sin(\theta) \quad (3.26)$$

The Jacobian matrix of this conversion shown in Equation (3.27).

$$J = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \\ 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

The above Jacobian is then used to calculate the true covariance matrix as shown in Equation (3.28). Due to the relatively small angles present throughout most of the maneuver J was assumed to be an identity matrix, and the filter functioned well without the need for linearization at each time step.

$$R = J \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_{\dot{r}}^2 \end{bmatrix} J^T = \begin{bmatrix} \sigma_r^2 \cos^2 \theta + r^2 \sigma_\theta^2 \sin^2 \theta & (\sigma_r^2 - r^2 \sigma_\theta^2) \sin \theta \cos \theta & 0 \\ (\sigma_r^2 - r^2 \sigma_\theta^2) \sin \theta \cos \theta & \sigma_r^2 \sin^2 \theta + r^2 \sigma_\theta^2 \cos^2 \theta & 0 \\ 0 & 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \quad (3.28)$$

3.5.2 LiDAR

The ToF LiDAR sensors present on both the TurtleBot4s and the testing vehicles for Chapter 6 measure ranges and azimuths, but do not report range rates. Therefore in the LiDAR only filters velocities are unmeasured states. The previously described RADAR measurement model is modified as shown in Equation (3.29).

$$z = \begin{bmatrix} x & y \end{bmatrix}^T, \quad h(\mathbf{x}) = H\mathbf{x}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.29)$$

A conversion to the measurement covariance matrix (R) is needed for the LiDAR as well. The Jacobian (J) and converted covariance matrix are shown in Equation (3.30) and Equation (3.31) respectively.

$$J = \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix} \quad (3.30)$$

$$R = J \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} J^T = \begin{bmatrix} \sigma_r^2 \cos^2 \theta + r^2 \sigma_\theta^2 \sin^2 \theta & (\sigma_r^2 - r^2 \sigma_\theta^2) \sin \theta \cos \theta \\ (\sigma_r^2 - r^2 \sigma_\theta^2) \sin \theta \cos \theta & \sigma_r^2 \sin^2 \theta + r^2 \sigma_\theta^2 \cos^2 \theta \end{bmatrix} \quad (3.31)$$

The same small angle and identity matrix Jacobian assumptions described in Section 3.5.1 are also applicable for this model.

3.5.3 Sensor Fusion

The fused filter case uses the LiDAR range measurements and RADAR range rate measurements. Two separate measurement updates were performed, first the RADAR measurements were used to update the velocity states and covariances and then the LiDAR measurements to update position states and covariances. One singular time update was done the same as the filters that used only one of the two sensors. The tuned values of Q and R for each experiment can be found in Appendix A.

Chapter 4

Overtaking via Model Predictive Control

In addition to evaluating the accuracy of the pose estimation when a direct range rate measurement is present by comparing to truth, the practical necessity of a direct range rate measurement can be evaluated by using the pose estimates to inform an overtaking maneuver. Since overtaking is a main objective in autonomous racing, the capability of an autonomous system to perform this task well is necessary. If a pose estimate that is more computationally efficient, but slightly less accurate, still consistently results in a successful overtake, this trade-off may be allowable. Recent autonomous overtaking research relies on various artificial intelligence methods [36]. For the sake of validating the pose estimation done with the probabilistic data association filter, the most cutting-edge method is not necessary. Instead a standard MPC algorithm that combines path planning and control into one step was implemented.

4.1 Model Predictive Control Problem Formulation

Design of MPC requires setting up an optimal control problem with an objective function and constraints. An existing solver, OSQP-MPC, was used for the MPC done in this thesis [37]. This solver uses the objective function shown in Equation (4.1).

$$J = \sum_{k=1}^N (\mathbf{x}_k - \mathbf{x}_{\text{ref}})^T \mathbf{Q}_k (\mathbf{x}_k - \mathbf{x}_{\text{ref}}) + \sum_{k=1}^{N-1} \mathbf{u}_k^T \mathbf{R}_k \mathbf{u}_k \quad (4.1)$$

By minimizing this quadratic function the control effort and the error between the true and desired states is minimized. The above cost function includes various constraints to be optimized about which are discussed in the next section.

4.1.1 System Dynamics

The dynamic model used for path planning is a linearized and rotated version of the unicycle model presented in Chapter 2. The state space model form is simply

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \quad (4.2)$$

where the states and inputs are defined in Equations (4.3-4.4).

$$\mathbf{x}_k = \begin{bmatrix} x_{rel} \\ y_{rel} \\ \psi \\ v_{rel} \end{bmatrix}_k, \quad (4.3)$$

$$\mathbf{u}_k = \begin{bmatrix} v_{follow} \\ \omega_{follow} \end{bmatrix}_k \quad (4.4)$$

The discrete state transition matrix (A) and input matrix (B) are given in Equations (4.5-4.6).

$$\mathbf{A}_k = \begin{bmatrix} 1 & \bar{\omega}_{follow} \Delta t & -\bar{v}_{rel} \sin(\bar{\psi}) \Delta t & \cos(\bar{\psi}) \Delta t \\ -\bar{\omega}_{follow} \Delta t & 1 & -\bar{v}_{rel} \cos(\bar{\psi}) \Delta t & -\sin(\bar{\psi}) \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$$\mathbf{B}_k = \begin{bmatrix} -\Delta t & \bar{y}_{rel} \Delta t \\ 0 & -\bar{x}_{rel} \Delta t \\ 0 & \Delta t \\ 0 & 0 \end{bmatrix} \quad (4.6)$$

In order to calculate the values for the matrices above a linearization point needed to be selected at each time step. The values from the last planned trajectory at the given horizon step and the

current yaw rate measurement were used for this calculation.

4.1.2 State Constraints

The state constraints consist of the physical limits on the first 3 states and an obstacle avoidance constraint to avoid the lead vehicle during the overtake. The fourth constraint creates a half-space boundary tangent to a circular buffer region centered at the leader's estimated position and normal to the line of sight vector between the two robots. A visual representation of this is given in Figure 4.1. The state constraint inequality and matrix ($\mathbf{C}_k$) are shown in Equations (4.7-4.8).

$$\mathbf{lb}_{x,k} \leq \mathbf{C}_k \mathbf{x}_k \leq \mathbf{ub}_{x,k} \quad (4.7)$$

$$\mathbf{C}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hat{n}_{k,x} & \hat{n}_{k,y} & 0 & 0 \end{bmatrix} \quad (4.8)$$

Values for the lower ($\mathbf{lb}_{x,k}$) and upper ($\mathbf{ub}_{x,k}$) bounds are provided in Table A.7 found in Appendix A.

The half-space constraint described above requires components of the unit vector along the line of sight to be present in the last row of the constraint matrix. The calculation of these components is show in Equation 4.9.

$$\hat{n}_{k,x} = \frac{\bar{x}_{rel,k}}{\bar{d}_k}, \quad \hat{n}_{k,y} = \frac{\bar{y}_{rel,k}}{\bar{d}_k}, \quad \bar{d}_k = \sqrt{\bar{x}_{rel,k}^2 + \bar{y}_{rel,k}^2} \quad (4.9)$$

Table 4.1 summarizes the variables used in the calculation of the unit vector above.

4.1.3 Control Constraints

The control constraints are straight forward and consist of the limits on the control efforts. These were initially set to the actuator limits for each experiment but more conservative limits

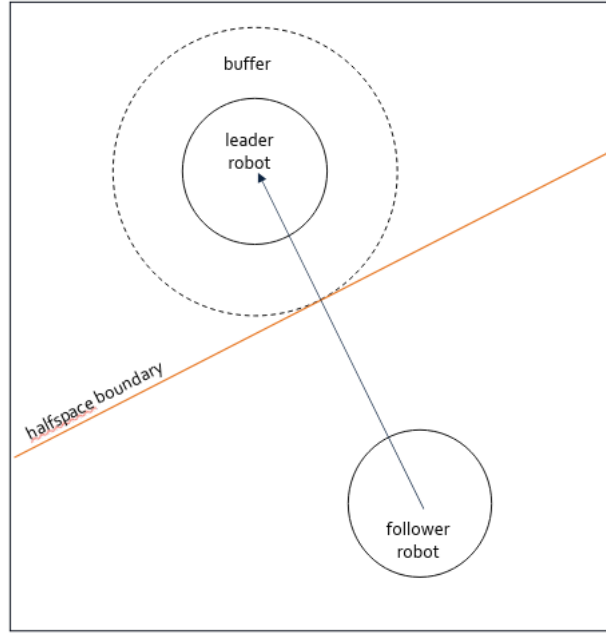


Figure 4.1: Obstacle Avoidance Constraint Depiction

Table 4.1: MPC State Constraint Variables

Variable	Description
k	Current Time Step
$\mathbf{lb}_{x,k}$	State Lower Bound
$\mathbf{ub}_{x,k}$	State Upper Bound
$\hat{n}_{k,x}$	X Component of the Unit Vector
$\hat{n}_{k,y}$	Y Component of the Unit Vector
$\bar{x}_{rel,k}$	X Component of the Line of Sight Vector
$\bar{y}_{rel,k}$	Y Component of the Line of Sight Vector
$\bar{y}_{rel,k}$	Y Component of the Line of Sight Vector
$\bar{d}_k$	Magnitude of the Line of Sight Vector

were placed on some iterations to prevent overly aggressive control. The selected values are provided in Table A.8 found in Appendix A. The control constraint inequality and matrix ($\mathbf{D}_k$) are shown in Equations (4.10-4.11).

$$\mathbf{lb}_{u,k} \leq \mathbf{D}_k \mathbf{u}_k \leq \mathbf{ub}_{u,k} \quad (4.10)$$

$$\mathbf{D}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.11)$$

4.1.4 Initial and Final State Conditions

The starting state vector was defined based off of the estimate from the PDA filter. The initial positions are defined directly from the PDA filter estimates as shown in Equation(4.12).

$$x_{rel} = x, \quad y_{rel} = y \quad (4.12)$$

The initial relative velocity and heading values were calculated based on the velocity estimates from the filter and the follower vehicle's own velocity. Equation 4.13 shows this calculation and Table 4.2 describes the variables used.

$$v_{rel} = \sqrt{(v_x + v_{follow})^2 + v_y^2}, \quad \psi = \text{atan2}(-v_y, v_x + v_{follow}) \quad (4.13)$$

Table 4.2: MPC Initial Condition Variables

Variable	Description
x	PDAF X Position Estimate
y	PDAF Y Position Estimate
v_x	PDAF X Velocity Estimate
v_y	PDAF Y Velocity Estimate
v_{follow}	Follower's Velocity

The final condition was defined as being facing forward directly next to the lead vehicle at a safe distance depending on the specific platform for the given experiment. After this point was reached the path planning was turned off and the follower vehicle was commanded to continue straight ahead at a speed faster than the follower. A second lane change ahead of the leader was not done, because at this point the leader would have left the RADAR's field of view and an estimate of the leader's position will likely have drifted from truth.

Chapter 5

TurtleBot4 Experimentation

While the intended application of this work is autonomous Indy racing, a more accessible testbed was needed for initial experimentation. Testing with the racecar itself is difficult because track time is expensive and limited. Preliminary testing is also needed to avoid excessive damage to the car. The chosen stand-in was a standard TurtleBot 4, shown in Figure 5.1. A leader-follower demonstration was developed to mirror a scaled-down version of the Indy Autonomous Challenge. The TurtleBot comes with a camera and LiDAR on board. A Continental RADAR was added to the follower TurtleBot.



Figure 5.1: Standard TurtleBot4

5.1 Methodology

5.1.1 Gazebo Simulation

An open source TurtleBot4 simulator is already available in gazebo. The demonstration was developed with this simulator before being implemented on hardware. The LiDAR and

camera are already simulated in gazebo as they are present on the physical TurtleBot. However, the RADAR measurement simulation needed to be developed. Simulation FMCW RADAR is a non-trivial task. While recent work has been done to accurately model the noise characteristics of RADAR [38–40], there is not an open source measurement level simulation that can be easily implement with ROS 2 and gazebo. Development of a sophisticated RADAR simulator is beyond the scope of this thesis. Therefore, the implemented simulation method is similar to the one used by Wang et al. [41], which augments the simulated LiDAR measurements. Zero-mean Gaussian noise was added to the range, range-rate, and azimuth measurements with standard deviations based on values from the Continental ARS430 RADAR specifications. These values can be found in the Appendix B. Some of the LiDAR measurements were also randomly discarded to account for RADAR measurements being more sparse. Figure 5.2 shows the simulation and sensor visualization. The red and blue markers represent LiDAR and RADAR detections respectively. The larger cyan and pink squares are the PDA filter solutions of LiDAR and RADAR respectively.

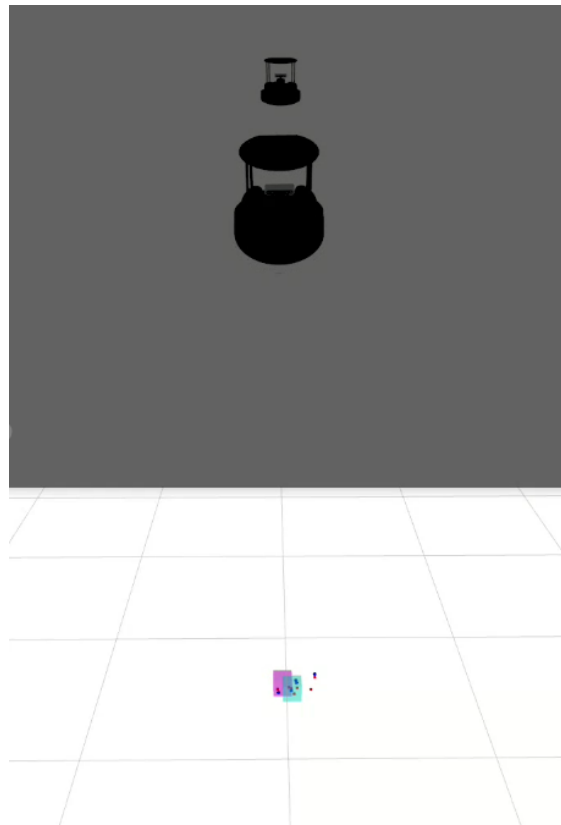


Figure 5.2: Demonstration in Gazebo Simulation

The lead robot was commanded a constant forward velocity of 0.15 m/s. A TurtleBot4's maximum speed is approximately 0.3 m/s. The slower speed was chosen for the leader to make an overtaking maneuver possible. A second TurtleBot started at approximately 1.2 meters behind the leader. The follower robot's sensor measurements were used to track the leader robot and the solution of the filter was fed to the MPC to plan to execute a pass.

The simulation was ultimately used as an initial confirmation that the filter and control would successfully run real-time, before testing on the real robots. Since the RADAR simulation was derived directly from the LiDAR simulation, comparison of the performance would likely not be reflective of the real sensors. Therefore all results presented are from the real hardware.

5.1.2 Real Robots

The same demonstration that was developed in simulation was also done with physical TurtleBot4s. Since they do not come with a RADAR, a Continental ARS430 RADAR was mounted on the front of the follower robot as shown in Figure 5.3. This RADAR is an FMCW automotive RADAR similar to the one that is used on the IAC racecars. The LiDAR measurements were filtered using the PDAF to use as a reference for the RADAR position solution. The TurtleBot4 publishes an odometry solution for velocity that is solved for using the measurements from the IMU and wheel encoders onboard the robot. This velocity solution was used as the reference for both the LiDAR and RADAR-based velocity estimates. A third filter that fused the two sensors was also tested, using LiDAR measurements of position and RADAR measurements of velocity. In addition to the full overtaking maneuver, tracking was tested with a stationary follower and the leader driving straight forward at a constant speed, and with the follower continuing behind the leader at a constant speed. The tuned values of Q and R for both the filter and controller are in Appendix A. All position and velocity results are presented in the follower robot's body frame.



