Cooperative Vehicle Perception and Localization Using Infrastructure-based Sensor Nodes

by

Mobin Khamooshi

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Mohammad Biglarbegian Associate Professor, Mechanical Engineering University of Guelph
Supervisor(s)	Amir Khajepour Professor, Mechanical and Mechatronics Engineering University of Waterloo
	Ehsan Hashemi Assistant Professor, Mechanical Engineering University of Alberta
Internal Member	Yue Hu Assistant Professor, Mechanical and Mechatronics Engineering University of Waterloo
	Zhao Pan Assistant Professor, Mechanical and Mechatronics Engineering University of Waterloo
Internal-external Member	Christopher Nielsen Professor, Electrical and Computer Engineering University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Reliable and accurate Perception and Localization (PL) are necessary for safe intelligent transportation systems. The current vehicle-based PL techniques in autonomous vehicles are vulnerable to occlusion and cluttering, especially in busy urban driving causing safety concerns. In order to avoid such safety issues, researchers study infrastructure-based PL techniques to augment vehicle sensory systems. Infrastructure-based PL methods rely on sensor nodes that each could include camera(s), Lidar(s), radar(s), and computation and communication units for processing and transmitting the data. Vehicle to Infrastructure (V2I) communication is used to access the sensor node processed data to be fused with the onboard sensor data.

In infrastructure-based PL, signal-based techniques- in which sensors like Lidar are used- can provide accurate positioning information while vision-based techniques can be used for classification. Therefore, in order to take advantage of both approaches, cameras are cooperatively used with Lidar in the infrastructure sensor node (ISN) in this thesis. ISNs have a wider field of view (FOV) and are less likely to suffer from occlusion. Besides, they can provide more accurate measurements since they are fixed at a known location. As such, the fusion of both onboard and ISN data has the potential to improve the overall PL accuracy and reliability.

This thesis presents a framework for cooperative PL in autonomous vehicles (AVs) by fusing ISN data with onboard sensor data. The ISN includes cameras and Lidar sensors, and the proposed camera-Lidar fusion method combines the sensor node information with vehicle motion models and kinematic constraints to improve the performance of PL. One of the main goals of this thesis is to develop a wind-induced motion compensation module to address the problem of time-varying extrinsic parameters of the ISNs. The proposed module compensates for the effect of the motion of ISN posts due to wind or other external disturbances. To address this issue, an unknown input observer is developed that uses the motion model of the light post as well as the sensor data.

The outputs of the ISN, the positions of all objects in the FOV, are then broadcast so that autonomous vehicles can access the information via V2I connectivity to fuse with their onboard sensory data through the proposed cooperative PL framework. In the developed framework, a KCF is implemented as a distributed fusion method to fuse ISN data with onboard data. The introduced cooperative PL incorporates the range-dependent accuracy of the ISN measurements into fusion to improve the overall PL accuracy and reliability in different scenarios. The results show that using ISN data in addition to

onboard sensor data improves the performance and reliability of PL in different scenarios, specifically in occlusion cases.

Acknowledgments

I would like to express my deep appreciation to my research supervisors, Prof. Amir Khajepour and Prof. Ehsan Hashemi, for their exceptional guidance, unwavering support, and encouraging guidance throughout my thesis journey. Their knowledge and insights have been instrumental in shaping my thoughts not only for research but on the practical side of tasks and life in general, which I highly value.

I would also like to thank my family for their unwavering support throughout my academic journey. My lovely wife, Samaneh, has been my pillar of strength, always encouraging me to pursue my dreams and providing a loving and supportive environment that enabled me to focus on my research.

I would also like to thank my beloved mother, Nasrin, whose sacrifices and dedication to my education have been instrumental in shaping who I am today. To my dear sister, Mahya, and her beautiful daughter, Mehrsana whose endless love and encouragement have been a constant source of inspiration to me.

Lastly, I would like to acknowledge my late father and gradfather, Alireza and Ali. Although they passed away during my Ph.D. program, I know they would have been proud and happy to see me complete this milestone. Their memory and the values they instilled in me continue to inspire me in my personal and professional life.

I would like to express my deep appreciation to Prof. Christopher Nielsen, Prof. Mohammad Biglarbegian, Prof. Yue Hu, and Prof. Zhao Pan for serving on my doctoral committee. Their expertise and feedback have been indispensable in helping me navigate the complexities of this research.

I would also like to express my sincere appreciation to my colleagues, who have been an integral part of my journey. Their guidance, support, and encouragement have played a vital role in my success. I would also like to extend my heartfelt thanks to my dear colleagues and friends Ali Shahidi, Ehsan Mohammadbagher, Neel P. Bhat, Amin Habibnejad Korayem, Reza Hajiloo, Mehdi Zabihi, Ahmad R. Alghooneh, Mehdi Abroshan, Ruihe Zhang, and Yukun Lu.

Lastly, I would like to express my gratitude to the funding agencies, including the Ontario Research Fund (ORF) and the Natural Sciences and Engineering Research Council of Canada (NSERC), for their financial support, which has made this thesis possible.

Dedication

To my beloved family.

I would do anything for them and will forever be grateful for their love and support.

Table of Contents

List of Figures xi

List of Tables xiv

List of Abbreviations xv

Nomenclature xvii

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Thesis Objectives 4

1.3 Main Contributions 5

1.4 Thesis Outline 5

Chapter 2 Literature Review and Background 7

2.1 Introduction 7

2.2 Multi-sensor Fusion 7

2.3 Infrastructure-based navigation in robotics 10

2.3.1 Visual-based PL in robotics 12

2.3.2 PL by artificial markers 13

2.4 Infrastructure-based navigation in vehicles 15

2.5 Multi-sensor Fusion in Cooperative PL 18

2.6 Point Set Matching 22

2.7 Pose Estimation 23

2.7.1 Camera-based Pose Estimation 24

2.8 Summary and research gap 25

Chapter 3 Infrastructure-based Sensor node 27

3.1 Introduction 27

3.2 Overview of the sensor fusion at node 28
3.3 Vehicle motion model 30
3.3.1 Constant velocity (considering heading angle) 30
3.3.2 Combined longitudinal and lateral model 31
3.3.3 Kinematic constraints 33
3.4 Sensor uncertainty formulation 35
3.4.1 Sensor noise/uncertainty 35
3.4.2 Correlation between sensor uncertainty and distance 35
3.5 Wind-induced Motion Compensation 37
3.5.1 Camera-based Pose Estimation 38
3.5.2 Lidar-based Pose Estimation 40
3.5.3 Pose Estimation using Camera/Lidar Augmentation 41
3.6 Dynamic Motion Model of Light Post 42
3.7 Unknown Input Observer 45
3.8 Simulation results 47
3.8.1 Sensor and Simulation Parameters 47
3.8.2 Scenario 1 – Constant Velocity 49
3.8.3 Scenario 2 – Turning Right 50
3.8.4 Motion Compensation Results 52
3.9 Summary 55
Chapter 4 Infrastructure-aided PL for urban driving 57
4.1 Introduction 57
4.2 Data Correspondence 57
4.2.1 Rigid Registration for Data Correspondence 58

4.2.2 Point Set Matching for Data Correspondence 61
4.2.3 Coherent Point Drift for Data Correspondence 63
4.3 Estimation formulation for Fusion 67
4.3.1 Optimization-based methods 67
4.3.2 Filter based methods 67
4.4 Cooperative PL 68
4.4.1 Kalman- Consensus Filter 70
4.5 Simulation results 74
4.5.1 Constant velocity motion 74
4.5.2 Cooperative PL Using One Sensor Node 76
4.5.3 Cooperative PL Using Two Sensor Nodes 79
4.5.4 Constant Acceleration maneuver 80
4.5.5 Data Correspondence Simulation Results 82
4.6 Summary 87
Chapter 5 Experimental Studies 89
5.1 Introduction 89
5.2 Experimental Results 92
5.3 Summary 98
Chapter 6 Conclusions and Future Work 99
6.1 Conclusions 99
6.2 Future Work 100
References 103

List of Figures

Figure 1.1: Comparison of the FOV of the onboard camera with the ISN camera 1
Figure 1.2: Overall block diagram representation for cooperative PL using an ISN 3
Figure 2.1: Distributed fusion architecture 7
Figure 3.1: ISN node holds one camera and one Lidar 27
Figure 3.2. The overlapping covered area of two ISN cameras, simulated in MATLAB/Simulink 28
Figure 3.3: Structure of the ISN node 29
Figure 3.4: Illustration of the vehicle and the states of the model 31
Figure 3.5: Kinematic bicycle model 32
Figure 3.6: Bounding Box of the vehicle in different distances 35
Figure 3.7: Location, and the density of Lidar 1 and 2 37
Figure 3.8: Example of reference points in the camera frame 39
Figure 3.9: General framework of camera-based pose estimation 39
Figure 3.10: General framework of Lidar-based pose estimation 40
Figure 3.11: General framework of pose estimation using camera/Lidar data 41
Figure 3.12: Dynamic model of light post as a vertical cantilever beam in the *x*-direction 43
Figure 3.13: First two modes of a vertical cantilever beam 43
Figure 3.14: Simulated scenario in MATLAB/Simulink Automated Driving. 47
Figure 3.15: The visualization of the estimation error 48
Figure 3.16: Estimation results of (a)ISN 1 and (b) ISN 2 in *X* and *Y* directions for scenario 1. 49
Figure 3.17: Estimation result of the ISN 1 for turning scenario 50
Figure 3.18: Estimated Trajectory of the vehicle performing scenario 2. 51
Figure 3.19: Estimation error statistics using different sensor nodes for scenario 2. 51
Figure 3.20: Pose estimation for the wind loading applies in 0.5 and 1 Hz modes 53
Figure 3.21: Orientation estimation for the wind loading applies in 0.5 and 1 Hz modes 53
Figure 3.22: Position estimation for the wind loading applies at 1, 3, and 10 Hz modes 54
Figure 3.23: Orientation estimation for the wind loading applies at 1, 3, and 10 Hz modes 55
Figure 4.1: Position of objects in different coordinates 58
Figure 4.2: Block diagram of data correspondence module 60
Figure 4.3: Block diagram of cooperative PL 68
Figure 4.4: Constant velocity scenario 74

Figure 4.5: Noise and measurement covariance illustration 75
Figure 4.6: Sensor Configuration in the third scenario, simulated in MATLAB 76
Figure 4.7: Estimation error for localizing the ego vehicle 77
Figure 4.8: Estimation error statistics using different sensor nodes 77
Figure 4.9. Estimation error of the other vehicle's position 78
Figure 4.10: Sensor locations in the second scenario, simulated in MATLAB 79
Figure 4.11: Estimation error of using two fixed sensors 79
Figure 4.12: Longitudinal position of the vehicles in the constant acceleration scenario 80
Figure 4.13: The relative distances of the vehicles in sensor coordinates 81
Figure 4.14: Estimation results of the vehicles 81
Figure 4.15: Estimated position of the objects including the ego vehicle in different coordinates 82
Figure 4.16: Estimated position of objects in coordinates of sensor 2 (ISN) and sensor 1 (ego vehicle) 83
Figure 4.17: Sensor registration by using the ICP algorithm 84
Figure 4.18: Estimated position of objects in different coordinates after convergence 84
Figure 4.19: Example of the local minimum issue 85
Figure 4.20: Estimated heading angle in a "local minimum issue" case 85
Figure 4.21: Statistic results of the local min. issue for two point set matching algorithms 86
Figure 4.22: Estimated heading angle of the AV using the improved CPD algorithm 87
Figure 5.1: ISN, including two cameras and a Lidar, mounted on a scissor lift 89
Figure 5.2: WATonoBus, an autonomous shuttle bus with Vehicle to Infrastructure (V2I) connectivity 90
Figure 5.3: Intersection scenario with WATonoBus, another vehicle, and a pedestrian 90
Figure 5.4: Covered area by the ISN's cameras and Lidar 91
Figure 5.5: Image frames of the ISN's cameras 91
Figure 5.6: Estimated trajectories of dynamic objects by the ISN 92
Figure 5.7: Estimated trajectories of dynamic objects by WATonoBus 93
Figure 5.8: Estimated position of object 2 by the ISN and WATonoBus 94
Figure 5.9: Estimated position of the other vehicle by the ISN and WATonoBus 95
Figure 5.10: Estimated position of the other vehicle during occlusion 96
Figure 5.11: Measurement covariance regarding the other vehicle 97

Figure 6.1: Images captured by the simulated camera in (a) rainy weather and (b) a different light condition 101

List of Tables

Table 2.1: Summary of infrastructure–based PL works in-vehicle application 18
Table 4.1: ID correspondence between the objects in the ISN and the-vehicle 59

List of Abbreviations

Abbreviation	Meaning
CI	Covariance Intersection
CKF	Constrained Kalman Filter
DoF	Degree of Freedom
EI	Ellipsoidal Intersection Method
EKF	Extended Kalman Filter
FOV	Field of View
ICP	Iterative Closest Point
IMG	Image
ISN	Infrastructure-based sensor
IEA	Internal Ellipsoidal Approximation
KCF	Kalman- Consensus filter
LE	Largest Ellipsoid
NN	Nearest Neighbor
PL	Perception and Localization
PC	Point cloud
PnP	Perspective n Points
QR	Quick Response
RFID	Radio Frequency Identification
RSSI	Receive signal strength indicator
RSU	Roadside Unit
SLAM	Simultaneous robot localization and map-building
UIO	Unknown Input Observer
UTM	Universal Transverse Mercator

V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
ZOH	Zero-order hold

Nomenclature

Local fusion

$\boldsymbol{k}$	Time step
$\boldsymbol{si}$	Sensor node
$\boldsymbol{\Delta t}$	Sampling time
$\boldsymbol{z_k^C}$	Camera-based positioning output
$\boldsymbol{R_k^C}$	Measurement covariance of Camera-based positioning
$\boldsymbol{z_k^L}$	Lidar-based positioning output
$\boldsymbol{R_k^L}$	Measurement covariance of Lidar-based positioning
$\boldsymbol{X_k}$	State vector
$\boldsymbol{u_k}$	Input vector
$\boldsymbol{F}$	System matrix regarding tracking objects (motion model of the vehicle)
$\boldsymbol{G}$	Input matrix regarding tracking objects (motion model of the vehicle)
$\boldsymbol{\omega_k}$	Process noise
$\boldsymbol{Q_k}$	Process covariance matrix
$\boldsymbol{Z_k^{si}}$	Measurement of the sensor node including the positions of objects
$\boldsymbol{H^{si}}$	Observation matrix of the sensor node
$\boldsymbol{v_k^{si}}$	Observation noise of the sensor node
$\boldsymbol{R_k^{si}}$	Measurement covariance of the sensor node
$\boldsymbol{\widetilde{X}_k^{si}}$	Estimated state by each sensor node
Kinematic Constraints	
$\{\widetilde{\boldsymbol{X}}\}_{\boldsymbol{k-N+1}}^{\boldsymbol{k-1}}$	Fixed-sized sequence of the past estimated states
$\boldsymbol{n_p}$	Window size of previous estimated states to calculate the predicted state
$\boldsymbol{X_k^*}$	Predicted state based on the previous states

$\boldsymbol{T}$	Weighting matrix regarding kinematics constraints
Covraince Scaling	
$\boldsymbol{d_{i,k}}$	Relative distance between the object and the sensor node
$\boldsymbol{a_n}$	Constant parameter of the covariance scaling
Motion Models	
$\boldsymbol{X_k^0}$	Global *x* position of the ego vehicle
$\boldsymbol{Y_k^0}$	Global *y* position of the ego vehicle
$\boldsymbol{\psi_k^0}$	Heading angle of the ego vehicle
$\boldsymbol{V_k^0}$	Longitudinal velocity of the ego vehicle
$\boldsymbol{\dot{\psi}_k^0}$	Yaw rate of the ego vehicle
$\boldsymbol{\delta}$	Steering angle of the ego vehicle
$\boldsymbol{L}$	Wheelbase of the ego vehicle
Motion Compensation Module	
$(\boldsymbol{x^c}, \boldsymbol{y^c}, \boldsymbol{z^c})$	Position of the camera
$(\boldsymbol{\zeta^c}, \boldsymbol{\theta^c}, \boldsymbol{\varphi^c})$	Orientation of the camera
$\boldsymbol{P_i} = [\boldsymbol{X_i}, \boldsymbol{Y_i}, \boldsymbol{Z_i}]^{\boldsymbol{T}}$	World coordinates of the reference points in the pose estimation
$\boldsymbol{r_0^{si}}$	Initially calibrated rotation matrix of the sensor node
$\boldsymbol{t_0^{si}}$	Initially calibrated translation vector of the sensor node, estimated using only sensor data
$\boldsymbol{r_k^{si}}$	Rotation matrix of the sensor node, estimated using only sensor data
$\boldsymbol{t_k^{si}}$	Translation vector of the sensor node, estimated using only sensor data
$\boldsymbol{\hat{r}_k^{si}}$	Rotation matrix of the sensor node, estimated by using the motion model and sensor data
$\boldsymbol{\hat{t}_k^{si}}$	Rotation matrix of the sensor node, estimated by using the motion model and sensor data
$\boldsymbol{x^{si}}$	*x* position of the sensor node
$\boldsymbol{y^{si}}$	*y* position of the sensor node