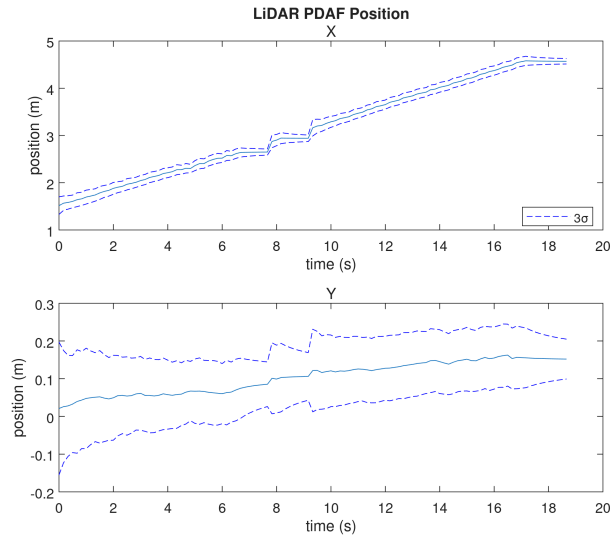
Figure 5.3: TurtleBot4 with Continental RADAR

5.2 Results

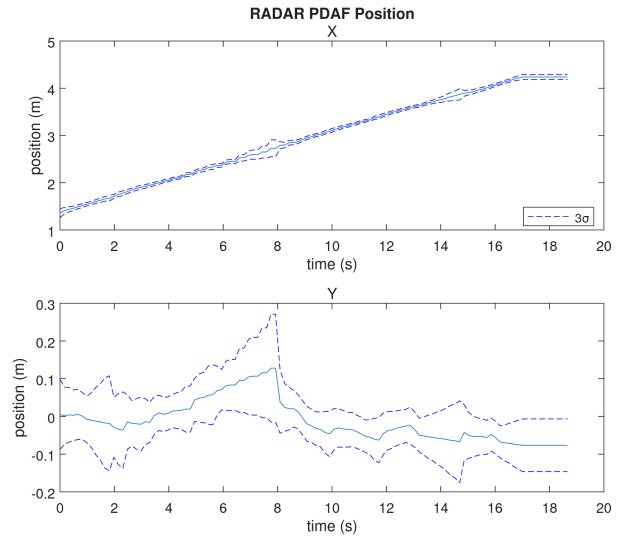
5.2.1 Static Sensor

The first experimental test case involved keeping the second robot stationary while the first moved forward at a constant speed. Figure 5.4 shows the x and y position estimates of the moving robot relative to the stationary one along with their 3σ bounds.

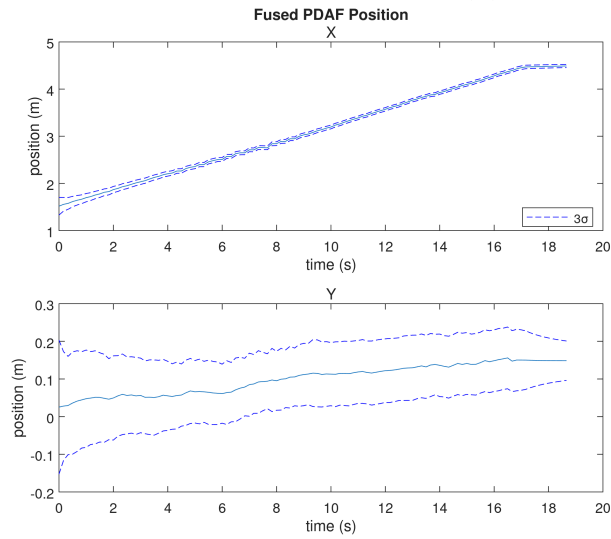
All three filters perform similarly in tracking the x position. The two filters that rely on LiDAR measurements for position tracking produced similar results for y position, while the RADAR filter did not. Since there was no other localization available for the robots, there was not an additional reference to compare the three filter estimates to. However, the LiDAR has a more reliable measurement and the solution is more accurate to the relative motion that was visually observed during experimentation. The tracking performance in the y axis is important in the case of an overtaking maneuver to ensure the lane change creates enough clearance to avoid collision.



(a) LiDAR Position Estimate



(b) RADAR Position Estimate



(c) Fused Position Estimate

Figure 5.4: Position Estimates of Leader From Static Sensor

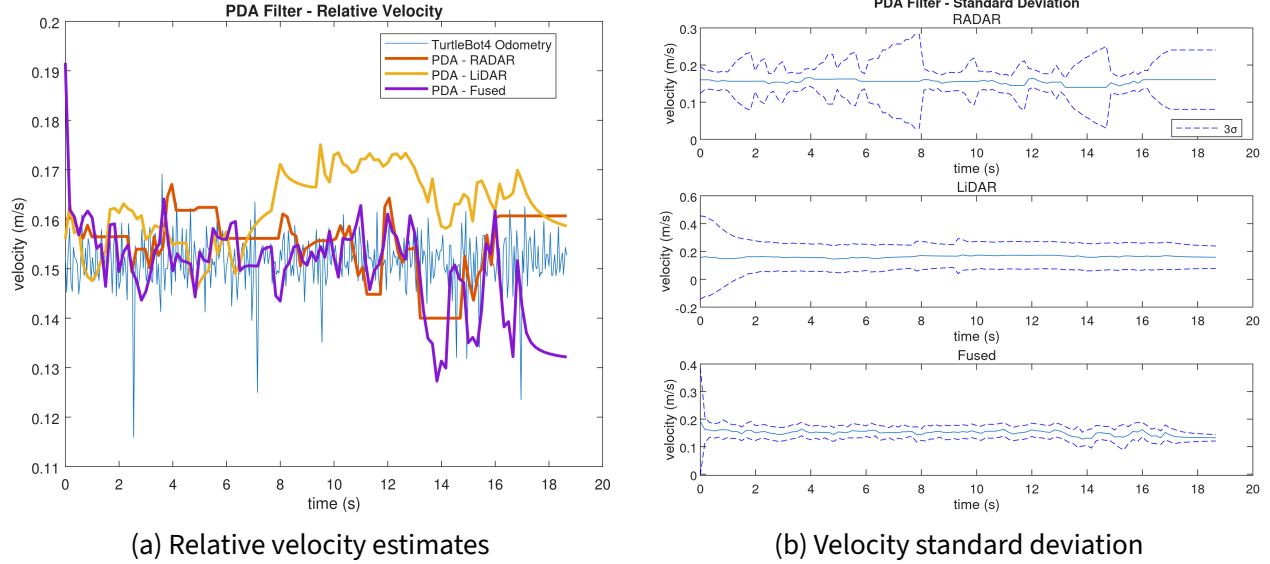
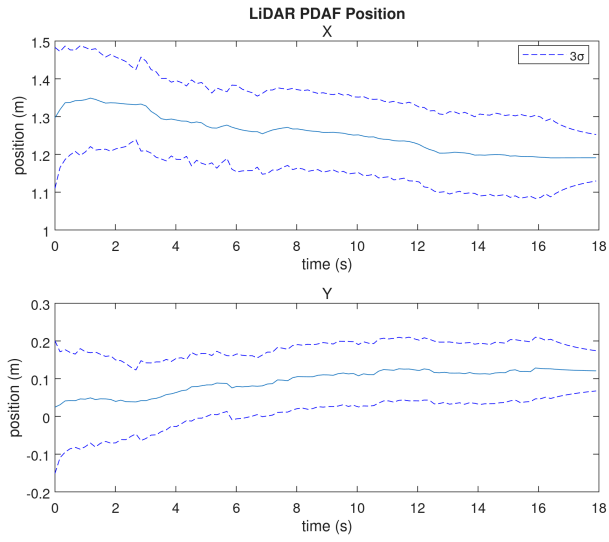


Figure 5.5: Velocity Estimates of Leader From Static Sensor

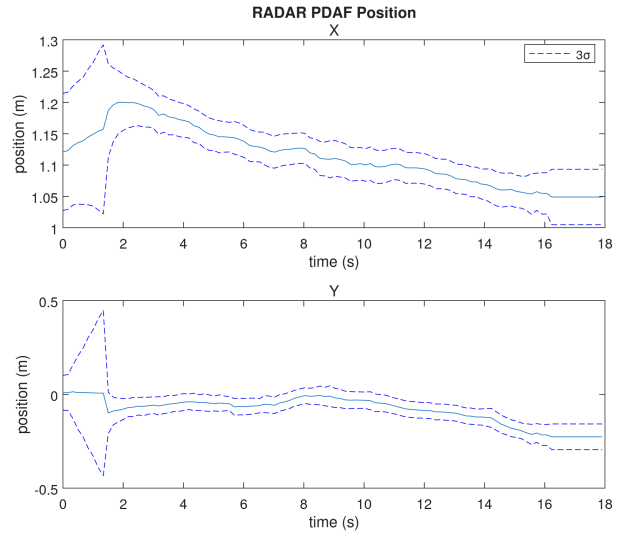
Figure 5.5a shows the velocity in the x direction of all three filters plotted against the difference of the two TurtleBots provided odometry solutions, for reference. The y direction was not considered since all motion was assumed to be along the x axis. Figure 5.5b shows the same velocity solutions plotted with their respective 3 standard deviation bounds. The LiDAR velocity solution has a slight offset, whereas the two filters that rely on RADAR measurements for the velocity estimate do not have the same offset. The fused filter is able to track velocity with greater confidence as it is still informed with LiDAR range measurements when all RADAR measurements are rejected by the PDA filter.

5.2.2 Following

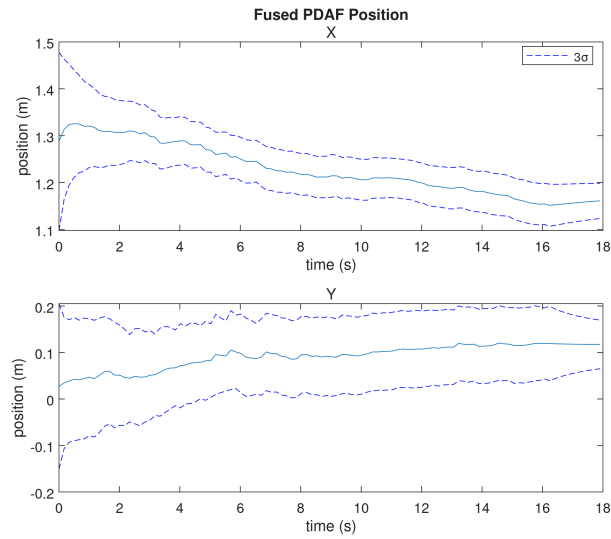
For the second experiment the second robot followed behind the first at relatively the same constant speed. Figure 5.6 is similar to Figure 5.4 and shows the position results of all filters for the following experiment. The performance of position tracking across all three filters is similar in the following case as in the static case. The RADAR-based estimate drifts in the y direction and can be assumed to be overall less accurate in both directions than the LiDAR-based position estimates of the other two filters.



(a) LiDAR Position Estimate



(b) RADAR Position Estimate



(c) Fused Position Estimate

Figure 5.6: Position Estimates of Leader From Follower

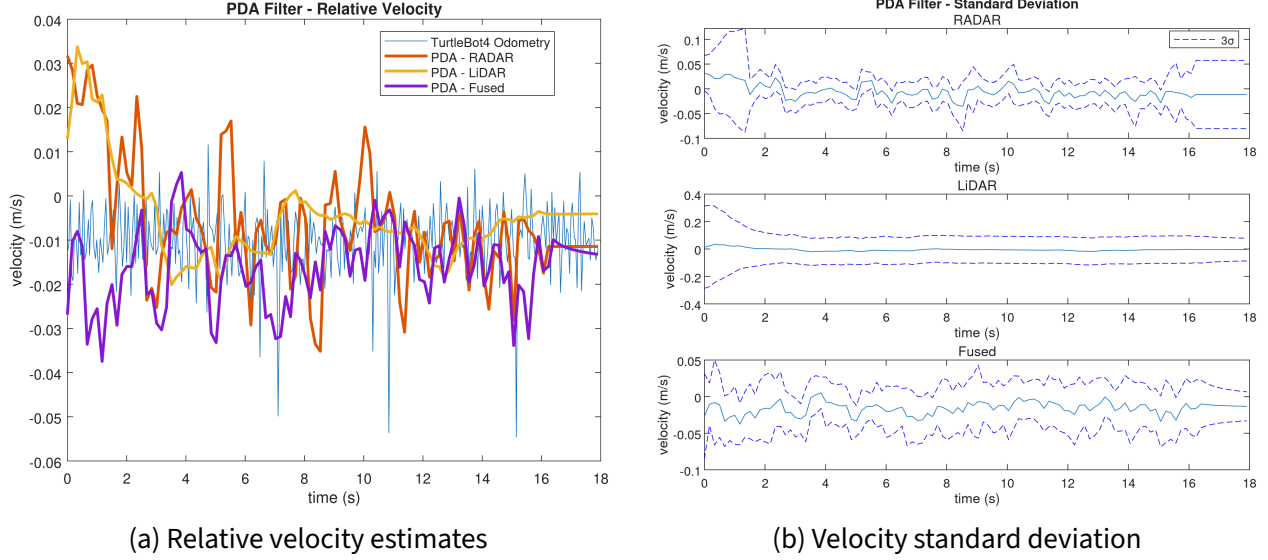


Figure 5.7: Velocity Estimates of Leader From Follower

Figure 5.7 presents the velocity results for the following case the same way Figure 5.5 does for the static case. Similar to the static experiment, the fused filter offers the most confident velocity estimate of the three filters.

5.2.3 Overtaking

Since the RADAR only solution was unable to accurately track the position of the leader robot, the RADAR-based position estimate was not used to perform an overtake. Two cases were tested for the overtaking experiments. The first relied solely on the the LiDAR filters estimates of all states in the MPC execution. The second used the LiDAR position estimates and RADAR velocity estimates to inform the controller. For the rest of the thesis the first case is referred to as LiDAR-based overtaking and the second as RADAR-based overtaking. The fused filter was not used to inform the real-time overtaking maneuver, but the data from each maneuver was post processed with it.

There were 5 trials run of each of the two overtaking cases. Results from one of each are shown below with the rest of the trials being included in Appendix C. Figure 5.8 and Figure 5.10 show position results. Figure 5.9 and Figure 5.11 show velocity results. All overtaking tracking

results are presented the same way they were for the static and following experiments. For all trials the tracking results of all three filters are presented regardless of which was used to inform the overtake.

LiDAR-Based Overtaking

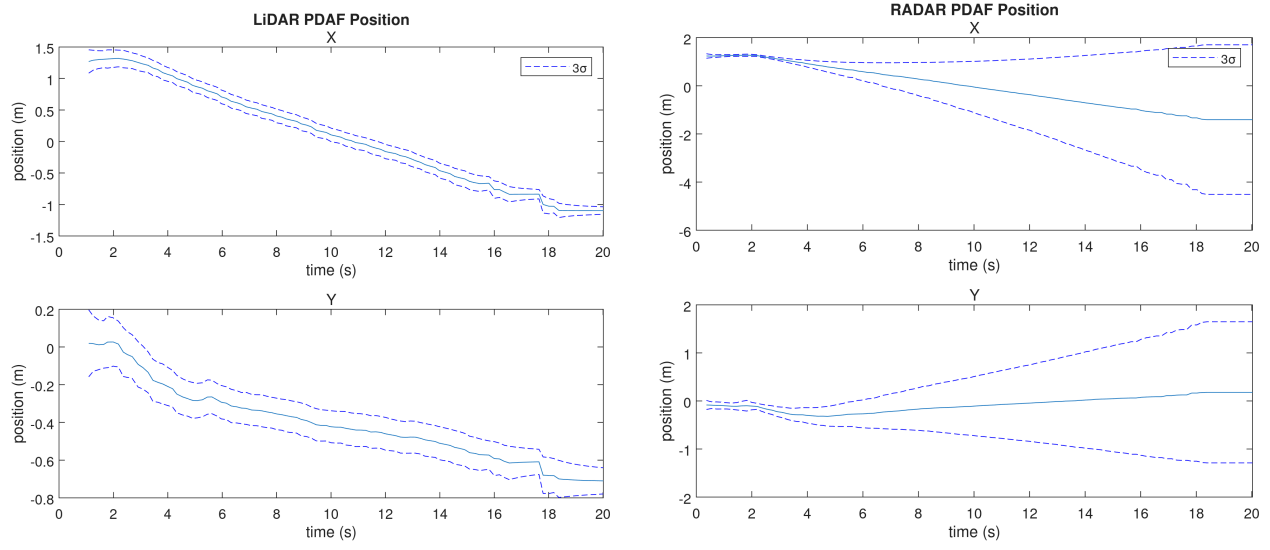
In both overtaking cases the RADAR filter was unable to continue tracking position once the follower turned to begin the lane change. This supports what was seen in the static and following cases, that the RADAR does not track the position of the TurtleBot reliably. The LiDAR and fused filters once again performed similarly. The fused filter tracked with slightly more confidence, which is the result of having a better estimate of velocity and therefore a better prediction of the change in position.

The overtaking maneuver creates an abrupt change in relative velocity when the follower speeds up to pass the leader. All three filters track the change, but with slightly different behavior. The two RADAR-based velocity estimates immediately adjust, whereas the LiDAR filter takes a couple of seconds to settle around the new value. While both the fused and RADAR filters quickly and accurately track velocity, the fused filter does so with smaller covariance.