$\boldsymbol{\theta^{si}}$	Pitch angle of the sensor node
$\boldsymbol{\zeta^{si}}$	Roll angle of the sensor node
$\boldsymbol{X^{si}}$	State vector of the sensor node mounted on the light post
$\boldsymbol{\bar{A}}$	System matrix regarding motion model fo the light post
$\boldsymbol{\bar{B}}$	Input matrix regarding motion model fo the light post
$\boldsymbol{\bar{C}}$	Observation matrix regarding motion model fo the light post
$\boldsymbol{h^{si}}$	Height of the senor node mounted on the light post
$\boldsymbol{\omega_{x,1}^{si}}$	First natural frequency of the light post
$\boldsymbol{E^{si}}$	Elastic modulus of the light post
$\boldsymbol{I_x^{si}}$	Moment of inertia of the light post in the *x* direction
$\boldsymbol{\xi^{si}}$	Damping ratio of the light post

Data Correspondence Module

$\boldsymbol{p_{n,k}^{G}}$	Estimated global position of the object by the sensor node
$\boldsymbol{N_k}$	Set of all estimated global positions of the objects by the infrastructure sensor node
$\boldsymbol{ID_k(n)}$	ID number of the object detected by the sensor node
$\boldsymbol{p_{m,k}^{rel}}$	Estimated relative position of the object by the autonomous vehicle
$\boldsymbol{M_k}$	Set of all estimated relative positions of the objects by the autonomous vehicle
$\boldsymbol{ID_k(m)}$	ID number of the object detected by the autonomous vehicle
$\boldsymbol{C_k = \{c_{m,n}\}_k}$	ID Correspondence matrix between the autonomous vehicle and the sensor node
$\boldsymbol{r_k^{MN}}$	Rotation matrix to alight point sets M_k and N_k
$\boldsymbol{t_k^{MN}}$	Translation vector to alight point sets M_k and N_k
$\boldsymbol{\varepsilon}$	Convergence criteria
$\boldsymbol{j_{max}}$	Maximum number of iterations
$\boldsymbol{\sigma^2}$	the isotropic covariance in the CPD algorithm

w	weight of the uniform distribution in the CPD algorithm
$\mathcal{T}$	Transformation parameters consisting of rotation and translation
Cooperative PL	
$p_{0,k}^{G}$	Global position of the autonomous vehicle
H_{tj}^{si}	Observation matrix regarding the object *j* by sensor *i*
H_{tj}^{AV}	Observation matrix regarding the object *j* by the ego vehicle
u_k^{si}	Weighted measurement of sensor node *i*
U_k^{si}	Information matrix of sensor node *i*
m_k^{si}	Information package broadcast by sensor *i*
y_k^0	Fused weighted measurement formed by the autonomous vehicle
S_k^0	Fused information matrix formed by the autonomous vehicle
$\widehat{X}_k^{si}$	Estimated state by the sensor node *i*
P_k^{si}	the prior estimated covariance matrix by the sensor node *i*
$\widehat{X}_k^0$	Estimated state by the autonomous vehicle cooperating with other sensor nodes
N^0	Neighboring sensor node of the ego vehicle
K_k^{si}	Optimal Kalman gain
M^i	the estimated covariance matrix by the sensor node *i*
C_i	Consensus gain

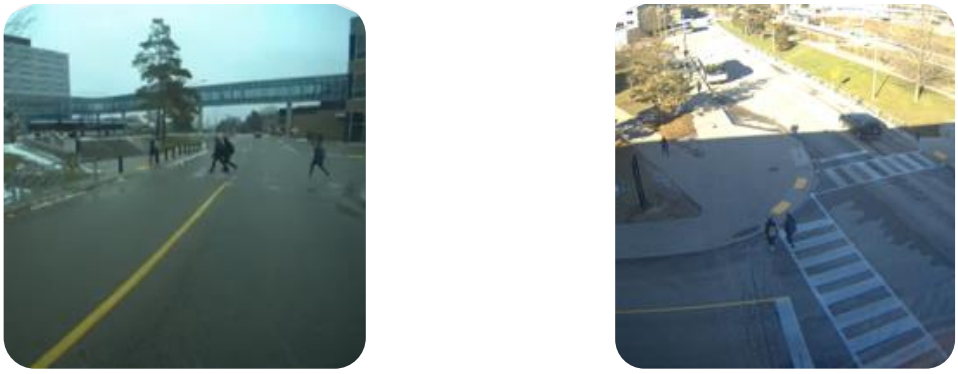
Chapter 1
Introduction

The focus of this thesis is on improving the PL of autonomous vehicles (AVs) using ISN data. To elaborate, the position and the heading angle of the objects estimated by ISN are fused with an AV's onboard sensor data to improve the overall PL of the vehicle to eliminate any occluded or partially occluded objects.

1.1 Motivation

Road safety and traffic management are two main potential benefits of the intelligent transportation system. Intelligent traffic management can decrease the cost of travel time, consumed fuel, and CO_2 emissions. In such an intelligent transportation system, autonomous vehicles (AVs) need to connect to other vehicles and also infrastructure. With the growing development of communication technology, there is an opportunity to have a connected vehicle-infrastructure system. Connected vehicles can communicate with each other and infrastructure units to improve automated driving and transportation safety.

ISN nodes are mounted high over the ground level, therefore they have a different field of view and are less likely to suffer from occlusion, compared to the onboard sensors as can be seen in Figure 1.1. Besides, they can provide more accurate positioning since they are fixed at a known location. In this thesis, the ISN nodes consist of two cameras and one Lidar with the same FOV.



(a) Onboard camera (b) ISN camera with a top-down view

Figure 1.1: Comparison of the FOV of the onboard camera with the ISN camera

ISN nodes can be used in two main applications: urban driving and autonomous fleet operation. In the first application, the ISN data are transmitted that AVs that can fuse the information with their onboard sensor data. Using this approach, the safety of AVs can potentially be improved in complex urban environments, especially intersections and roundabouts where occlusion is common. To avoid the occlusion, additional information provided by ISN can be cooperatively utilized with the vehicle-based PL. Using ISN and transmitting the data to the vehicles, the local occlusion can be reduced or eliminated.

Another motivation to use ISN is autonomous fleet operation. An autonomous fleet can be found in public transportation, airports, and resorts among others. In such applications, the PL of the environment and also the connectivity of the fleet are two essential parts. Using the ISN in the autonomous fleet can further improve safety and performance. The PL is carried out by using ISN and the data is then sent to the vehicles. This platform has the potential of reducing the cost and the complexity of typical vehicle-based automated driving.

The overall diagram representation of cooperative PL using the ISN is depicted in Figure 1.2. The different parts of the cooperative PL, the inputs and outputs of each part, and their interface and interaction are illustrated. In this work, two main components of the cooperative PL are the AV and the ISN.

In each ISN, the sensor suite consists of camera(s) and a Lidar with the same FOV. The sensor data in addition to the vehicle motion model and kinematic constraints are used for the fusion at the sensor node data. The output of the ISN node is the estimated positions and the velocities of all objects. One key aspect of the proposed fusion algorithm is the addition of the sensors' range-dependent accuracy. To do so, the estimated positions from the fusion block are used to calculate the measurement covariance by using the relative distance of the objects from the sensor node.

Another important module of the ISN is motion compensation. The ISN oscillates due to external disturbances like wind. Therefore, its extrinsic parameters deviate from their initial calibrated values. The motion compensation module implements an unknown input observer that utilizes the sensor data in addition to the dynamic motion of the light post to estimate the deviated extrinsic parameters.

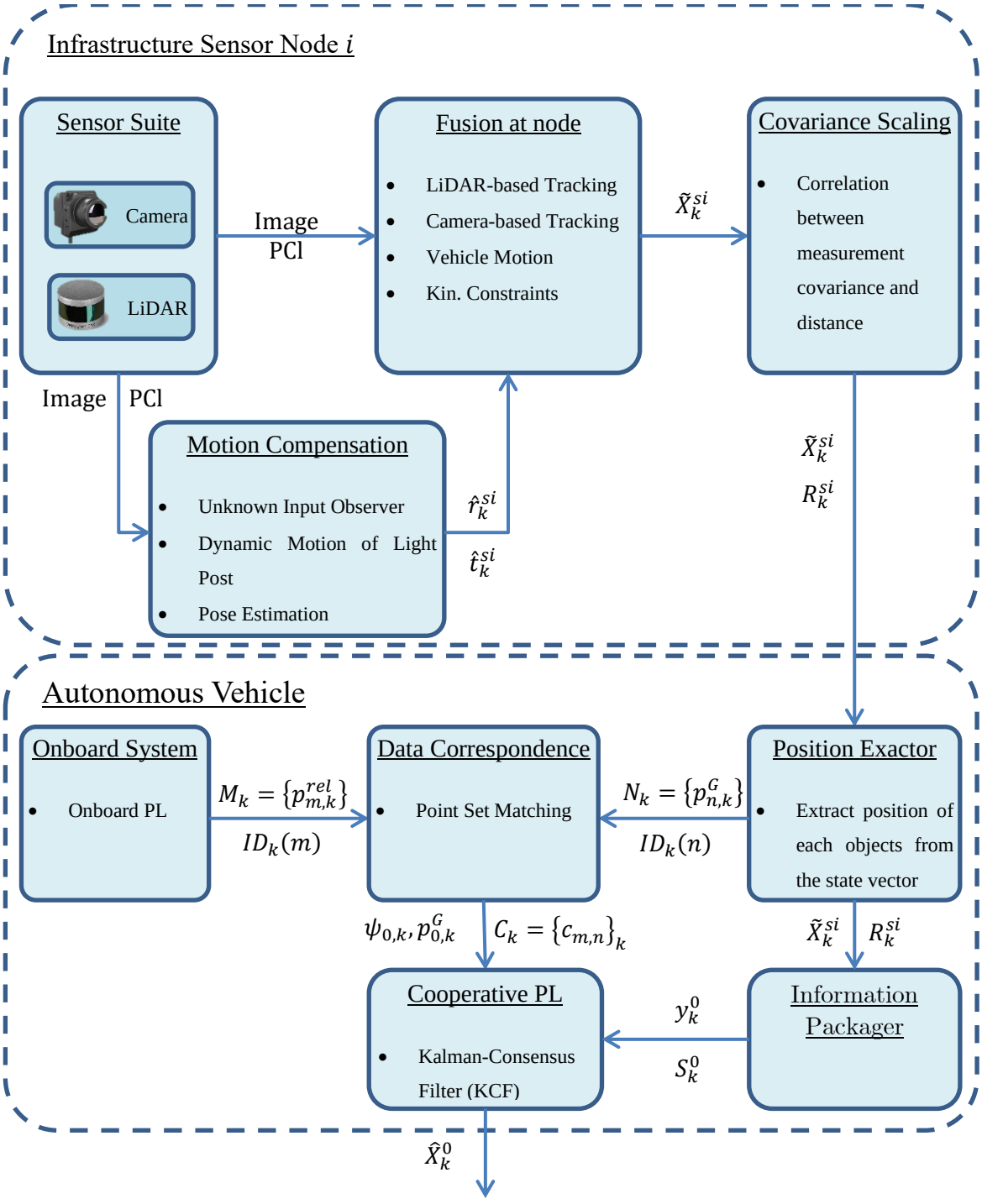


Figure 1.2: Overall block diagram representation for cooperative PL using an ISN

The output of the ISN node is the estimated positions of all objects with the related range-dependent covariance. This information is broadcast so that any AV can receive it and fuse it with its onboard data. The AV also has positioning awareness of its surrounding environment, meaning that each AV knows the relative position of objects in its coordinate in addition to its assigned ID. Although, IDs assigned to the objects in ISN's and ego vehicle's coordinates are different. Therefore, the correspondence between the estimated positions of the objects by the ISN and the AV should be found. For this purpose, the positions of the objects received by the AV are extracted and sent to the data correspondence module. The point set matching algorithm inside the data correspondence uses the above data in addition to the relative positions of the objects estimated by the AV itself to find the correspondence and also the rotation, and translation of the onboard coordinate with respect to the ISN coordinate. Once the above rotation and the translation are found, the global position and heading angle of the AV can easily be calculated.

Another block of the cooperative PL is the information packager. Its main task is to form an information vector based on the estimation positions of the objects and their related measurement covariance which will be discussed in detail in Chapter 4.

When the correspondence between estimated objects in the ISN coordinate and the onboard coordinate is found, Then, the KCF is performed to find the cohesive estimation using both ISN and onboard data. The inputs of this part are the estimated positions, related measurement covariance, and correspondence. The output is the balanced consensus on the estimation of different sensor nodes including the ISNs and the onboard AV's sensor node.

1.2 Thesis Objectives

The main objective of this thesis is to develop and implement a cooperative PL system using ISN nodes that can be applied to two main applications: 1. Safe handling of busy urban intersections and, 2. Autonomous fleet operation.

In the first application which is for the safe handling of busy urban intersections, the sensor node outputs are broadcast so that AVs can receive and use the information cooperatively with the onboard sensor data to improve the PL.

In the second application, autonomous fleet operation, the output of each ISN node can be used in a cloud-based autonomous fleet operation. The data from the sensor nodes are used in a central PL with

minimum vehicle-based sensors. This approach has the potential of reducing the costs and complexity of individual control of the fleet vehicles.

Another objective of this thesis is to study the distance parameter affecting the PL accuracy of the ISN. The ISN with the stationary camera(s) and Lidar(s) have a distance-dependent nature in their accuracy that will be studied and incorporated into the cooperative PL systems designed in this thesis.

1.3 Main Contributions

The main contribution of this thesis is a novel solution for improving the perception and localization of autonomous vehicles in urban driving by fusing infrastructure-based sensor node data with onboard sensor data. By using infrastructure-based sensor nodes which are less likely to suffer from occlusion, the proposed solution addresses the issue of occlusion which is common in urban areas, especially intersections. Therefore, it enhances the overall perception and localization of autonomous vehicles, ultimately improving the safety of autonomous driving.

In addition, this thesis proposes a unique approach for fusing Lidar and camera data that takes into account the vehicle's motion, kinematic constraints, and distance-dependent covariance scaling. By incorporating these factors into the data fusion process, the proposed approach surpasses previous methods that did not consider these elements.

Finally, the research proposes a motion compensation module that addresses the problem of time-varying extrinsic parameters for infrastructure-based sensor nodes. To address this issue, an unknown input observer is developed that utilizes the motion model of the light post as well as the sensor data, a novel technique that has not been reported in the literature.

Overall, this research provides a novel solution for improving the perception and localization of autonomous vehicles by integrating infrastructure-based sensor node data with onboard sensor data, and by developing novel approaches for fusing Lidar and camera data and addressing the problem of time-varying extrinsic parameters for infrastructure-based sensor nodes.

1.4 Thesis Outline

The remainder of this proposal is organized as follows. Chapter 2 contains a brief review of the related literature and background information on cooperative PL in robotics and vehicle applications. In addition, multi-sensor fusion methods and their common challenges will be reviewed.

Chapter 3 will present the structure of the ISN node and its components. Then the fusion of the sensor data with the vehicle motion will be performed at the node. The nature of the range-dependent accuracy of the sensor will be also explained. The correlation between the distance of the object and the sensor with the measurement covariance will be studied. This chapter also introduces the motion compensation module that aims to estimate the position and orientation of the sensor node in the presence of external disturbances such as wind.

In Chapter 4, Kalman-Consensus Filter will be described as a distributed fusion algorithm to fuse the transmitted infrastructure-based data with the onboard sensor data. In this chapter, observability analysis will be discussed as well. This chapter presents the data association algorithm required to correspond the detected objects in the ISN's coordinate with the onboard AV's coordinate. The results of simulating an infrastructure-based PL algorithm for autonomous urban driving. The results can be used to evaluate the measurement noise uncertainty in different conditions. The preliminary results of implementing the cooperative PL algorithm for urban autonomous driving will be discussed. The results of fusing the ISN data with onboard sensor data are provided for different scenarios. the measurement noise uncertainty in different conditions.