RADAR-Based Overtaking

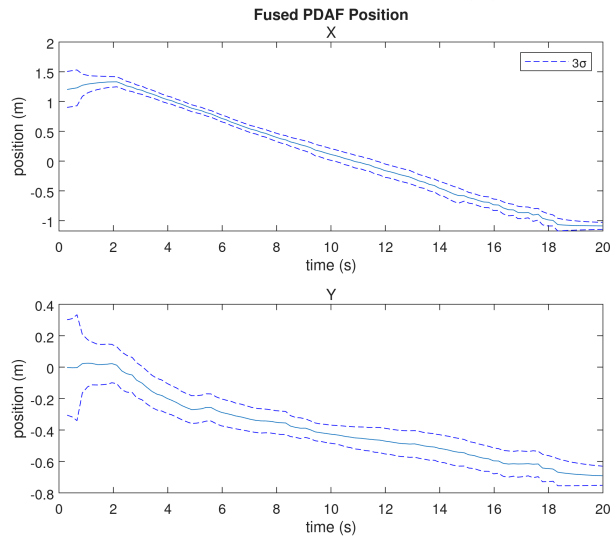
The results from the RADAR-based overtaking maneuvers were all very similar to those of the LiDAR-based overtaking. The time to overtake across all trials is shown in Table 5.1. The average time to overtake for the LiDAR trials was 9.04 seconds with a standard deviation of 0.48 seconds. The average time to overtake for the RADAR trials was 9.61 seconds with a standard deviation of 1.42 seconds. The difference in the average time to overtake between the two sensors is negligible. The main notable difference is that there is a greater variance when using the RADAR estimate to inform the MPC. This means that on an individual run using the RADAR is more likely to lead to a faster or slower time to overtake than the LiDAR.

The remaining trial results shown in Appendix C also show the sensor responses to other unexpected behaviors, such as the leader suddenly stopping, or the follower speeding up and



(a) LiDAR Position Estimate

(b) RADAR Position Estimate

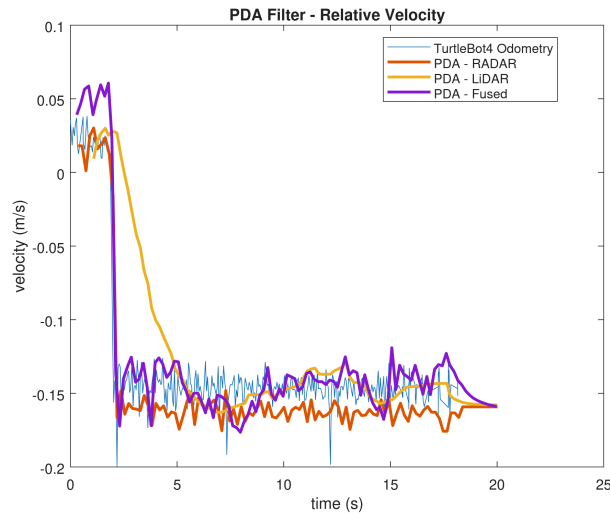


(c) Fused Position Estimate

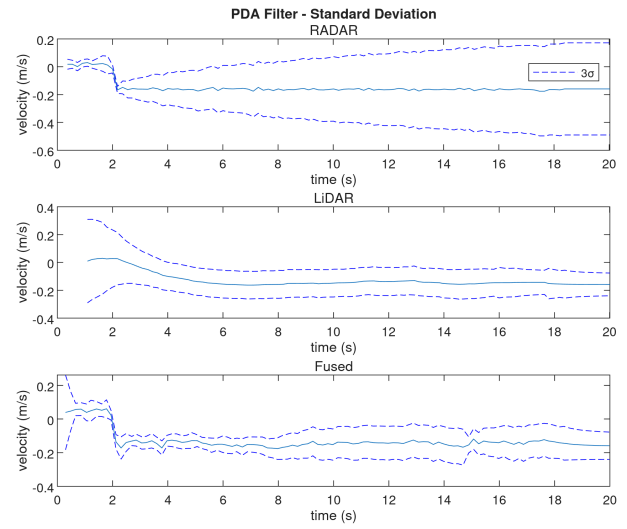
Figure 5.8: Position Estimates of Leader From Follower

Table 5.1: Time to Overtake in Seconds For All TurtleBot Overtaking Trials

Trial:	1	2	3	4	5
RADAR	11.22	7.72	10.46	10.04	8.63
LiDAR	8.79	9.52	9.46	9.05	8.36



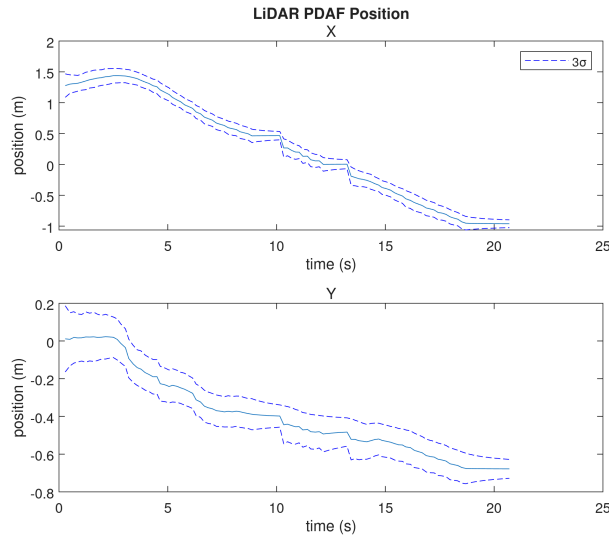
(a) Relative velocity estimates



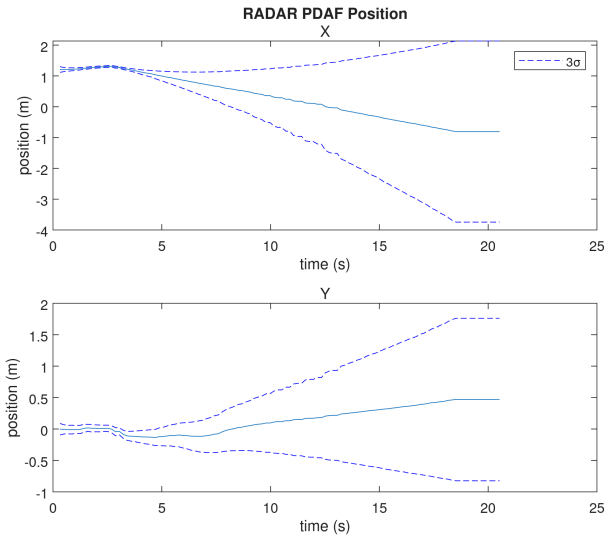
(b) Velocity standard deviation

Figure 5.9: Velocity Estimates of Leader From Follower

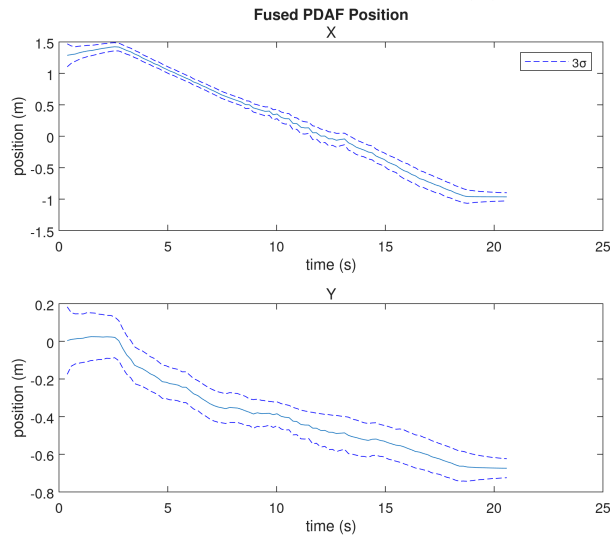
slowing down throughout the course of an overtake. Overall, the results consistently show the LiDAR tracks position better, the RADAR tracks velocity better, and sensor fusion allows for better tracking of all states.



(a) LiDAR Position Estimate

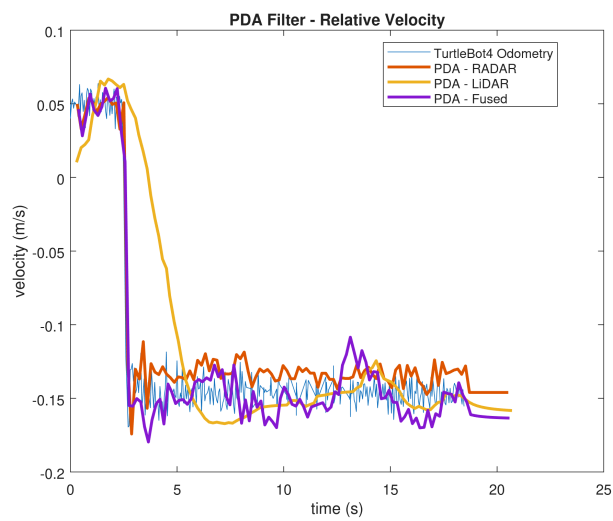


(b) RADAR Position Estimate

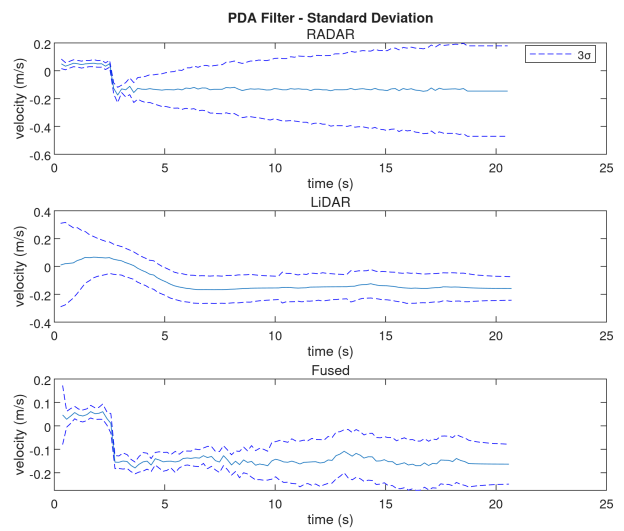


(c) Fused Position Estimate

Figure 5.10: Position Estimates of Leader From Follower



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure 5.11: Velocity Estimates of Leader From Follower

Chapter 6

Vehicle Experimentation

After experimentation with TurtleBots, the developed filter was then tested with two cars. The leader vehicle was a Kia Optima and the follower a Lincoln MKZ, both are shown in Figure 6.1. The same Continental RADAR was mounted on the follower vehicle. As opposed to the 2D LiDARs that come with the TurtleBot4s, the Lincoln was equipped with a Luminar Hydra H3 ToF LiDAR.

6.1 Methodology

Similar experiments as the ones run on the robots were done with the full-scale vehicles. A stationary follower with the leader driving at a constant speed, a follower also driving at a constant speed, and full overtake were all tested. Both cars have GNSS/INS NovAtel receivers that were used to collect truth reference data. The position and velocity solutions were differenced and rotated to convert the reference measurements from the ECEF frame to the Lincoln MKZ sensor frame. Unlike with the TurtleBot4s, autonomy was not tested with the full-scale vehicles, instead, all maneuvers were done by human drivers.



Figure 6.1: Autonomous Lincoln MKZ

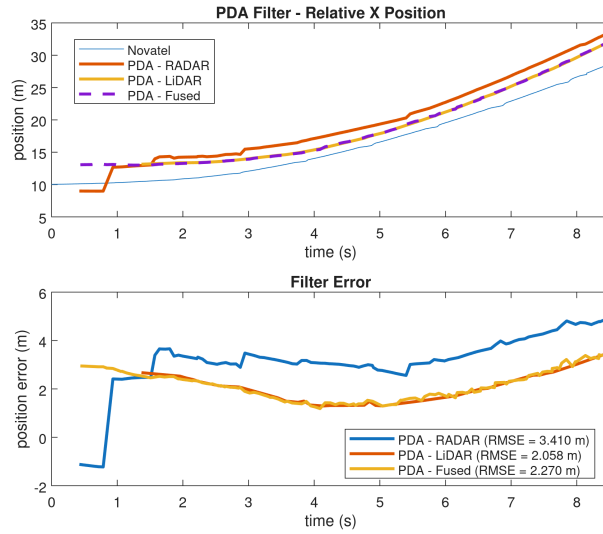


Figure 6.2: X Position Estimate Errors From Static Sensor

6.2 Results

In all of the results presented in this section, the filter estimations start some amount of time after the zero timestamp. This is because each filter does not begin reporting an estimate until it receives a measurement that passes the initial measurement validation step of the PDA filter. Note that the x position estimates have some constant offset due to perception sensors being mounted near the front windshield of the Lincoln and the measurements coming from the back of the Kia, whereas the NovAtel localization solutions are located near the center of the vehicles. A constant was subtracted from the NovAtel reference to account for this. It is not possible to determine exactly where on the lead vehicle the perception measurements originated from, which leads to some discrepancy still being present in the results.

6.2.1 Static Sensor

The x position estimates and errors from all three filters are shown in Figure 6.2. Figure 6.3 shows both the x and y position solutions, while Figure 6.2 focuses only on the x axis as most if not all of the vehicles motions is assumed to be along this axis. A near constant offset is present across all three filters, but the LiDAR and fused filter match closely as both of their position estimates are based off of the same measurements. All estimates closely follow the same trend

as the NovAtel reference; however, the LiDAR-based estimates result in a smaller RMSE, which is to be expected from the LiDAR's more accurate range measurement.

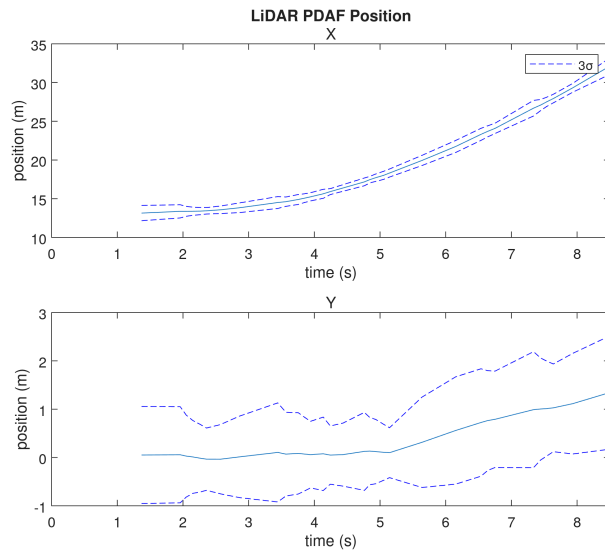
Figure 6.3 shows the position estimates along both axes with their 3σ boundaries. The LiDAR measurement results in a more consistent confidence interval than the RADAR. When the RADAR produces a good detection the standard deviation drops below that of the LiDAR, but the RADAR produces less scans with a validated detection than the LiDAR. All three y position estimates are consistent with the leader not moving significantly in the lateral direction.

All three filters perform similarly in estimating longitudinal velocity. The fused filter has the smallest RMSE in this case, which suggests that pairing the LiDAR position measurement with RADAR's range rate allows for this direct measurement to be better leveraged than the RADAR's own less accurate range measurement.

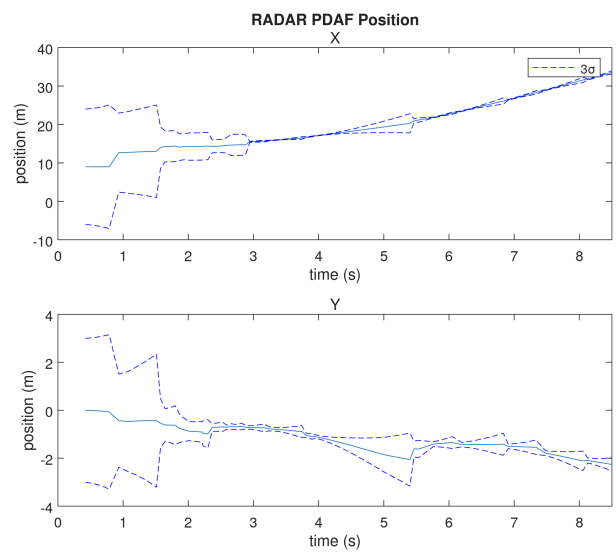
6.2.2 Following

Figure 6.5 and Figure 6.6 show the position tracking results of the car following experiment. The position estimates were fairly consistent across all three filters in the following experiment. The RADAR-based RMSE was only slightly higher than the other two. A constant offset is also present here as expected. The effect of relative inconsistency in valid RADAR measurements compared to the LiDAR can once again be seen in the 3σ bounds of the position estimates in Figure 6.6. The true relative position between the two test vehicles along the y axis was near zero by experimental design, and the estimates from all filters reflect this.