Chapter 5 includes the experimental setup and the implementation of the proposed framework in Chapters 3 and 4 for real data. The experimental setup and sensors are presented in detail. Also, the performance of the developed algorithm in a real-time application is examined.

Finally, Chapter 6 provides the conclusions of this thesis. It also suggests future works and potential directions for future research.

Chapter 2
Literature Review and Background

2.1 Introduction

Reliable and accurate PL are important components of autonomous vehicle operations. In such autonomous applications, using ISN nodes can be beneficial. For example, incorporating the ISN data into the vehicle-based PL can add additional information about the blind spots of the onboard sensors. Fusing the above-mentioned data, local occlusion can be avoided in crowded areas such as intersections. In this chapter, multi-sensor fusion methods will be comprehensively reviewed. The research gap and a summary of this chapter are provided at the end.

2.2 Multi-sensor Fusion

Data fusion architectures are generally classified into two types:

- Centralized architectures: in this approach, raw data from multiple sensors, without any local filtering or fusion, are directly transmitted to the central fuse node.
- Distributed architectures: in this technique, each sensor node uses its raw data to provide a local estimation in terms of the mean and covariance. The estimates are then sent to the central node. The schematic diagram of a distributed fusion architecture can be seen in Figure 2.1.

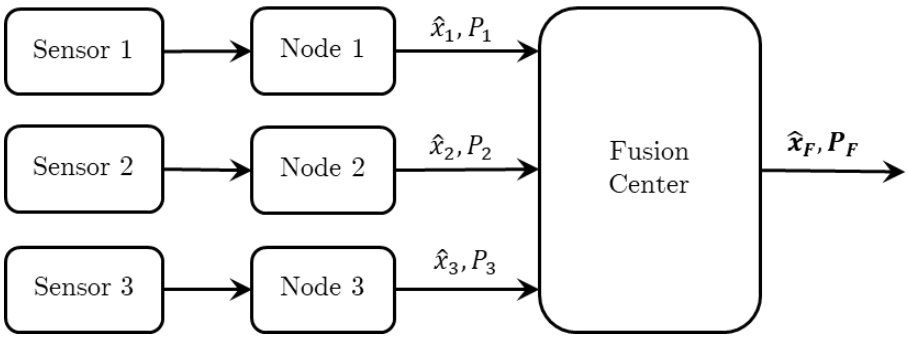


Figure 2.1: Distributed fusion architecture

For cooperative vehicle PL, data fusion needs to address several challenges. Measurements always have some level of uncertainty. Data fusion algorithms use data redundancy to reduce uncertainty [1] for localization applications.

Probabilistic methods [2], like Bayesian fusion, rely on probability distribution functions to express data uncertainty. Bayes' estimator calculates the posterior (conditional) probability distribution of the hypothetical state x_k at time k given the set of measurements, $Z^k = \{z_1, \dots, z_k\}$ and the prior distribution. Bayes' estimation can be applied at every time step *k*, updating the probability distribution by fusing the new input data, i.e. z_k, recursively. However, both the normalizing term and the prior distribution contain integrals that cannot be always found analytically.

A well-known example of a Bayesian filter with an exact analytical solution is the Kalman filter (KF). The objective is to impose simplifying constraints on the system dynamics to be linear-Gaussian i.e. It is assumed that the measurement and motion models have a linear form and are contaminated by zero-mean Gaussian noise. The Kalman filter is one of the most popular fusion methods due to its simplicity of implementation, and optimality in terms of mean-squared error. The Kalman filter is, however, sensitive to data that contains outliers.

Approximation techniques are typically required when dealing with non-linear system dynamics. Two Kalman filter extensions that are suitable to non-linear systems, the Extended KF [3] and Unscented KF [4] are based on first- and second-order Taylor series expansions about the present estimate, respectively. But the non-linearity can only be partially handled by these two approaches. Although grid-based approaches offer an option to approximate non-linear probability density functions, high-dimensional situations make them computationally intractable [5].

The fusion system becomes more complicated when multi-target tracking is used as opposed to single-target tracking [6]. One of these brand-new difficulties is the data association problem, which has two variations: measurement-to-track and track-to-track. The first involves determining the target from which each measurement was taken, whereas the second involves separating and combining tracks that are predicting the state of the same target [7].

Several algorithms have been presented for data association, for instance, nearest neighbor (NN) and K-means. NN is based on the distance metric of measurements from each other. The mentioned distance can be based on an absolute distance, the Euclidean distance, or a statistical function of the distance.

NN has shown poor performance in the case of frequent false measurements [8]. K-means is an alternative approach that divides the dataset into different clusters.

On the other hand, the association can be performed by assigning an association probability to each case from measurement [9]. This method is known as probabilistic data association. In this approach, valid measurements are those observations that fall within the validation gate of the target at that instant. The state estimation of the target is computed by summing the estimated states for all hypotheses. Different measurements can be associated with a particular target. Consequently, the algorithm uses the association probabilities between the different measurements and a specific target as weights to estimate the target state. This approach is mainly useful for tracking targets whose movement patterns do not change abruptly. If it changes its movement patterns abruptly, it will most likely lose the target.

Since local estimates may be dependent due to some reasons, the correlations among local estimates should be taken into account. To deal with this issue, different approaches have been proposed depending on the availability of cross-correlation:

In known cross-correlation cases, the Bar-Shalom Campo (BC) formula [10] can incorporate two measurement data. For more than two measurements, generalized approaches were proposed in [11] and [12]. The multi-scale theory was used to propose a method of fusion for discrete systems with multiple rates [13].

In unknown cross-correlation cases, it is difficult to estimate the cross-correlation among the data sources, especially with a distributed fusion architecture. To address this problem, there are different approaches. If the reason for the correlation is double counting, data decorrelation is one of the basic methods to avoid correlation before the fusion [14].

Another approach is to model the cross-correlation between local measurements, although the exact model is difficult which gives us the last option. These techniques approximate the intersection of many data sources without taking into account cross-correlation in order to offer a fused solution that is suboptimal but consistent under the premise of bounded cross-correlation. The Covariance Intersection Method (CI) and its variations are examples of the method [15] and [16], the Largest Ellipsoid Method (LE) [17], Internal Ellipsoidal Approximation (IEA) [18], and Ellipsoidal Intersection Method (EI).

CI techniques reduce the upper bound of the fusion estimation error variance for the following benefits. Cross-covariance identification and computation are fully avoided. It produces a fused

estimate that is consistent, making it a no-divergent filter. It provides a typical upper bound on the actual estimation error variances and is robust to unidentified correlations. There are three main types of distributed CI fusion structures [19]. The best accuracy is achieved via batch processing, which is computationally expensive due to the high-dimensional nonlinear optimization problem. The two-sensor CI fusion approach is iterated in series processing. First, the two-sensor CI fusion technique fuses any two local estimators. The third local estimator is fused with the fused estimator after that, and so on. After all local estimators are fused, the SCI fusion estimator is finally obtained. Consequently, it can be easily solved using numerical techniques. Combining the first two approaches is known as parallel processing.

2.3 Infrastructure-based navigation in robotics

Using multiple sensor fusion has been numerously studied in robotics applications. Signal-based methods such as laser radar [20], infrared light [21], RFID [22], and laser range finder [23] have been implemented in this application.

In [21], the authors presented the Active Badge system, an indoor localization system that uses infrared communication to track the location of personnel. The system consists of small, wearable devices called badges and fixed infrared receivers located throughout the building. The paper described the design and implementation of the system, including the communication protocol and data management system. It also discussed the evaluation of the system's performance in terms of accuracy and scalability. The paper provided a detailed insight into the development of the Active Badge system and its capabilities as an indoor location tracking system, it also serves as a valuable reference for researchers and practitioners interested in the field of indoor location tracking systems.

In [22], the authors introduced the so-called LANDMARC system, an indoor localization sensing system that used active radio frequency identification (RFID) technology. The system used RFID readers and active RFID tags to track the location of personnel and assets within a building. The paper provided the design and implementation of the system, including the communication protocol and data management system. It also discussed the evaluation of the system's performance in terms of accuracy and scalability.

Another work [24] discussed the use of RFID technology to assist in indoor localization and communication for first responders. The proposed system used a combination of RFID tags and readers to provide accurate location information for first responders in emergencies. The system was tested in

a simulated emergency scenario and the results showed that it was able to accurately locate first responders and improve communication between them. The paper also highlighted the potential of the proposed system in improving the efficiency and safety of emergency response operations. Overall, the paper suggests that RFID-assisted localization and communication system can be a valuable tool for first responders to navigate and communicate in indoor environments during emergency situations.

RFID proposed in [24], can be placed on the interested object and the RFID reader that has an antenna does the work of finding objects using RF tags. By computing the triangulation of multiple landmarks in the local detection area, the location of the objects of interest can be determined. However, the detection of each tag is limited to about 5 meters. The mentioned approaches are not suitable for dynamic obstacles as they need to scan static features, based on the discussion in [25].

Laser range finders are widely used because they provide highly accurate and quick measurements. [23] presented the development of an autonomous navigation system for a mobile robot in an orchard environment. The system utilized a 2D laser scanner to detect and avoid obstacles, and a path-planning algorithm to generate a safe and efficient path for the robot. The authors carried out experiments to evaluate the performance of the proposed system in terms of accuracy, speed, and safety. The results showed that the system was able to navigate the orchard environment successfully and avoid obstacles while maintaining a safe distance. Despite their high standards of measurement accuracy and performance, they are costly to implement because their high performance is dependent on high-priced hardware [23].

In [26], the authors presented an extension to the existing phase accordance method for ultrasonic localization of moving nodes. The proposed method improved the accuracy of the localization by considering the phase difference between the received signals at multiple reference nodes. The method was tested with simulations and experiments, and the results showed that it significantly improved the localization accuracy compared to the traditional phase accordance method. The paper concluded that the proposed method is a promising solution for accurate ultrasonic localization of moving nodes. Although the ultrasonic PL [26] provides location awareness, for ultrasonography to be highly accurate and effective, it requires a lot of infrastructure resources.

The RADAR method is based on processing overlapping signal strength information. In modeling signal propagation, real measurements can be used to localize the objects. In terms of accuracy, RADAR was reported to be 2-3 meters, as mentioned in [27].

2.3.1 Visual-based PL in robotics

Using a camera as a sensor node is another approach that has several challenges in addition to benefits like not being disturbed by other sensors, as mentioned in [28]. In this context, there are two general approaches. Robots are often equipped with cameras for navigation. A fixed camera mounted on the infrastructure at a known position is used in the second type of solution to track the subject throughout the building. The first approach is suitable for relative PL, whereas the second one works well in the global coordinate, as expressed in [27].

Using fixed sensors instead of onboard sensors has benefits that have been taken advantage of in the robotics community. Onboard sensors are commonly utilized for location awareness of the robot; however, if the number of robots gets larger, the cost of this approach increases accordingly. It is possible to reduce the cost required for the localization of multi-robot applications by using an external device to determine all the robots' locations and transmit that information to them. For instance, multiple surveillance cameras installed indoors were used to effectively localize the robot [29]. The proposed method is based on localizing the robot and the objects using multiple surveillance cameras mounted on the ceiling. The authors claimed that the proposed method is relatively low-cost and computationally light in terms of calibration. Mounting the cameras in opposition to each other, the proposed approach leads to little occlusion.

Since the positions of the objects are obtained by the external cameras and then sent to the robot, the camera's coordination and the robot's need to be exactly known. In [28], the robot's position estimation was formulated as a simultaneous robot PL and map-building (SLAM) framework. For this purpose, the problem was divided into two separate problems. The first problem was to estimate the cameras; parameters in order to obtain a calibration between the global frame and the camera frames. The second problem was to global PL of the robots by using multiple cameras. In this case, local measurements of each camera were fused to provide an exact estimation of the robot's position. The first problem, camera calibration, which is calculating the camera parameters and metric measurements has been thoroughly studied and is not the focus of this research. On the other hand, fusing multiple camera data to estimate the position of the robots in the global coordinate can be formulated as a data fusion problem; an EKF was used in [28] or a Particle Filter in [30].

In visual-based approaches, some studies use two-camera PL while some use only one camera for this goal. In the latter, the distance is achieved by the ratio of the size between the pre-defined reference

points and the interested object [31]. This technique is based on the fact that if the distance between the camera and the object changes, the size of the object varies in the image. In other words, the number of pixels containing the object changes as the distance increases or decreases. The results show that the pixel counts and the distance have a simple linear relationship, as shown in [31]. In another work [25], Li et al. proposed a single-webcam distance measurement method for the location awareness of the robot. To do this, a known-size rectangle pattern was used as a ground tile. To elaborate, the location of the robot was obtained using two parallel lines intersecting four edge points of the ground tile in the camera frame.

Another approach to single-camera methods was proposed in [27]. The method is based on using four corners of the coverage field of the camera as a reference and then estimating the location of the object by Euclidian interpolation. The presented technique includes four steps detecting the four corners of the view field, producing a grid approximating the coverage area, calculating the intention points, and mapping image pixels to global coordinates using the nearest neighbor search algorithm in addition to the Euclidean interpolation for the points which are not on the intersections.

2.3.2 PL by artificial markers

The cost and complexity of the vision-based PL methods can be greatly reduced by using simple tags. Therefore, using tags can decrease the processing time of the algorithms. Using radio frequency identification is one of the non-vision examples of this approach [32]. This approach, using simple tags as landmarks, can be applied in vision-based PL as well [33]. The authors aimed to develop a precise indoor localization method for multiple mobile robots by using adaptive sensor fusion of odometry and vision data. The authors proposed a method that utilizes both odometry data from the robot's wheels and visual data from a camera to determine the robot's position. They also proposed an adaptive sensor fusion algorithm that adjusts the weighting of the odometry and visual data in real-time based on the quality of the data. The proposed method was tested and evaluated using multiple mobile robots and the results showed that it was able to achieve high precision and robustness in the localization of the robots.

[34] aimed to develop an efficient indoor navigation technique for blind individuals by using QR codes. The authors proposed a method that uses QR codes placed in the environment to guide blind individuals along the most efficient path. They also proposed a routing algorithm that takes into account the individual's current location and destination, as well as the location of the QR codes, to determine

the optimal route. The proposed method was evaluated using a prototype system and the results showed that it was able to accurately guide blind individuals along the optimal route.

The simple tags can have a non-color feature like Quick Response (QR) codes, as proposed in [34], and also bar codes. For instance, [35] presented a real-time vision-based pattern-tracking method that does not require predefined colors. The authors proposed a method that uses a combination of edge detection, feature extraction, and a correlation-based matching algorithm to track patterns in real-time, without relying on predefined colors. They also proposed a method for determining the optimal feature extraction parameters based on the characteristics of the pattern to be tracked. The proposed method was evaluated using a real-time tracking system and the results showed that it was able to accurately track patterns in real-time, even when the pattern's color was not predefined.

The proposed method in [36] took advantage of fusing QR-based positioning and dead reckoning data. It proposed a method for indoor localization of mobile robots using QR code detection and dead reckoning data fusion. The authors developed a system that employed QR codes placed in the environment and odometry data from the robot's wheels to determine the robot's position. They also developed a data fusion algorithm that fused the information from the QR codes and odometry data to enhance the precision of the localization. The proposed method was evaluated using a mobile robot and the results displayed that it was able to achieve high precision and robustness in the localization of the robot.

Two-dimensional QR codes were utilized to localize the robots in the warehouse [37]. The authors proposed a method for robot navigation in a warehouse using RFID and QR codes. The authors developed a system that utilizes RFID tags and QR codes placed in the warehouse to guide the robot's navigation. They also presented an algorithm that combines the information from the RFID tags and QR codes to determine the robot's position and plan the optimal path. The proposed method was evaluated using a robot in a warehouse environment and the results showed that it was able to accurately guide the robot along the optimal path. In addition, RFID passive tags which was an omnidirectional clue was used since the fast movement of the robot makes the image blurry leading to less accurate QR data. Similar works have been done by Amazon in this manner.

Another work [38] implemented a binary tag named Cantag for their algorithm. Using color patterns in the tag can increase the possible number of identifiable robots. The mentioned work described Cantag, an open-source software toolkit for designing and deploying marker-based vision systems. The

authors presented the software which allows users to create and print custom markers that can be used for tasks such as object tracking, augmented reality, and robot localization. They also described the features of Cantag, including its ability to handle different types of markers, its compatibility with various cameras and platforms, and its easy-to-use interface. The proposed software was evaluated using various tasks, and the results showed that it was able to accurately track markers, even under challenging conditions.