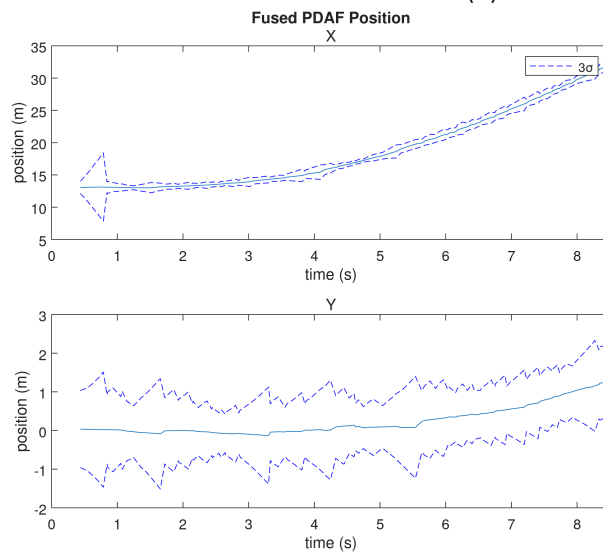
The benefit of the direct range rate measurement is more apparent in the velocity tracking results of this experiment shown in Figure 6.7. While all three estimates lag behind the reference, the LiDAR-based velocity has slightly more lag than the other filters. The fused and RADAR filters have a similar RMSE in the velocity estimate, but the LiDAR only estimation led to a small but still noticeable increase in RMSE. The consistent standard deviation of a LiDAR-based filter is also apparent in the velocity solutions presented here.



(a) LiDAR Position Estimate



(b) RADAR Position Estimate



(c) Fused Position Estimate

Figure 6.3: Position Estimates of Kia From Static Sensor

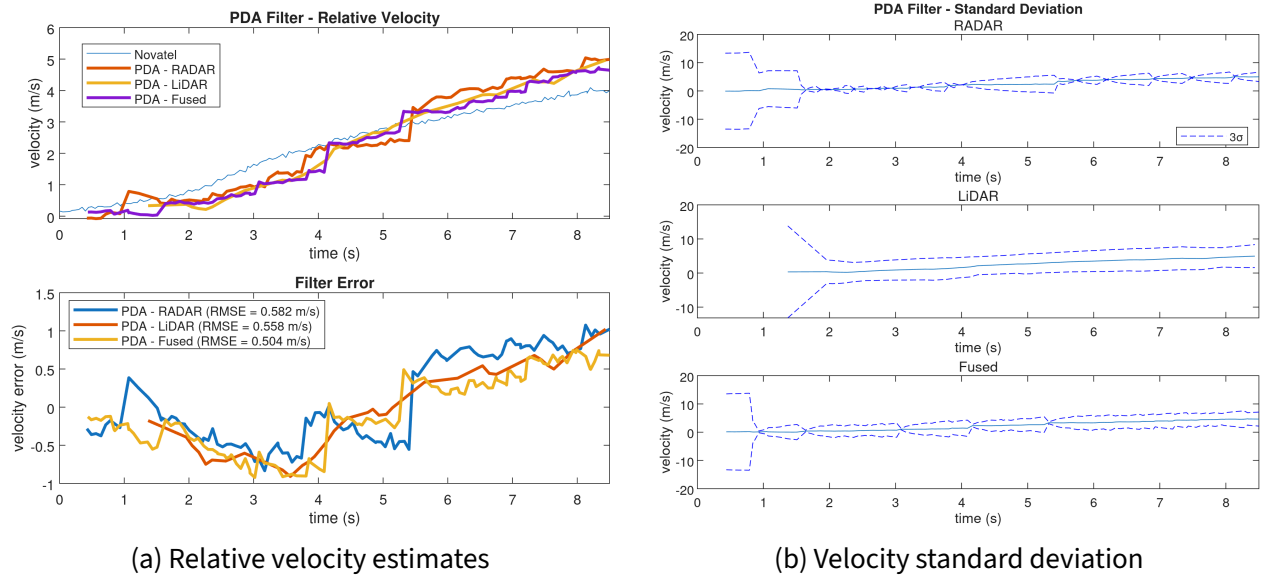


Figure 6.4: Velocity Estimates of Kia From Static Sensor

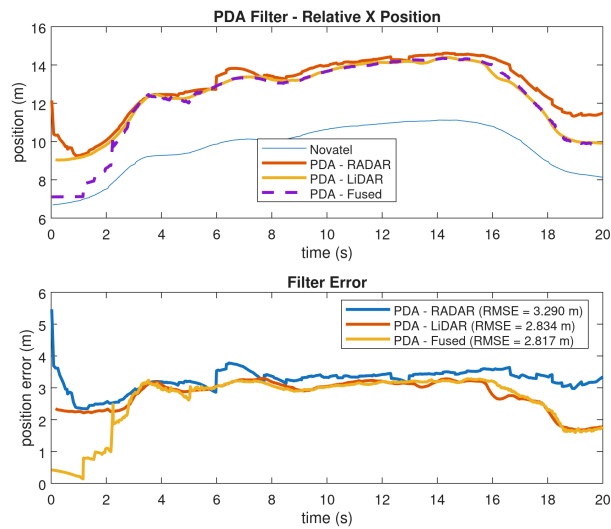
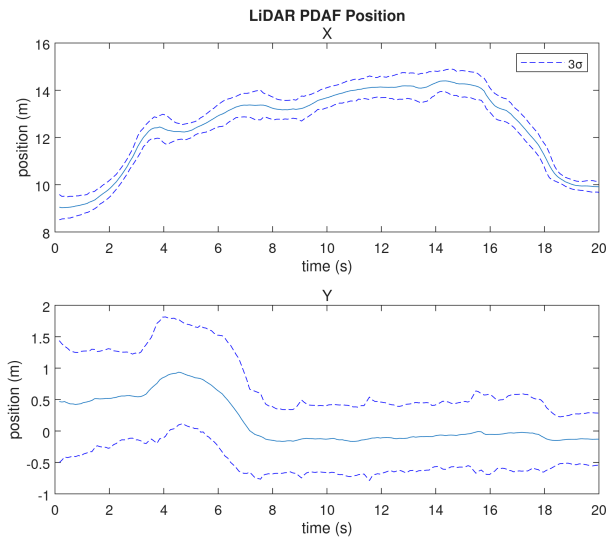
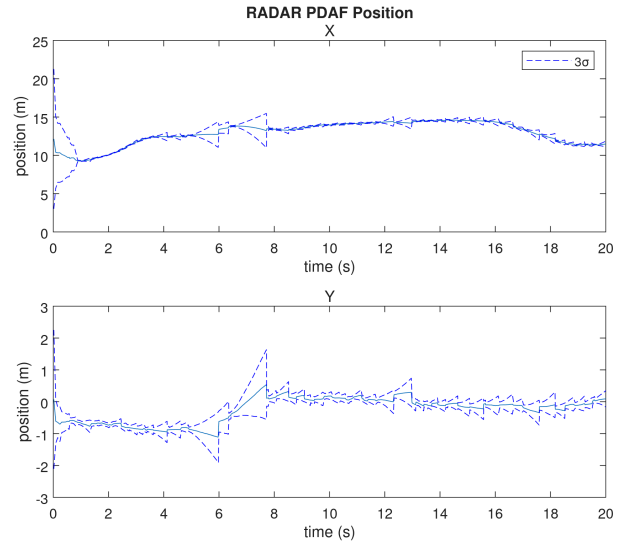


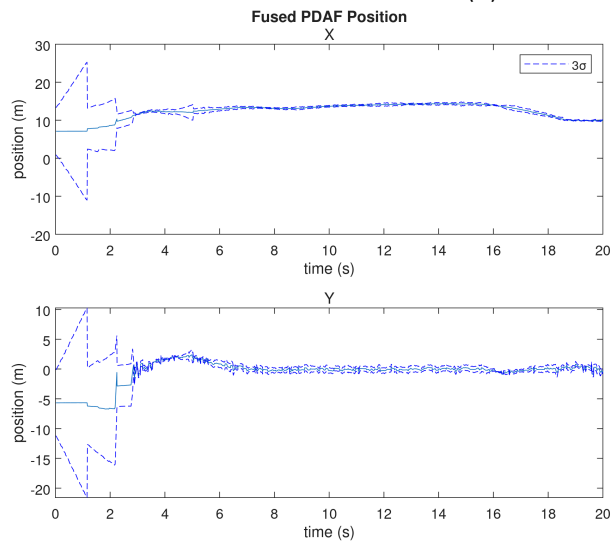
Figure 6.5: X Position Estimate Errors From Follower Vehicle



(a) LiDAR Position Estimate



(b) RADAR Position Estimate



(c) Fused Position Estimate

Figure 6.6: Position Estimates of Kia From Following Lincoln

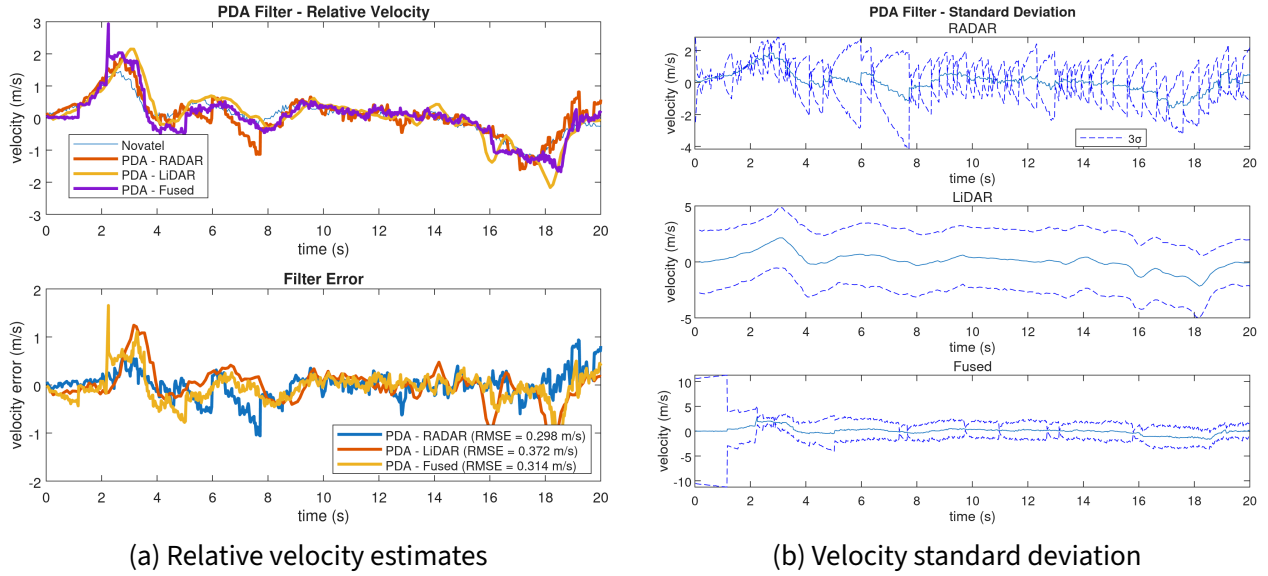


Figure 6.7: Velocity Estimates of Kia From Following Lincoln

6.2.3 Overtaking

Unlike with the TurtleBots, the overtaking experiment with the test vehicles was not autonomous. This section covers the tracking results of an overtake done by a human driver. Similar to the experimentation with the TurtleBots, it can be seen in the position tracking results in Figure 6.8 and Figure 6.9 that the RADAR tracking struggles when the ego vehicle rotates as it begins to pass. The LiDAR-based filters also seem to diverge from the trend of the reference when lateral motion occurs around 9 seconds. Before 9 seconds the same constant offset is present in the LiDAR-based position estimates, but the RADAR filter does not even follow the trend of the NovAtel reference as it did with the previous two vehicle experiments.

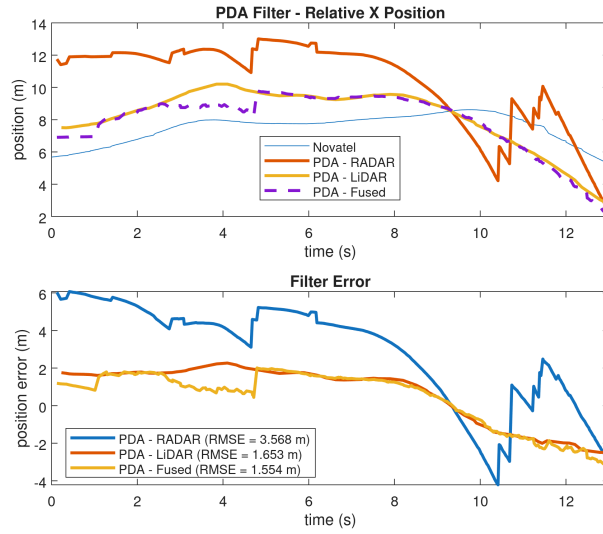
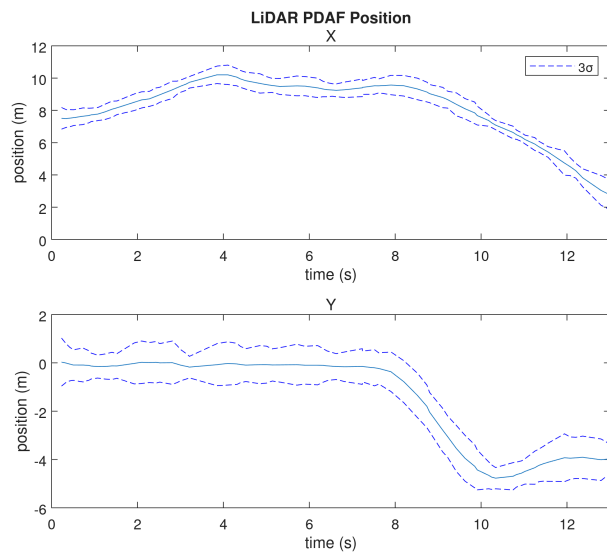


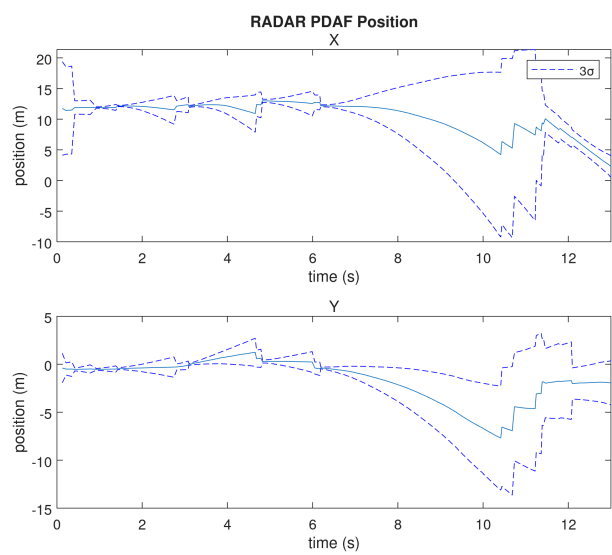
Figure 6.8: X Position Estimate Errors from Overtaking Follower

It is evident in Figure 6.10a that the LiDAR alone tracks lateral motion of the lane change required for an overtake far better than either of the other filters, including the fused filter. The position estimates from the LiDAR only filter were expected to be better than those of the RADAR only filter due to the difference in sensor accuracy. However, this suggests that using a RADAR measurement to aid in estimating a velocity state hinders the corresponding position estimate. This is likely due to the low speeds along the y axis, as the fused filter performed similarly to the LiDAR only filter in tracking position in the x axis.

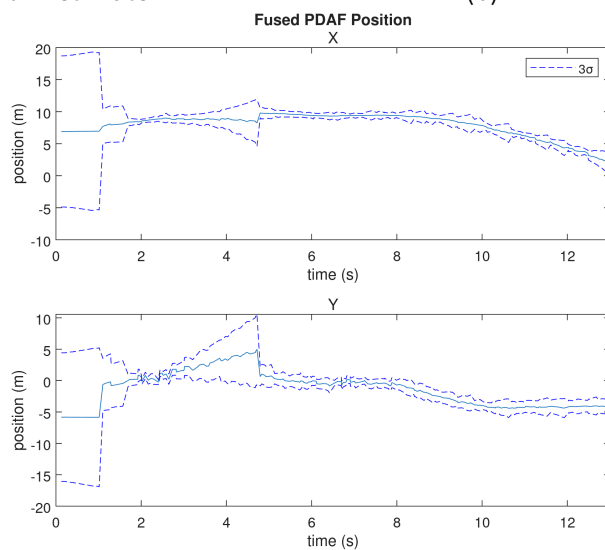
The lateral velocity estimates from the overtaking experiment in Figure 6.10 show that the benefit of the RADAR velocity measurement diminishes when the ego vehicle rotates and moves laterally. This is partially due to the constant relative velocity model no longer being accurate in the case of this maneuver. The RMSE of the LiDAR velocity estimate is significantly less than that of the RADAR and is also lower than the fused filter. This suggests that while the addition of RADAR may have a benefit in static and following cases using LiDAR alone is the better option in the case of a more dynamic maneuver.



(a) LiDAR Position Estimate

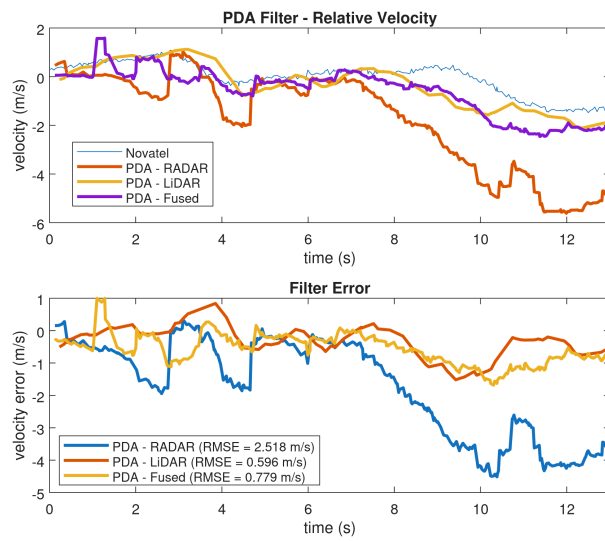


(b) RADAR Position Estimate

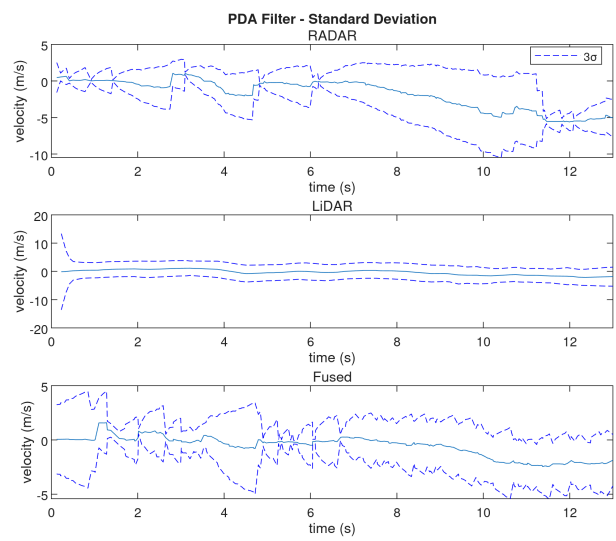


(c) Fused Position Estimate

Figure 6.9: Position Estimates of Kia From Overtaking Lincoln



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure 6.10: Velocity Estimates of Kia From Overtaking Lincoln

Chapter 7

Summary & Conclusions

7.1 Summary

In order to test the benefits of the direct range rate measurement from a RADAR sensor in autonomous driving, a modular autonomy pipeline was developed. This consisted of a PDA filter for lead vehicle tracking using the perception sensors and MPC for path planning and execution of an overtaking maneuver.

Initially the full pipeline was tested on a gazebo simulation of TurtleBot4s to confirm it would run before testing on the hardware. Experimentation was then done with the real robots where tracking with both the LiDAR and RADAR individually as well as fused tracking was tested. Overtaking trials were done with the MPC being informed by LiDAR position estimates. Some iterations used the LiDAR-based estimates of velocity as well, whereas others used the RADAR-based velocity estimates.

The tracking piece of the pipeline was then tested on full-scale vehicles. Both sensors were used to track a lead vehicle individually and a fused filter was also tested. In the case of these vehicles the MPC was not tested.

7.2 Conclusions

In all cases the LiDAR outperformed the RADAR when estimating relative position of the target. This indicates that a RADAR only tracking module would not be ideal. The benefit of a fused filter compared to a LiDAR only filter for the purpose of velocity tracking varies depending on the scenario. With small scale slow moving robots the tracking results that included a RADAR velocity estimate were only marginally better, and do not lead to an improvement in overtaking speed.

When it comes to full-scale vehicles moving at relatively slow speeds the RADAR had a

noticeable improvement in some scenarios. If the ego vehicle is static or following directly behind the leader a measurement of velocity did not improve the position solution, but the velocity estimate did improve. This suggests that prior to attempting an overtake, a RADAR measurement could be used to better detect the speed of an opponent if a constant velocity PDA filter is being used, but any remaining tracking after the overtaking begins should be done with the LiDAR in this case.

7.3 Future Work

All testing done in this thesis was performed at speeds significantly slower than that of an autonomous racecar. Further testing at highway speeds and then race speeds may better indicate the true extent to which using a RADAR to supplement LiDAR is worthwhile. All dynamic modeling used in the estimator used a constant velocity model. Using a different model that accounts for turn rate, acceleration or even a filter using multiple models, such as an IMM PDA filter, may result in better tracking across all sensor scenarios if it more accurately depicted the vehicles motion. The need for sensor fusion could also be eliminated entirely if a vehicle had an FMCW LiDAR, although the accuracy of this sensor would need to be compared to FMCW RADAR.

References

- [1] H. Liu, Z. Cao, X. Yan, S. Feng, and Q. Lu. *Autonomous Vehicles: A Critical Review (2004-2024) and a Vision for the Future*. May 2025.
- [2] *J3016_202104 : Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International*. https://www.sae.org/standards/j3016_202104-taxonomy-definitions-terms-related-driving-automation-systems-road-motor-vehicles.
- [3] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda. "A Survey of Autonomous Driving: Common Practices and Emerging Technologies." In: *IEEE Access* 8 (2020), pp. 58443–58469.
- [4] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam. "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing." In: *IEEE Open Journal of Intelligent Transportation Systems* 3 (2022), pp. 458–488.
- [5] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran. "Autonomous Vehicle Perception: The Technology of Today and Tomorrow." In: *Transportation Research Part C: Emerging Technologies* 89 (Apr. 2018), pp. 384–406.
- [6] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H.-L. Bloecher, K. Werber, and A. Sailer. "Automotive Radar the Key Technology for Autonomous Driving: From Detection and Ranging to Environmental Understanding." In: *2016 IEEE Radar Conference (RadarConf)*. May 2016, pp. 1–6.
- [7] L. Fan, J. Wang, Y. Chang, Y. Li, Y. Wang, and D. Cao. "4D mmWave Radar for Autonomous Driving Perception: A Comprehensive Survey." In: *IEEE Transactions on Intelligent Vehicles* 9.4 (Apr. 2024), pp. 4606–4620.
- [8] S. M. Patole, M. Torlak, D. Wang, and M. Ali. "Automotive Radars: A Review of Signal Processing Techniques." In: *IEEE Signal Processing Magazine* 34.2 (Mar. 2017), pp. 22–35.
- [9] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu. "RODNet: A Real-Time Radar Object Detection Network Cross-Supervised by Camera-Radar Fused Object 3D Localization." In: *IEEE Journal of Selected Topics in Signal Processing* 15.4 (June 2021), pp. 954–967.
- [10] D. C. de Ramos, L. R. Ferreira, M. M. D. Santos, E. L. S. Teixeira, L. R. Yoshioka, J. F. Justo, and A. W. Malik. "Evaluation of Cluster Algorithms for Radar-Based Object Recognition in Autonomous and Assisted Driving." In: *Sensors* 24.22 (Jan. 2024), p. 7219.
- [11] Y. Li and J. Ibanez-Guzman. "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems." In: *IEEE Signal Processing Magazine* 37.4 (July 2020), pp. 50–61.

- [12] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb, and P. Duthon. “Automotive LiDAR Performance Verification in Fog and Rain.” In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 1695–1701.
- [13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. *PointPillars: Fast Encoders for Object Detection from Point Clouds*. May 2019. arXiv: 1812.05784 [cs].
- [14] Z. Wang, Y. Wu, and Q. Niu. “Multi-Sensor Fusion in Automated Driving: A Survey.” In: *IEEE Access* 8 (2020), pp. 2847–2868.
- [15] H. Liu, C. Wu, and H. Wang. “Real Time Object Detection Using LiDAR and Camera Fusion for Autonomous Driving.” In: *Scientific Reports* 13.1 (May 2023), p. 8056.
- [16] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han. *BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation*. Sept. 2024. arXiv: 2205.13542 [cs].
- [17] K. Shi, S. He, Z. Shi, A. Chen, Z. Xiong, J. Chen, and J. Luo. *Radar and Camera Fusion for Object Detection and Tracking: A Comprehensive Survey*. Oct. 2024. arXiv: 2410.19872 [cs].
- [18] L. Wang, X. Zhang, J. Li, B. Xu, R. Fu, H. Chen, L. Yang, D. Jin, and L. Zhao. “Multi-Modal and Multi-Scale Fusion 3D Object Detection of 4D Radar and LiDAR for Autonomous Driving.” In: *IEEE Transactions on Vehicular Technology* 72.5 (May 2023), pp. 5628–5641.
- [19] M. Shah, Z. Huang, A. Laddha, M. Langford, B. Barber, S. Zhang, C. Vallespi-Gonzalez, and R. Urtasun. *LiRaNet: End-to-End Trajectory Prediction Using Spatio-Temporal Radar Fusion*. Nov. 2020. arXiv: 2010.00731 [cs].
- [20] A. Saba, A. Adetunji, A. Johnson, A. Kothari, M. Sivaprakasam, J. Spisak, P. Bharatia, A. Chauhan, B. Duff, N. Gasparro, C. King, R. Larkin, B. Mao, M. Nye, A. Parashar, J. Attias, A. Balciunas, A. Brown, C. Chang, M. Gao, C. Heredia, A. Keats, J. Lavariega, W. Muckelroy, A. Slavescu, N. Stathas, N. Suvarna, C. T. Zhang, S. Scherer, and D. Ramanan. “Fast and Modular Autonomy Software for Autonomous Racing Vehicles.” In: *Field Robotics* 4 (Jan. 2024), pp. 1–45. arXiv: 2408.15425 [cs].
- [21] P. Karle, F. Fent, S. Huch, F. Sauerbeck, and M. Lienkamp. “Multi-Modal Sensor Fusion and Object Tracking for Autonomous Racing.” In: *IEEE Transactions on Intelligent Vehicles* 8.7 (July 2023), pp. 3871–3883.
- [22] M. Cellina, M. Corno, and S. M. Savaresi. *Target Tracking via LiDAR-RADAR Sensor Fusion for Autonomous Racing*. July 2025. arXiv: 2505.20043 [cs].
- [23] M. Cellina, M. Corno, and S. M. Savaresi. *LiDAR-Based Vehicle Detection and Tracking for Autonomous Racing*. Jan. 2025. arXiv: 2501.14502 [cs].

- [24] A. Chawla. *Why Prefer Mahalanobis Distance Over Euclidean Distance?* <https://blog.daily-doseofds.com/p/why-prefer-mahalanobis-distance-over>. Apr. 2023.
- [25] C. Jung, A. Finazzi, H. Seong, D. Lee, S. Lee, B. Kim, G. Gang, S. Han, and D. H. Shim. *An Autonomous System for Head-to-Head Race: Design, Implementation and Analysis; Team KAIST at the Indy Autonomous Challenge*. Mar. 2023. arXiv: 2303.09463 [cs].
- [26] S. Cansiz. *Multivariate Outlier Detection in Python*. May 2023.
- [27] T. Staffas, A. Elshaari, and V. Zwiller. “Frequency Modulated Continuous Wave and Time of Flight LIDAR with Single Photons: A Comparison.” In: *Optics Express* 32.5 (Feb. 2024), p. 7332.
- [28] X. Li, F. Wang, N. Wang, and C. Ma. *Frame Fusion with Vehicle Motion Prediction for 3D Object Detection*. <https://arxiv.org/abs/2306.10699v1>. June 2023.
- [29] P. D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2nd ed. GNSS Technology and Application Series. Boston: Artech house, 2013.
- [30] L. Dal’Col, M. Oliveira, and V. Santos. “Joint Perception and Prediction for Autonomous Driving: A Survey.” In: *IEEE Transactions on Intelligent Transportation Systems* (2025), pp. 1–26.
- [31] X. Rong Li and Y. Bar-Shalom. “Tracking in Clutter with Nearest Neighbor Filters: Analysis and Performance.” In: *IEEE Transactions on Aerospace and Electronic Systems* 32.3 (July 1996), pp. 995–1010.
- [32] L. R. Kenari and M. R. Arvan. “Comparison of Nearest Neighbor and Probabilistic Data Association Methods for Non-Linear Target Tracking Data Association.” In: *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*. Oct. 2014, pp. 047–052.
- [33] Y. Bar-Shalom, F. Daum, and J. Huang. “The Probabilistic Data Association Filter.” In: *IEEE Control Systems Magazine* 29.6 (Dec. 2009), pp. 82–100.
- [34] 7 - Probabilistic Data Association Tutorial — Stone Soup 0.1b9 Documentation. https://stone-soup.readthedocs.io/en/v0.1b9/auto_tutorials/07_PDATutorial.html.
- [35] Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: YBS Publishing, 2011.
- [36] S. S. Lodhi, N. Kumar, and P. K. Pandey. “Autonomous Vehicular Overtaking Maneuver: A Survey and Taxonomy.” In: *Vehicular Communications* 42 (Aug. 2023), p. 100623.
- [37] J. T. Boylan. *JTylerBoylan/Osqp-Mpc*. June 2026.

- [38] A. Mock, M. Magnusson, and J. Hertzberg. “RadaRays: Real-time Simulation of Rotating FMCW Radar for Mobile Robotics via Hardware-accelerated Ray Tracing.” In: *IEEE Robotics and Automation Letters* 10.3 (Mar. 2025), pp. 2470–2477. arXiv: 2310.03505 [cs].
- [39] B. Ubezio, P. Sambath, A. Eryildirim, and H. Zangl. “Implementation and Validation of a Comprehensive, Noise-Aware FMCW Radar Simulation Framework.” In: *IEEE Sensors Journal* 25.9 (May 2025), pp. 15235–15246.
- [40] Z. Han, J. Wang, Z. Xu, S. Yang, L. He, S. Xu, J. Wang, and K. Li. *4D Millimeter-Wave Radar in Autonomous Driving: A Survey*. Apr. 2024. arXiv: 2306.04242 [eess].
- [41] C. Wang, A. Eizad, Mark Kleinhans, E. Widzyk-Capehart, and G. Kennedy. *A Digital Twin Simulation Platform for 4D Millimeter-Wave Radar-Based Navigation in Underground Mining Environment*. May 2025.
- [42] AUMOVIO Engineering Solutions. *ARS 430FM - FlexMount: Long Range Radar Sensor 77GHz*. <https://engineering-solutions.aumovio.com/>.
- [43] Shanghai Slamtec Co., Ltd. *RPLIDAR A1: Low Cost 360 Degree Laser Range Scanner, Introduction and Datasheet*. <https://www.slamtec.com>. 2020.
- [44] Luminar Technologies, Inc. *Hydra: High-Performance LiDAR and Perception Engine for Testing and Development*. <https://www.luminartech.com>. 2020.

Appendix A

Tuned Filter and Controller Values

The tables provided in this appendix include the values of all tuning parameters that were used in the filters and controllers throughout this thesis. Section A.1 contains the parameters for the TurtleBot experiments and Section A.2 has the filter values for the test vehicles.