This method was shown to be very cost-effective and robust in terms of PL, as described in [39]. A good example of using simple landmarks is to use sets of color tags on the robots [40]. The authors introduced a method for vision-based localization of multiple mobile robots using a low-cost vision sensor. The authors proposed a system that utilizes visual data from a low-cost camera mounted on the robot to determine the robot's position. They also proposed an algorithm that processes the visual data and uses it to localize the robot in real-time. The proposed method was evaluated using multiple mobile robots in an indoor environment and the results showed that it was able to accurately localize the robots in real-time using a low-cost vision sensor. The color set was utilized to estimate the position of the center of the tag and also the heading angle of the robot.

2.4 Infrastructure-based navigation in vehicles

PL in autonomous vehicles and robotic applications has some similarities and differences. As mentioned before, data fusion of on-board and off-board sensor data has received more attention recently by the research community dubbed as cooperative PL. In cooperative PL, the results show that employing the V2I communication provides a more accurate estimation than in the V2V case since ROADSIDE Units (RSUs) have fixed positions [41]. A key requirement of intelligent driver assistance systems is low latency, high accuracy, and robust security. Compared to current mobile networks, the fifth-generation (5G) mobile networks are expected to provide increased reliability, reduced latency, and higher throughput, offering a great alternative for vehicle communications.

In [42], integrating data from GPS, and radio-frequency identification were used in addition to V2I communication to improve positioning. It aimed to enhance the precision of GPS-based vehicle positioning for use in intelligent transportation systems. The authors proposed a technique that blended GPS and inertial measurement unit (IMU) data to enhance the accuracy of the vehicle's position and heading. They also suggested utilizing a Kalman filter to merge the GPS and IMU data and rectify errors in the GPS data. The results of the proposed method were compared to traditional GPS-only

positioning and demonstrated significant improvements in accuracy. In summary, the study presented a method for enhancing the accuracy of GPS-based vehicle positioning, which was crucial for the development of intelligent transportation systems.

In another work [43], signal-strength- based intervehicle-distance measurements, vehicle kinematics, and road maps were implemented to estimate the relative positions of vehicles. The authors aimed to develop a method for localizing vehicular nodes using the received-signal-strength indicator (RSSI). The authors proposed a method that utilizes RSSI data from multiple wireless access points (APs) to determine the position of a vehicular node. They also proposed a weighting scheme that takes into account the distance between the vehicular node and the APs, as well as the angle of arrival of the signal, to improve the accuracy of the localization. The proposed method was evaluated using simulation and experimental data, and the results showed that it was able to accurately localize vehicular nodes. The study concluded that the proposed method based on the RSSI data, and weighting scheme, was a reliable and effective way to localize vehicular nodes.

Both V2V and V2I data can simultaneously be used and fused with onboard sensor data, as proposed in [44]. The authors proposed a method that combines data from multiple sensors, such as GPS, Wi-Fi, and radio frequency identification (RFID), to improve the accuracy of vehicular localization in tunnels. They also proposed a robust data fusion algorithm that is able to handle errors and outliers in the sensor data and a cooperative localization scheme that utilizes data from multiple vehicles to improve the overall localization performance. In this work, data fusion relies on particle filtering. The proposed method was evaluated using simulation and experimental data, and the results showed that it was able to accurately localize vehicles in tunnels even in the presence of errors and outliers.

The proposed method in [45] aimed to localize the ego vehicle moving in and out of the infrastructure unit’s communication range using particle filters. The authors proposed a method that utilizes data from surrounding vehicles and infrastructure to correct errors in the GPS data and improve the overall accuracy of the car's location. The proposed method was evaluated using simulation and experimental data, and the results showed that it was able to significantly improve the accuracy of car GPS when compared to traditional GPS-only methods. The result showed that car GPS accuracy is improved by using V2V and V2I Communications.

The state of the ego vehicle is generally calculated using Bayesian filtering methods. These methods are rooted in modeling the noise and vehicle dynamics using linear/non-linear or Gaussian/non-

Gaussian theories for non-stationary systems such as a vehicle. There are several filtering techniques that have been utilized in literature such as Kalman Filter (KF), Extended Kalman Filter (EKF), Particle Filter (PF), and Unscented Kalman Filter (UKF) [46].

In the literature review, it is common to see that Multi-agent/sensor autonomous networks often employ the Kalman filter (KF) for their estimation capabilities. However, the KF uses a time-driven estimation methodology and requires synchronous sensor measurements, which results in a high communication latency. As reducing the communication overhead (data transfer rate) of agents in distributed autonomous systems is important from both a practical and theoretical perspective, there has been a recent increase in interest in developing event-triggered (ET) transmission, scheduling, and estimation algorithms [47].

Table 2.1 shows the summary of the works done in the PL of the vehicle using infrastructure connectivity.

Table 2.1: Summary of infrastructure–based PL works in-vehicle application

References	Data fusion	Sensors
[48]	Distributed interactive multiple-model Kalman filter	GPS, signal strength indicators (SSI)
[49]	Kalman filter	Dedicated short-range communications (DSRC)
[50]	Extended Kalman filter	GPS, ad-hoc
[44]	Particle filter	IR-UWB V2X Ranges, RSSI, GPS
[51]	Kalman filter	inertial navigation system (INS), RSS
[52]	Extended Kalman filter and particle filter	Time of arrival (TOA), IMU
[53]	Kalman filter and particle filter	GPS, RSS
[45]	Particle filter	GPS, RSS

2.5 Multi-sensor Fusion in Cooperative PL

In automated driving applications, data fusion of vehicle onboard sensors and infrastructure sensors has received more attention recently by the research community. A comprehensive review of sensors that are used in vehicle PL is given in [54]. Authors in [55] developed a collision-time prediction algorithm that uses fixed transportation ubiquitous sensor networks to measure vehicles' speeds and locations and then send the collision warning to the vehicles through V2I.

A cooperative 3D object detection was proposed in [56] that uses fixed infrastructure sensors. It was reported that the early fusion scheme was found to be more effective than the late fusion scheme, resulting in an improvement of up to 20% in detection performance in the T-junction scenario and 18% in the roundabout scenario. However, better performance comes with a higher communication cost. This is because the early fusion scheme requires more data to be transmitted, specifically the raw point clouds, compared to the late fusion scheme which only requires the transmission of the estimated

objects from the sensors to the central system. Although both early and late fusion schemes were evaluated in this work, the authors did not address the localization problem.

Authors in [42], used the integration of GPS data and radio-frequency identification along with V2I communication to improve positioning. Authors in [43], used signal-strength- based intervehicle-distance measurements, vehicle kinematics, and road maps to estimate the relative positions of vehicles. Both V2V and V2I data [57] can simultaneously be used and fused with onboard sensor data, as proposed in [44] and [58]. Authors in [45] proposed a method that aims to localize the ego vehicle moving in and out of the infrastructure unit's communication range using particle filters. The results showed that the vehicle's GPS accuracy can be improved by using V2V and V2I Communications. Authors in [59], employed V2I communications to send infrared signals from the vehicle to the infrastructure unit. Using four receiving modules, the strength of the signals was compared at these four receivers to estimate the position of the vehicle.

Authors in [60], employed measurements of the camera and Lidar for localization. The proposed approach consists of three main stages. The first stage calibrates the relative extrinsic parameters of the sensors. The target was detected and then tracked in the image frame. Also, the depth was tracked through an interacting multiple-model (IMM) estimator which used depth-image segmentations as measurements. Adding depth to the 2D information of the camera, the position of the target was estimated by finding the nearest point of the Lidar projected in the image frame.

The algorithm presented in [61] employed a target-level fusion of camera and radar data for location application in severe weather conditions. For this purpose, the joint data probabilistic data association method was used as the fusion algorithm. The problem of fusing uncalibrated Lidar and camera for depth estimation was addressed in [62]. Estimating the extrinsic parameters of Lidar-stereo, the depth can be estimated by a Deep convolutional neural network architecture. Despite the reported improvement, CNN-based methods require high computational complexity and memory usage which is still an issue for practical localization applications, especially for onboard systems.