A.1 TurtleBot4

A.1.1 PDA Filter Tuning

Process Noise

Table A.1: PDA Filter Process Noise Parameters

Filter	q_x	q_y
RADAR	0.00075	0.0001
LiDAR	0.00075	0.0001
Fused	0.00075	0.0001

Measurement Noise

Table A.2: PDA Filter Measurement Noise Parameters

Filter	Measurement	R Value
RADAR	x position	1×10^{-3}
RADAR	y position	1×10^{-3}
RADAR	range-rate	1×10^{-4}
LiDAR	x position	0.005
LiDAR	y position	0.005
Fused	x position (LiDAR)	0.005
Fused	y position (LiDAR)	0.005
Fused	range-rate (RADAR)	1×10^{-4}

Association Parameters

Table A.3: PDA Filter Association Parameters

Filter	γ	P_D	P_G	λ
RADAR	6.25 (3 DOF)	0.9	0.9	0.7
LiDAR	4.61 (2 DOF)	0.9	0.9	0.01
Fused (LiDAR component)	4.61 (2 DOF)	0.9	0.9	0.01
Fused (RADAR component)	2.71 (1 DOF)	0.9	0.9	0.7

A.1.2 MPC Controller Tuning

Table A.4: MPC Horizon and Timestep

Parameter	Value
Prediction horizon	30 steps
Timestep Δt	0.182 s

Table A.5: MPC State Cost Weights

State	Q Weight
x_{rel}	1.0
y_{rel}	2.0
ψ	2.0
v_{rel}	0.0

Table A.6: MPC Control Cost Weights

Control	R Weight
v_{follow}	0.01
ω_{follow}	10.0

Table A.7: MPC State Constraints

State	Lower Bound	Upper Bound
x_{rel}	-2.0 m	2.0 m
y_{rel}	-2.0 m	2.0 m
ψ	$-\pi$	π
v_{rel}	0.45 m/s	∞

Table A.8: MPC Control Constraints

Control	Lower Bound	Upper Bound
v_{follow}	0.0 m/s	0.3 m/s
ω_{follow}	-0.3 rad/s	0.3 rad/s

A.2 Vehicles

A.2.1 PDA Filter Tuning

Process Noise

Table A.9: PDA Filter Process Noise Parameters

Filter	q_x	q_y
RADAR	0.8	0.1
LiDAR	4.0	2.0
Fused	1.5	0.5

Measurement Noise

Table A.10: PDA Filter Measurement Noise Parameters

Filter	Measurement	R Value
RADAR	x position	1×10^{-3}
RADAR	y position	1×10^{-3}
RADAR	range-rate	1×10^{-4}
LiDAR	x position	1×10^{-4}
LiDAR	y position	1×10^{-4}
Fused	x position (LiDAR)	1×10^{-4}
Fused	y position (LiDAR)	1×10^{-4}
Fused	range-rate (RADAR)	1.75×10^{-3}

Association Parameters

Table A.11: PDA Filter Association Parameters

Filter	γ	P_D	P_G	λ
RADAR	6.25 (3 DOF)	0.9	0.9	0.01
LiDAR	4.61 (2 DOF)	0.9	0.9	0.01
Fused (LiDAR component)	4.61 (2 DOF)	0.9	0.9	0.01
Fused (RADAR component)	2.71 (1 DOF)	0.9	0.9	0.01

Appendix B

Sensor Specifications

Sensor specifications from the manufacturers of all perception sensors used for this thesis were used when selecting parameters for the filters. All relevant specifications are provided below.

B.1 RADAR (Continental ARS430)

Table B.1: RADAR Measurement Accuracy (Near Range) [42]

Item	Value
Distance accuracy (near)	± 0.10 m
Azimuth accuracy (near, 0°)	$\pm 0.3^\circ$
Azimuth accuracy (near, $\pm 45^\circ$)	$\pm 1^\circ$
Azimuth accuracy (near, $\pm 60^\circ$)	$\pm 5^\circ$
Velocity accuracy	± 0.1 km/h

B.2 LiDAR (RPLIDAR A1M8)

Table B.2: LiDAR Detection Range and Coverage [43]

Item	Value
Distance range	0.15–12 m
Angular range	0° – 360°

Table B.3: LiDAR Measurement Resolution [43]

Item	Value
Distance resolution (< 1.5 m)	< 0.5 mm
Distance resolution (full range)	< 1% of distance
Angular resolution (at 5.5 Hz)	$\leq 1^\circ$

Table B.4: LiDAR Timing [43]

Item	Value
Scan rate (typical)	5.5 Hz
Scan rate (range)	1–10 Hz
Sample frequency (typical)	≥ 8000 Hz

B.3 LiDAR (Luminar Hydra H3)

Table B.5: LiDAR Detection Range and Coverage [44]

Item	Value
Distance range	2–500 m
Angular range	120° (H), 0–30° (V, configurable)

Table B.6: LiDAR Measurement Resolution [44]

Item	Value
Range precision	1 cm
Horizontal angular resolution (min)	0.07°
Vertical angular resolution (min)	0.03°

Table B.7: LiDAR Timing [44]

Item	Value
Frame rate	1–30 Hz (software configurable)
Scan rate	640 lines/s (software configurable)

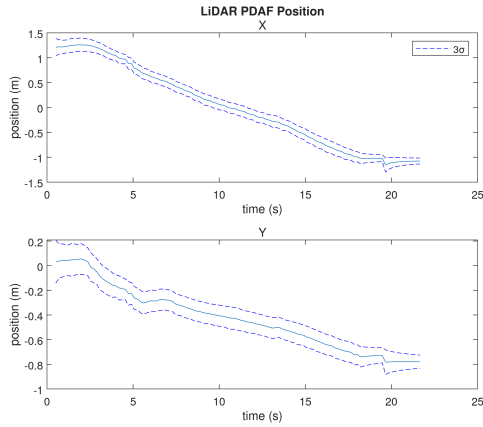
Appendix C

Additional TurtleBot4 Overtaking Trial Results

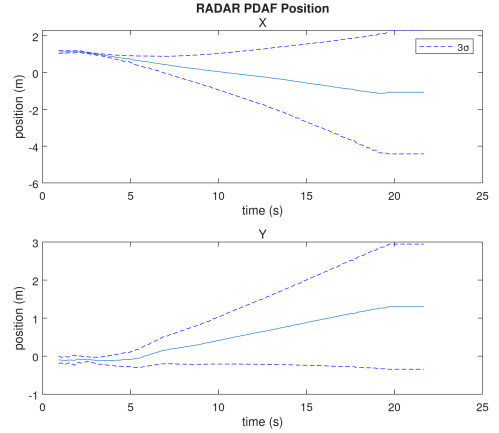
The figures provided in this appendix show the results for all TurtleBot overtaking experimental results that were not presented in Chapter 5. The results from these remaining trials contain similar tracking behavior to the ones previously shown and further support the conclusions outlined in Chapter 7.

C.1 LiDAR

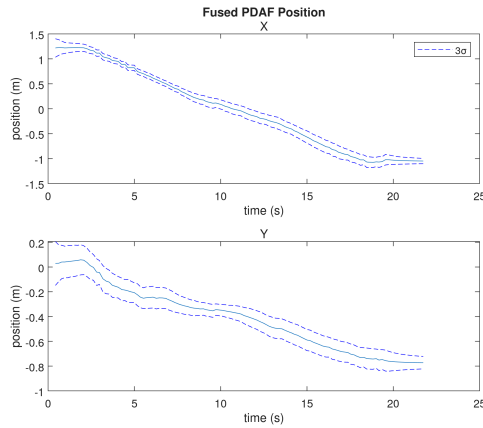
C.1.1 Trial 1



(a) LiDAR Position Estimate

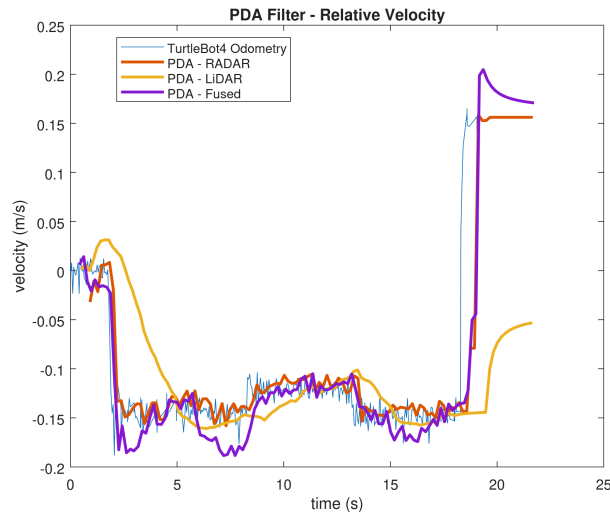


(b) RADAR Position Estimate

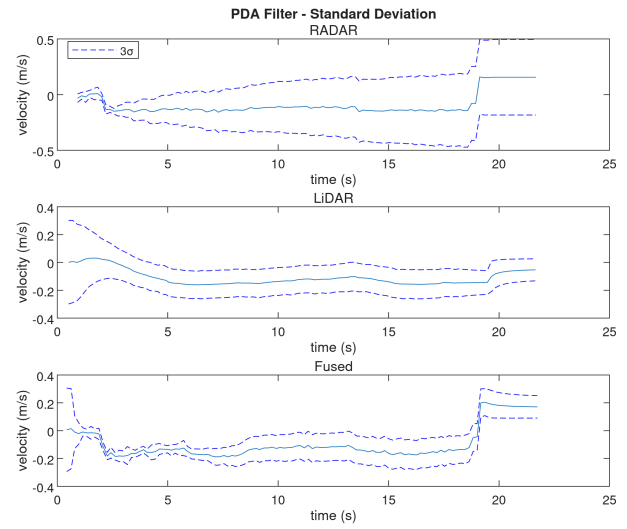


(c) Fused Position Estimate

Figure C.1: Position Estimates of Leader From Follower



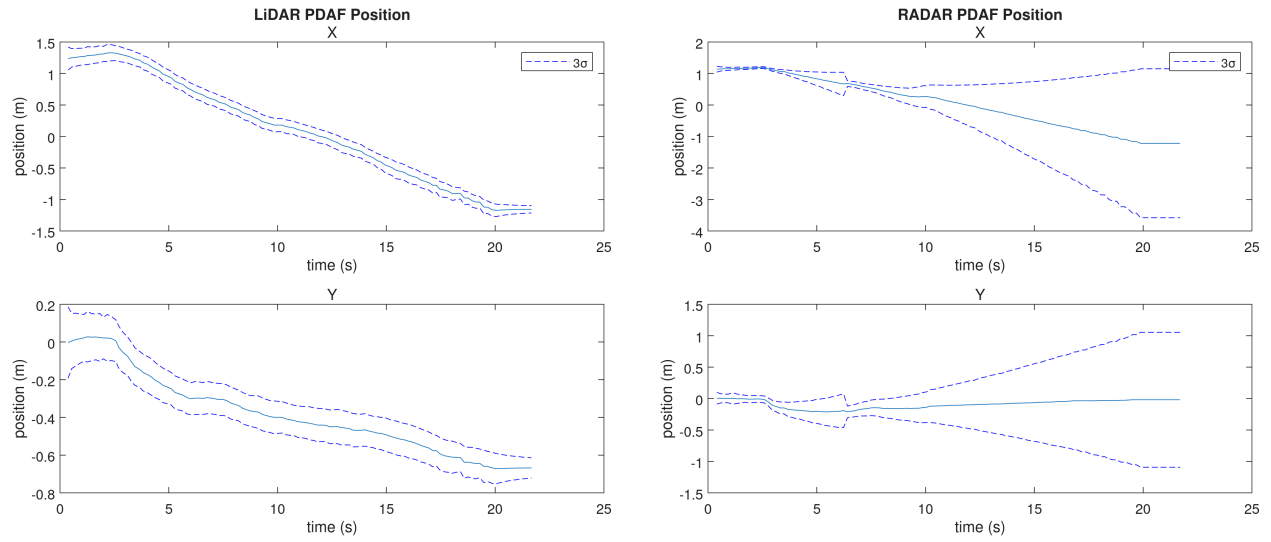
(a) Relative velocity estimates



(b) Velocity standard deviation

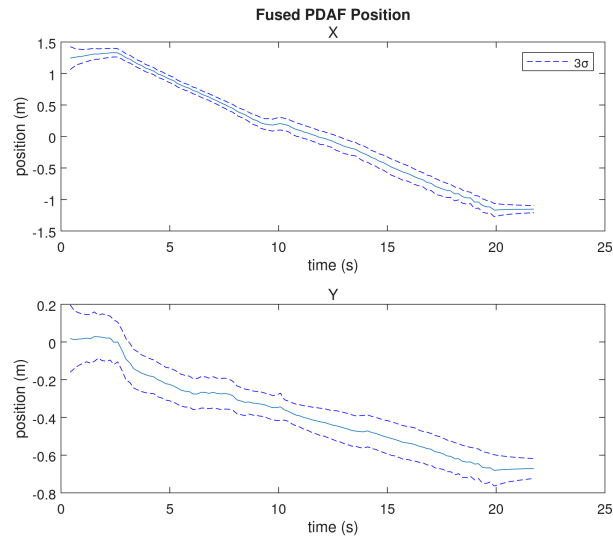
Figure C.2: Velocity Estimates of Leader From Follower

C.1.2 Trial 2



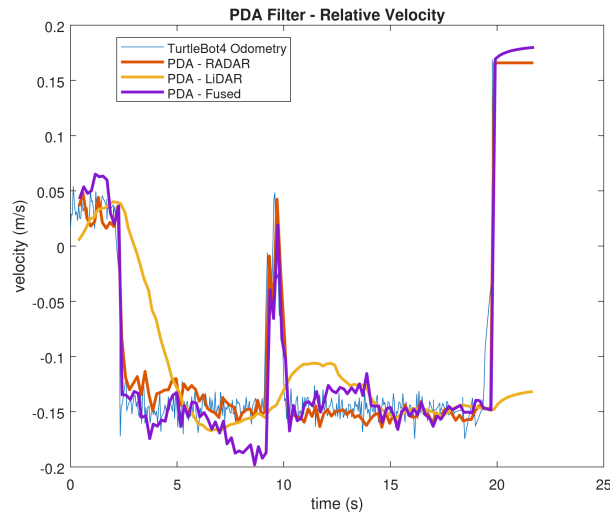
(a) LiDAR Position Estimate

(b) RADAR Position Estimate

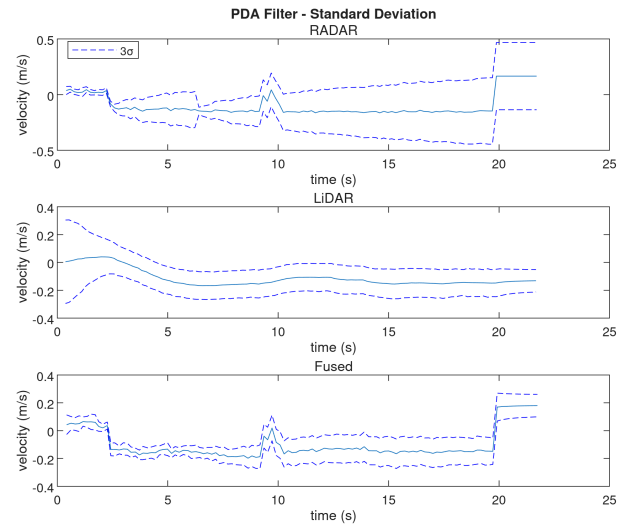


(c) Fused Position Estimate

Figure C.3: Position Estimates of Leader From Follower



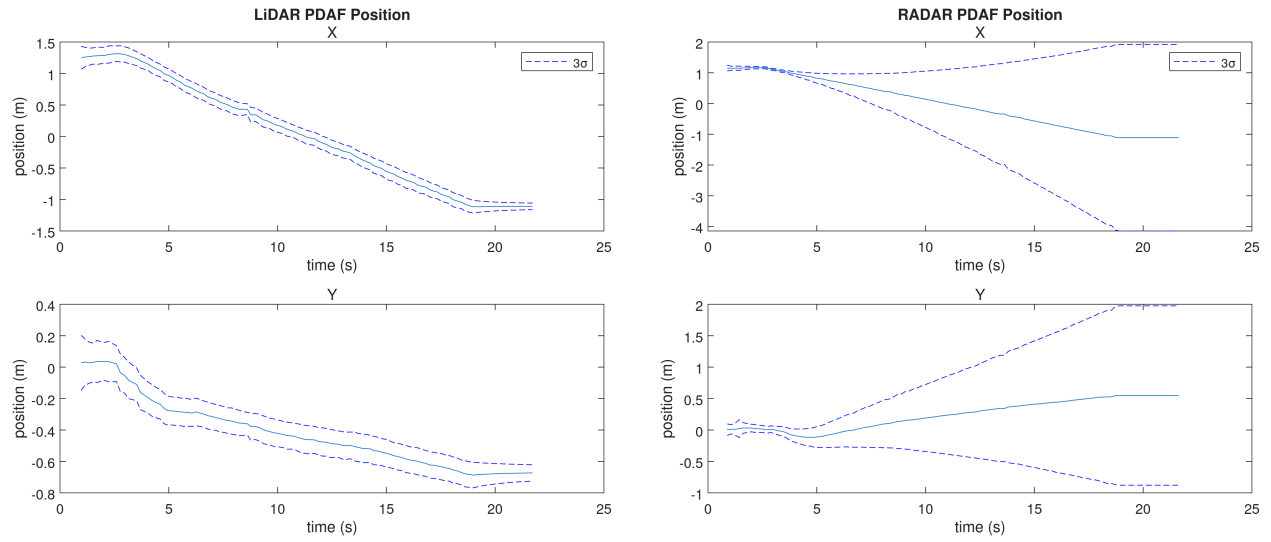
(a) Relative velocity estimates



(b) Velocity standard deviation

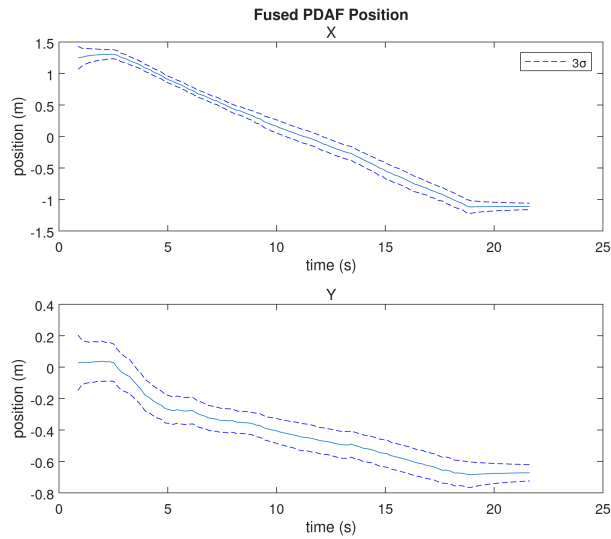
Figure C.4: Velocity Estimates of Leader From Follower

C.1.3 Trial 3



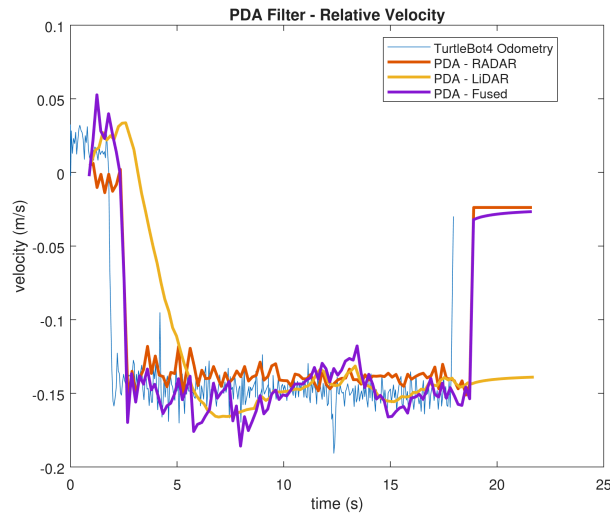
(a) LiDAR Position Estimate

(b) RADAR Position Estimate

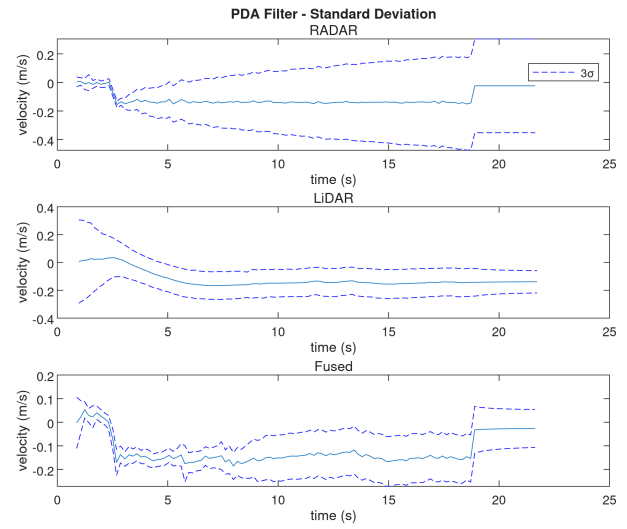


(c) Fused Position Estimate

Figure C.5: Position Estimates of Leader From Follower



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure C.6: Velocity Estimates of Leader From Follower

C.1.4 Trial 4

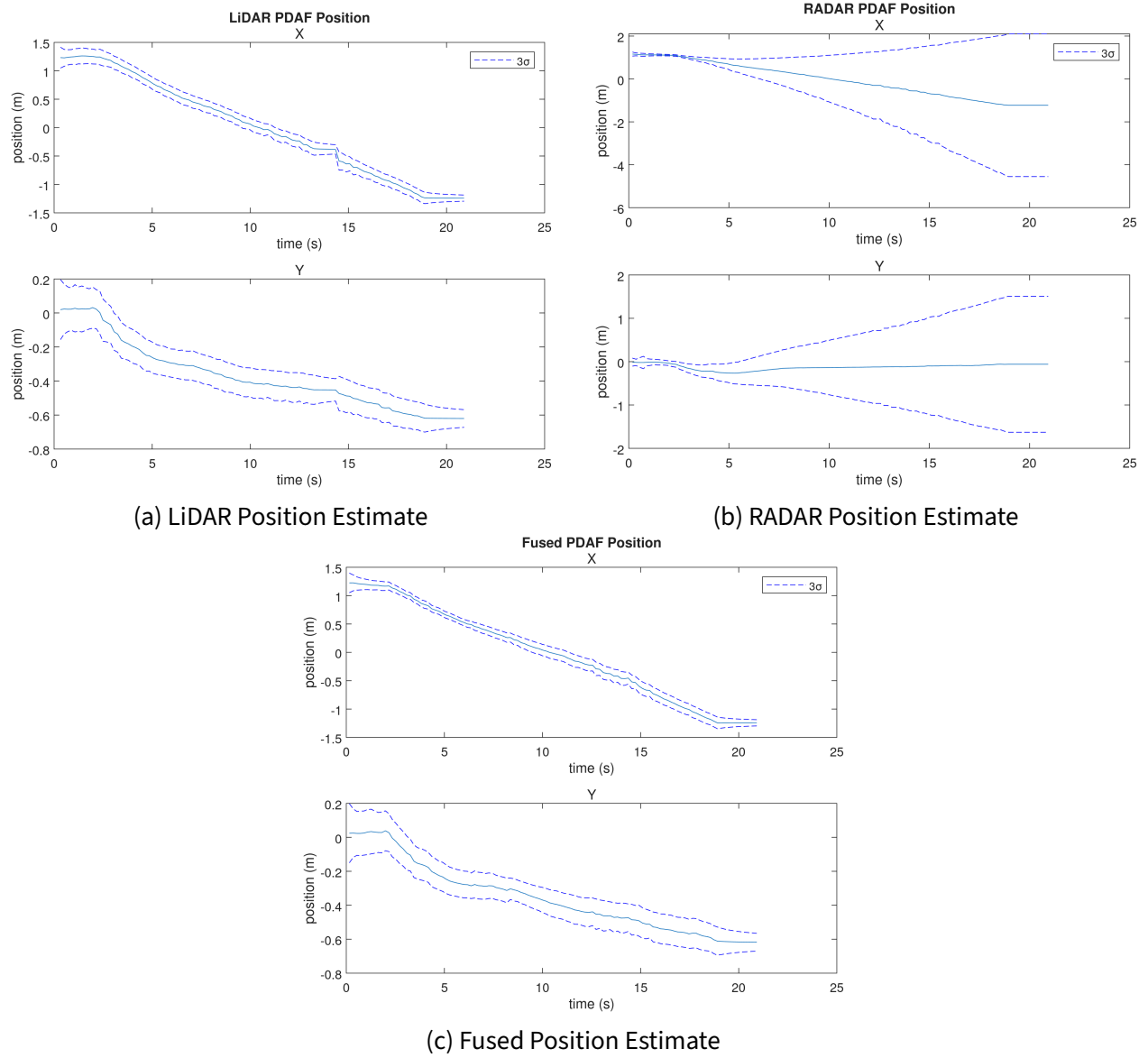
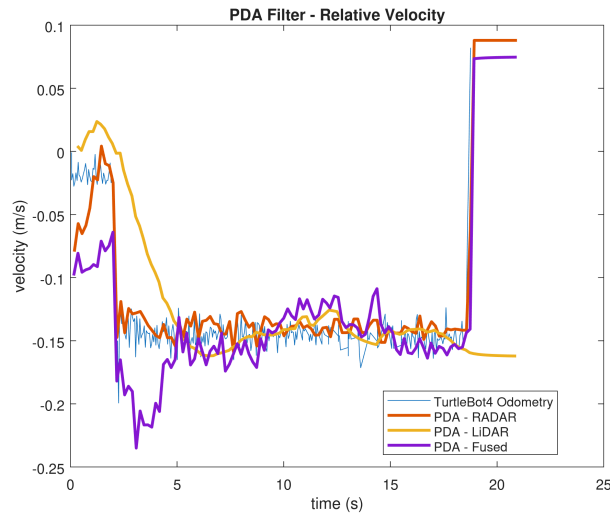
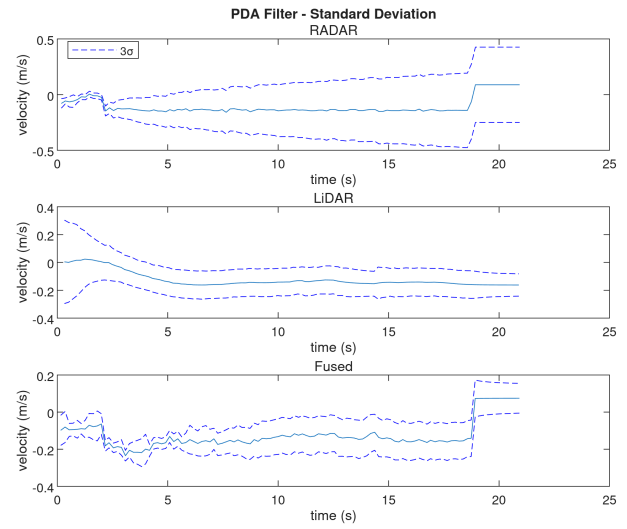


Figure C.7: Position Estimates of Leader From Follower



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure C.8: Velocity Estimates of Leader From Follower

C.2 RADAR

C.2.1 Trial 1

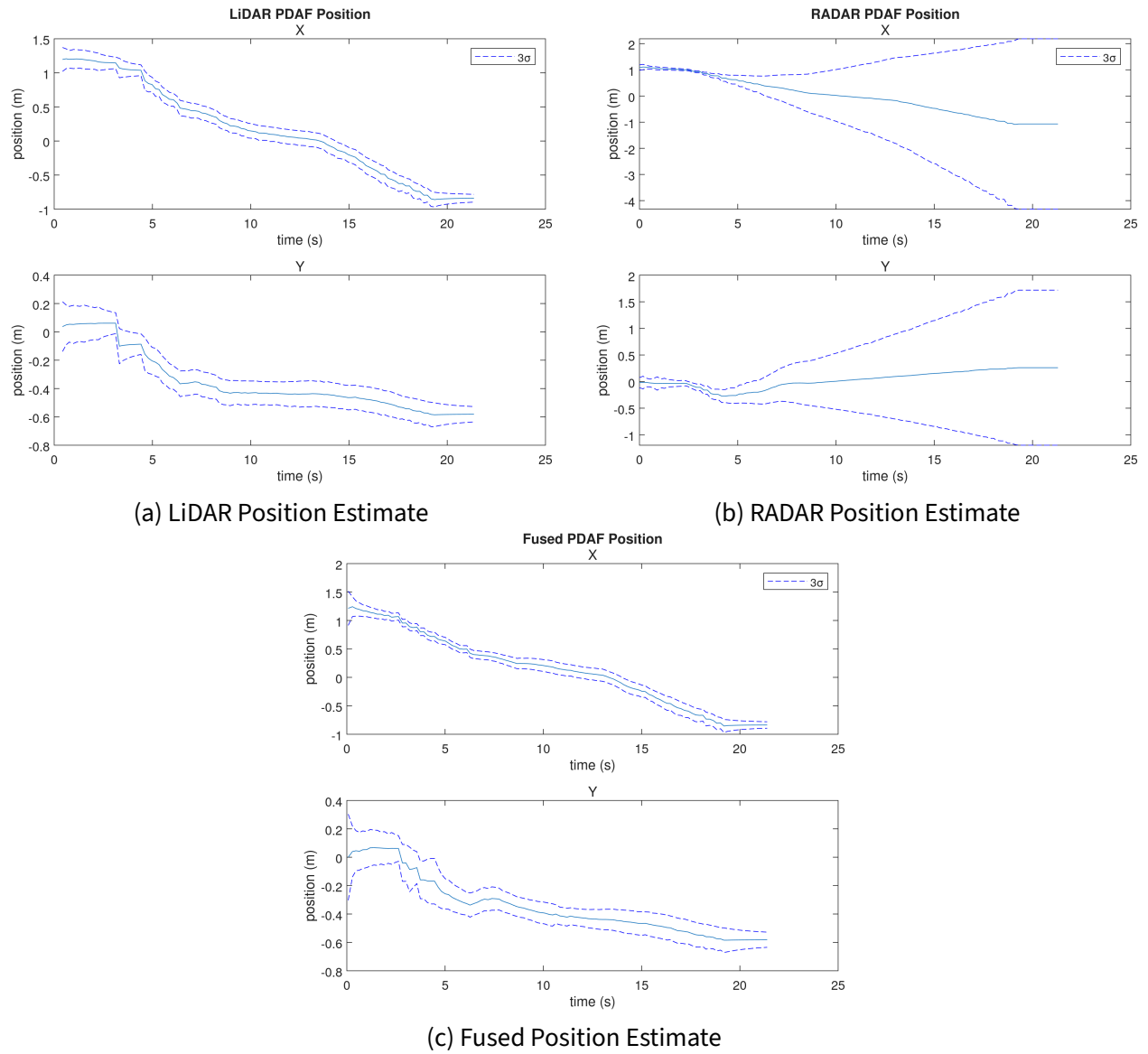
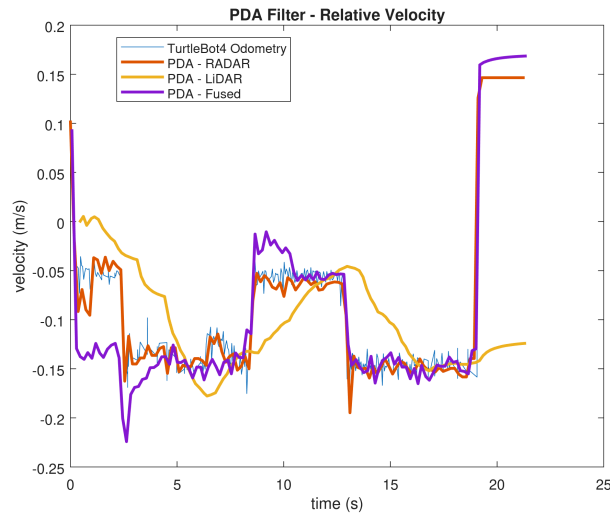
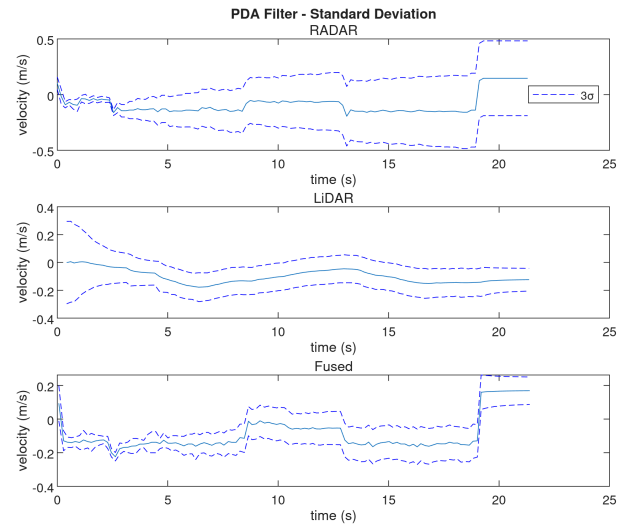


Figure C.9: Position Estimates of Leader From Follower



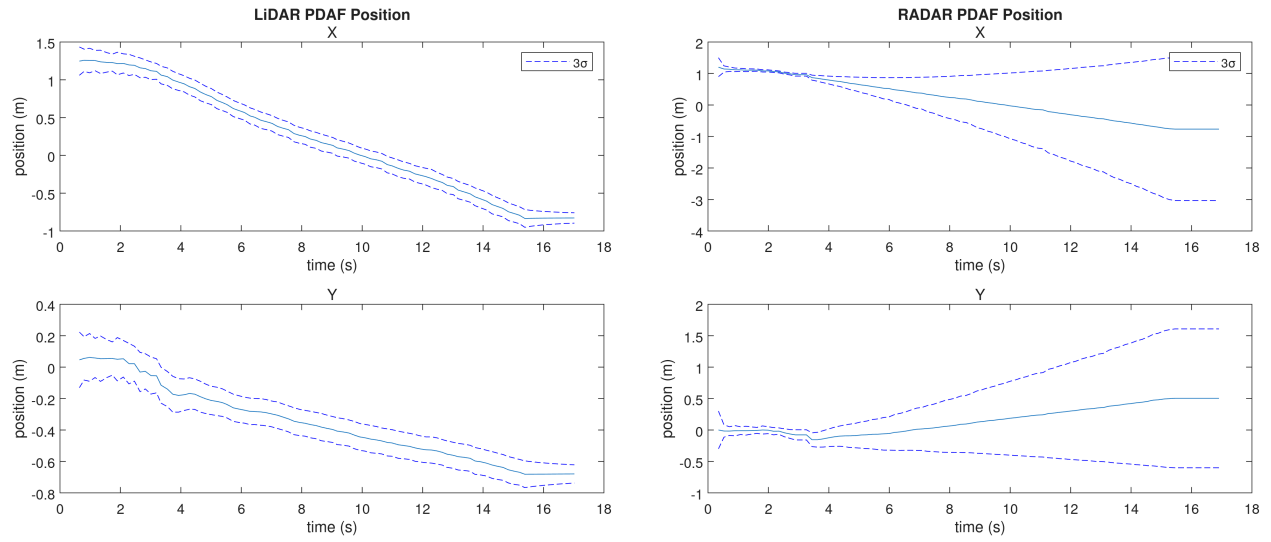
(a) Relative velocity estimates



(b) Velocity standard deviation

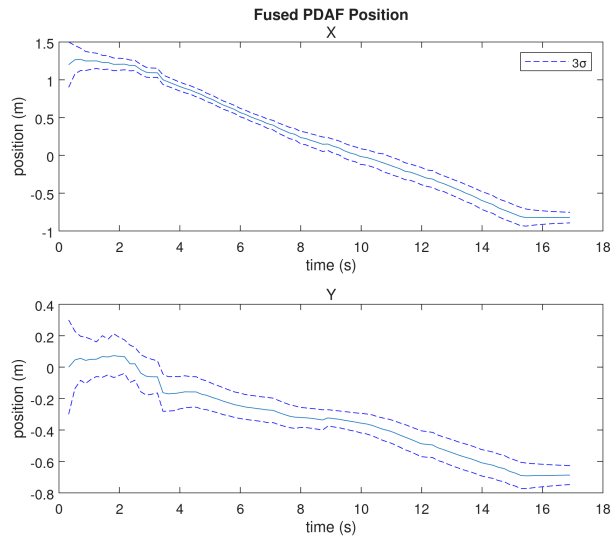
Figure C.10: Velocity Estimates of Leader From Follower

C.2.2 Trial 2



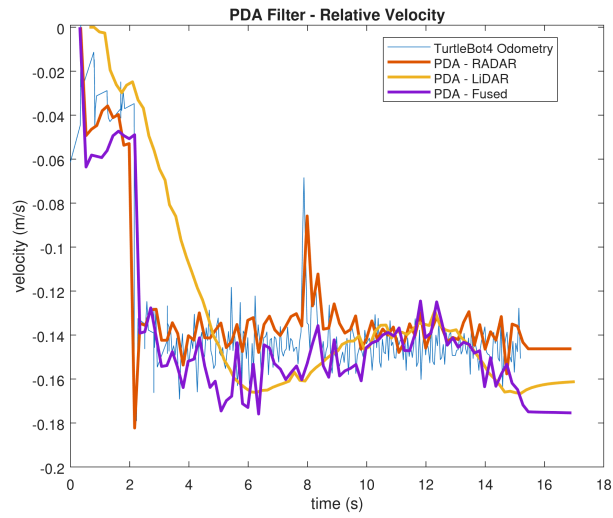
(a) LiDAR Position Estimate

(b) RADAR Position Estimate

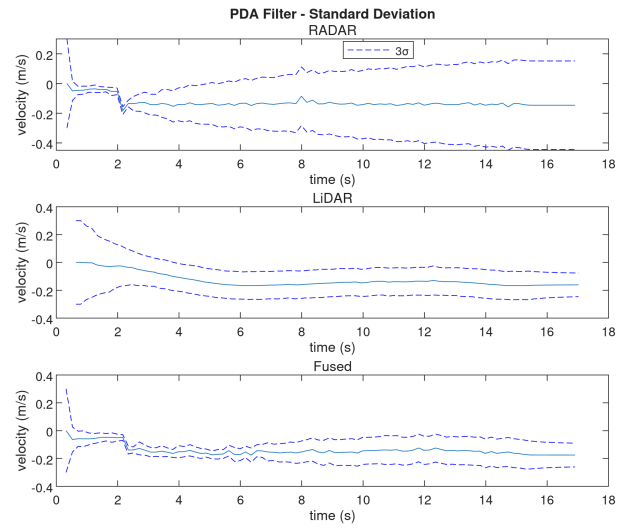


(c) Fused Position Estimate

Figure C.11: Position Estimates of Leader From Follower



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure C.12: Velocity Estimates of Leader From Follower

C.2.3 Trial 3

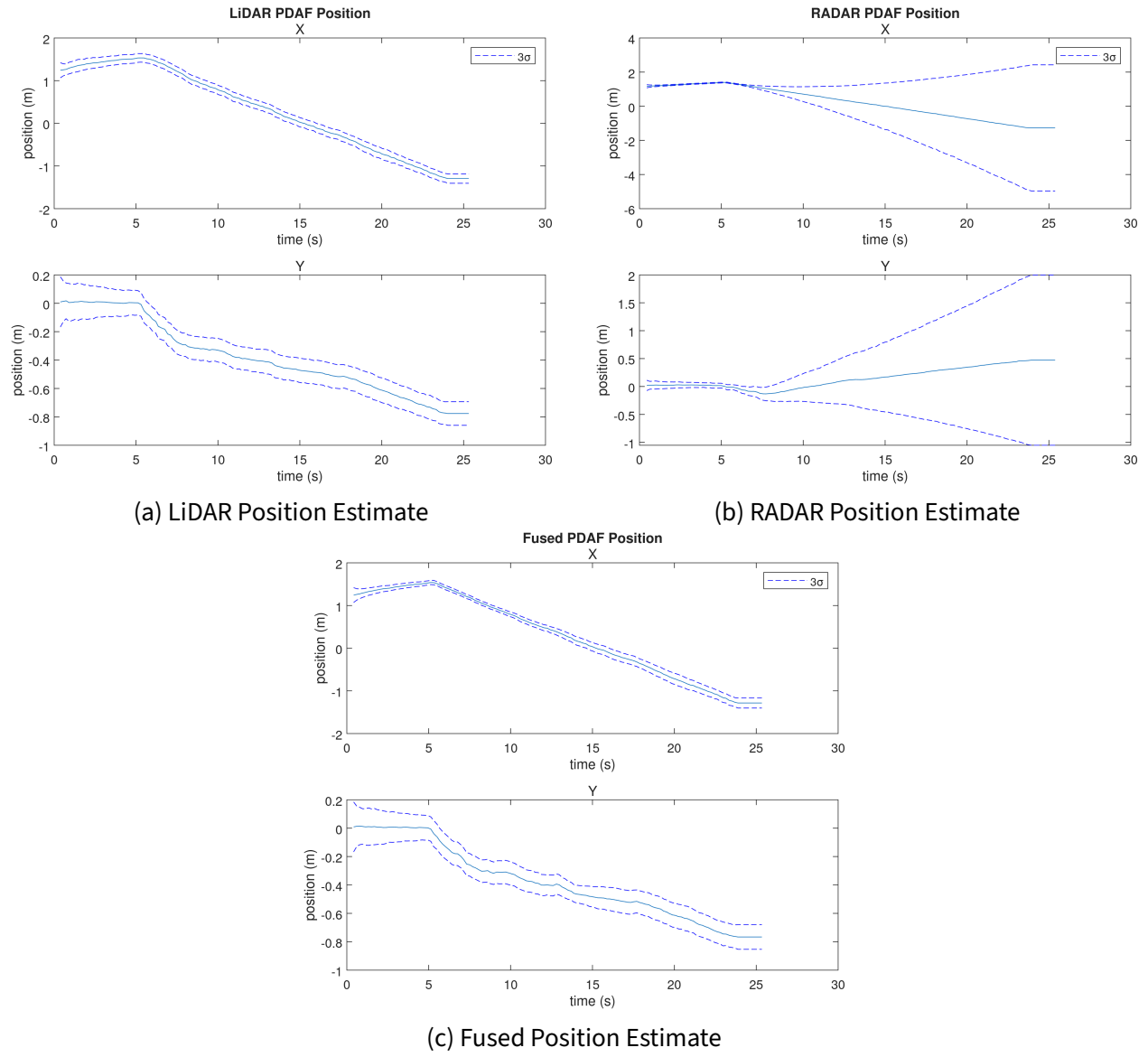
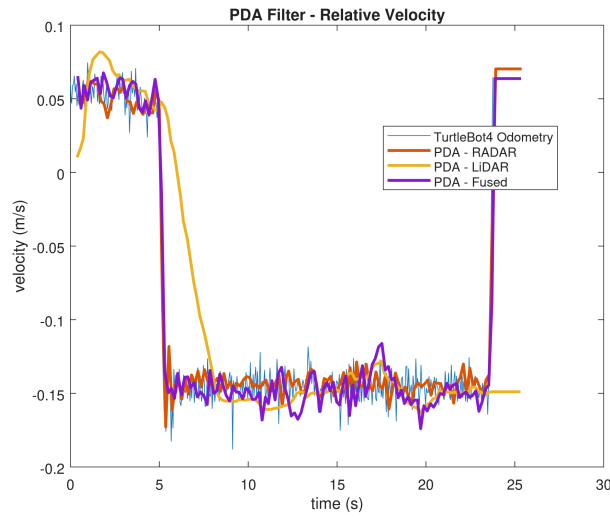
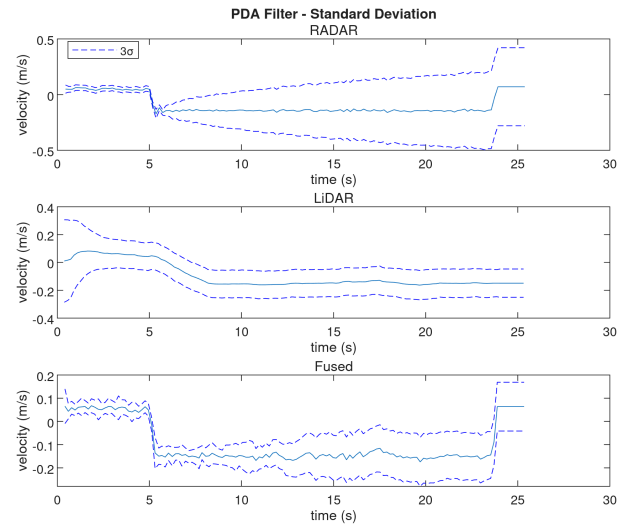


Figure C.13: Position Estimates of Leader From Follower



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure C.14: Velocity Estimates of Leader From Follower

C.2.4 Trial 4

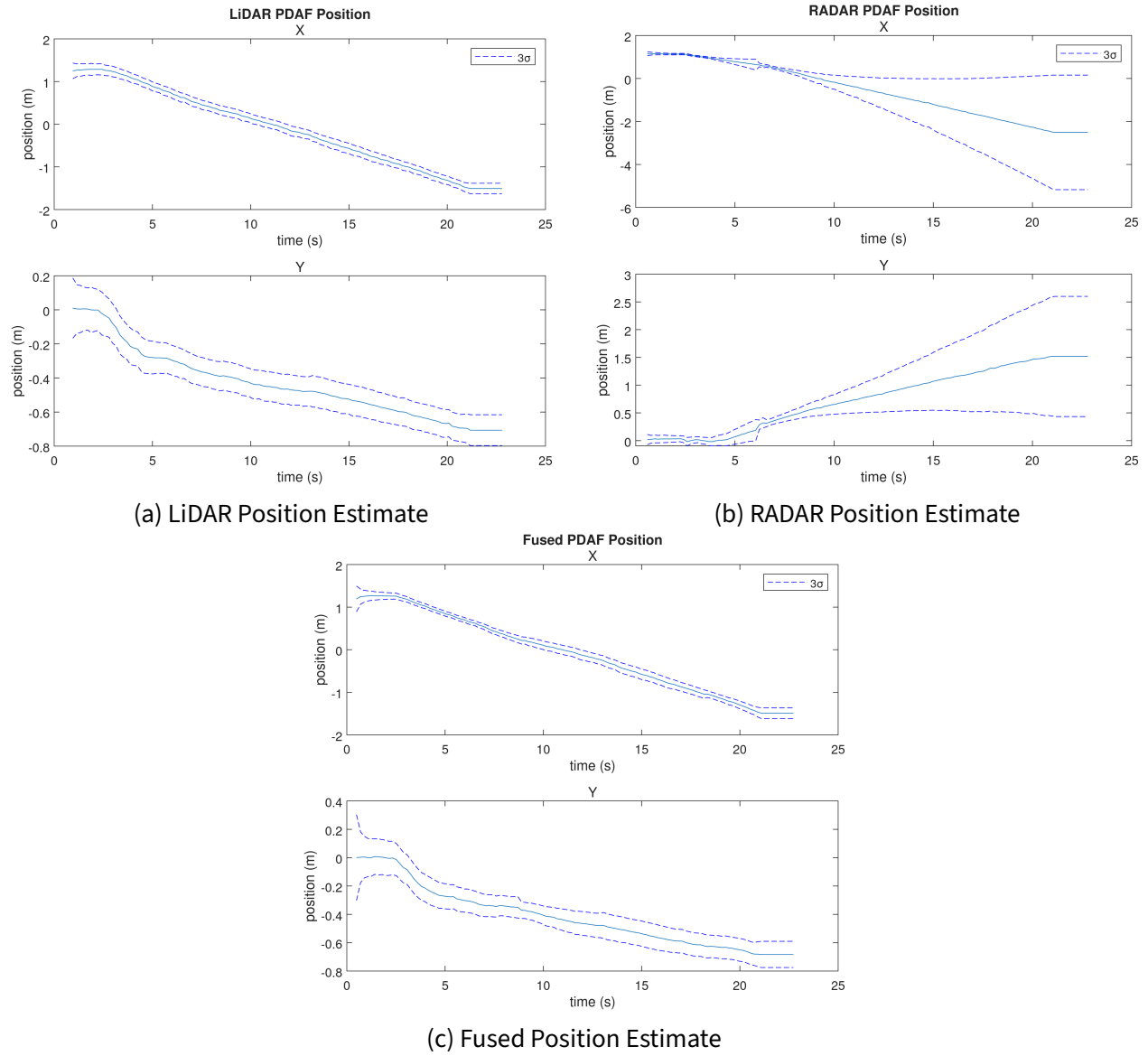
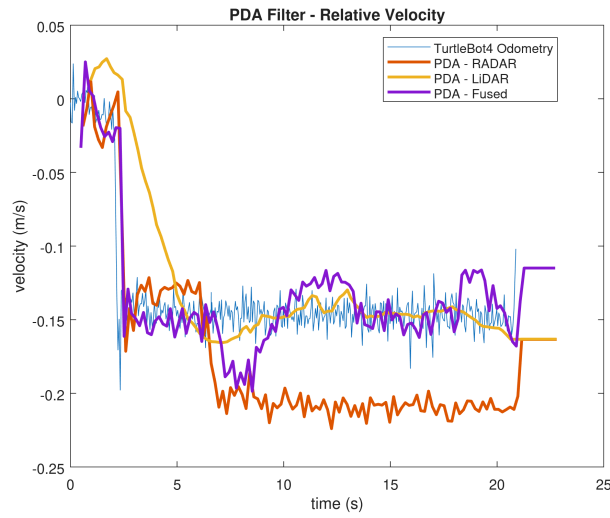
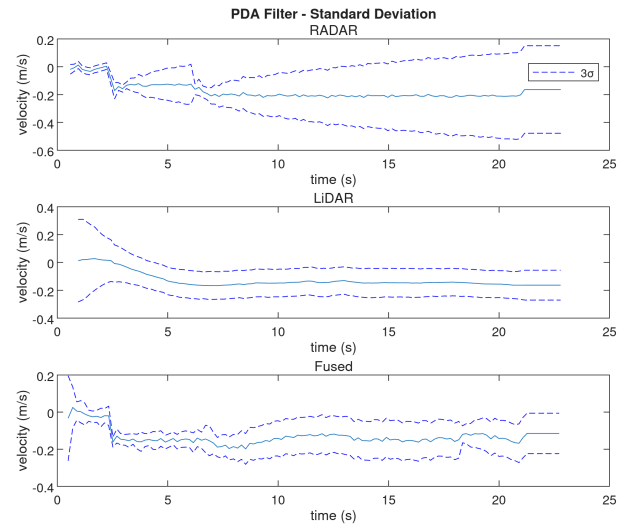


Figure C.15: Position Estimates of Leader From Follower



(a) Relative velocity estimates



(b) Velocity standard deviation

Figure C.16: Velocity Estimates of Leader From Follower