The framework in [63] was designed to fuse the onboard data with information from another node (vehicle or infrastructure). The framework includes two layers of local and global filters. Classic Kalman Filter (KF) was implemented for the local filter while for the global filtering, the likelihood maximization method was used.

In cooperative vehicle localization, a method referred to as Implicit Cooperative Positioning (ICP) with Data Association was presented in [64]. In this method, passive features were detected and paired with onboard sensor measurements to improve the GNSS accuracy. The authors proposed a new method for improving vehicle localization in urban traffic scenarios. The system used data from multiple vehicles to improve the accuracy of each individual vehicle's localization. A distributed Bayesian framework for cooperative data association and vehicle localization was developed, utilizing smart vehicles equipped with onboard sensors and communication devices. Two new techniques were also proposed to handle non-Gaussian location statistics and reduce communication and processing overhead. The study found that this method significantly improved the accuracy of vehicle localization in urban traffic scenarios and highlighted the importance of feature selection to avoid association errors. The system was also found to be robust to external factors such as vehicle dynamics and satellite signal degradation. The proposed method was expected to enable new innovative services in the field of automated driving.

In [65], the authors proposed a new method to collect real-time, high-resolution information about unconnected road users to address the gap in data from connected vehicles. The proposed approach utilized Lidar sensors installed on traffic infrastructures to actively sense the status of surrounding traffic participants and broadcast connected-vehicle messages through DSRC roadside units. The system architecture, data processing, and communication procedures were described in the paper, and results from a pilot implementation of the system at an intersection in Reno, Nevada were presented. The authors suggested that this approach could improve traffic safety, mobility, and fuel efficiency by enhancing the connected network for smart cities and accelerating the deployment of connected infrastructure for autonomous vehicles. This research added to the existing literature on the use of infrastructure sensor nodes for autonomous localization and perception, specifically focusing on the use of Lidar-enhanced connected infrastructures for sensing and broadcasting high-resolution traffic information in smart cities.

In [66], the authors presented a fusion system for combining data from multiple sensors in order to improve the accuracy of object detection and tracking in a vehicular environment. The system utilized edge computing, which allowed for real-time processing of sensor data at the source, rather than relying on a centralized system. The authors proposed a multi-stage approach for track-level data association that incorporated both kinematic and attribute information and used an approximate variant of the Covariance Intersection (CI) for track-level fusion. The system was evaluated using a combination of

simulated and real-world data, and it was shown that the proposed approach can improve the accuracy of object detection and tracking compared to using a single sensor. However, the paper had some limitations like not providing real-world deployment and testing data. Additionally, the paper did not provide any discussion on the system's scalability and the computational resources required for large-scale deployment and operation.

In [67], the authors presented a solution for vehicle-infrastructure cooperative localization that utilizes factor graphs to address the challenges of bandwidth limitations, difficulties in data association, overhead costs of coordinate transformations, and scalability. The proposed solution utilizes factor graphs, which is a probabilistic graphical model represented as a bipartite graph, to take into account the connections between vehicles and add constraints based on the distance between vehicles while localizing the vehicle. The solution was shown to be scalable for many vehicles without increasing computational complexity. The results of the simulation using a constant velocity model showed improved performance in terms of RMSE compared to the traditional Kalman filter approach, by incorporating a topology factor, which connects all the nodes in the system and is not present in the Kalman filter. As a limitation, the proposed solution has not been tested in real-world conditions and with more complex models, and the effects of clutter and obscuration have not been considered in the evaluation. Furthermore, with the growing number of vehicles, it has become increasingly difficult to correctly address the data association problem.

In order to tackle this problem, the authors in [68], proposed an approach that incorporated Symmetric Measurement Equations (SME) within a factor graph framework. The authors presented a graph-based approach to cooperative localization in the autonomous vehicle connected to the infrastructure sensors. The method utilized vertices to represent nodes in the network and edges to represent measurements between them. The localization problem was formulated as a nonlinear least squares optimization problem, which was solved using a Gauss-Newton algorithm. This approach was shown to be more robust to measurement noise and network topology changes when compared to other methods. Additionally, symmetric measurement equations were used for a more efficient solution to avoid the data association problem.

Data association, however, can be effectively addressed instead of avoided. In [69], using the probability data association method, the authors presented a joint method for robust localization of vehicles in a transportation system that used infrastructure-based sensors. The method used a factor

graph representation of the system, which allowed for the integration of sensor measurements and vehicle dynamics. The data association problem was tackled by using a probability data association algorithm. The proposed method was shown to be more robust against outliers.

2.6 Point Set Matching

Point set registration consists of assigning correspondences between two sets of points and determining how one set of points can be transformed into the other. A number of factors make the registration of point sets challenging, such as an unknown spatial transformation, noise, and outliers. An effective point set registration algorithm should have the following desirable properties: computing the accurate transformation required to align the point sets with tractable computational complexity; Easily handling high-dimensional point sets; Robustness to degradations caused by noise, outliers, and missing points. A set of outliers is a set of points that have no correspondence in the other point set, while missing points are features that aren't present in an image because of occlusion or improper feature extraction. Several methods aim to find the correspondence between two point sets and the transformation.

Due to its ease of use and low computing complexity, the iterative Closest Point (ICP) algorithm, developed by Besl and McKay [70], is the most widely used technique for rigid point set registration. The least-squares rigid transformation connecting the two point sets is discovered by ICP after iteratively assigning correspondences based on the closest distance criterion. After that, the algorithm updates the correspondences and keeps going until it finds the local minimum.

ICP performance is dependent on the initial guess. Local minimum and initial guess are two main limitations of the ICP method. There have been numerous proposed ICP variations that modify every aspect of the algorithm, including the point selection and matching process and the reduction method [71].

The binary correspondence is ICP's other shortcoming. Numerous probabilistic techniques, such as [72], were developed to get around this restriction. These techniques take into account all possible point combinations' correspondences based on a certain probability. The binary assignment of correspondences in ICP has been generalized to include this soft assignment of correspondences. The Robust Point Matching (RPM) algorithm, developed by Gold et al. [73], and its later variations [74], which combine soft assignment and deterministic annealing for the correspondence, are among these techniques.

The point set is expressed as Gaussian mixture model (GMM) centroids in a number of rigid point set approaches, including [75] and [76]. To match the GMM centroids to the data points, the registration is formulated as a maximum likelihood (ML) estimation problem. With these techniques, GMM centroids are re-parameterized using a set of rigid transformation parameters (translation and rotation). The likelihood function is optimized using the EM algorithm, which has two steps: the E-step for computing probabilities and the M-step for updating the transformation.

The introduction of an additional distribution term to account for outliers and deterministic annealing on the Gaussian size to prevent unfavorable local minima are features of such probabilistic approaches. These probabilistic approaches outperform traditional ICP, particularly when noise and outliers are present.

In [75], [77], Luo and Hancock found the rotation matrix through singular value decomposition (SVD), but they ignored some terms of the objective function, resulting in only an approximate solution. In [78], Myronenko derived the exact closed-form solution (M-step) for the rigid point set registration. A probability-based method, called Coherent Point Drift (CPD), which was presented, aligned the second point set onto the first point by maximizing the likelihood. The problem of aligning two point sets was formulated as a probabilistic estimation problem in which the second point set was formulated as Gaussian mixture model (GMM) centroids. The algorithm made the GMM centroids move coherently as a group to keep the point sets geometrically stable. In the rigid case, the coherence constraint was imposed through the reparameterization of GMM centroid locations in order to derive a closed-form maximum of the EM algorithm in arbitrary dimensions. Using a fast algorithm, the computation complexity was further lowered to linear.

In [79], it was demonstrated that the Expectation-Maximization (EM) technique for GMM, which treats one point set as GMM centroids with equal isotropic covariances and the other point set as data points, is similar to alternating soft-assignment of correspondences and transformation in RPM.

2.7 Pose Estimation

Given a set of n points in the world coordinate and their corresponding 2D projections in the image frame in addition to the intrinsic parameters of the camera, the position (X_c, Y_c, Z_c) and orientation $(\zeta_c, \theta_c, \varphi_c)$ can be estimated. This problem is referred to as Perspective-n-Point (PnP) problem.

2.7.1 Camera-based Pose Estimation

This problem can be addressed through iterative or non-iterative methods. In the iterative category, the method introduced in [80] formalizes the PnP problem as a minimization of the error in 3D space while the reprojection residuals are the values to be minimized in other methods. Similar to other methods, its performance is sensitive to the initial values since it can converge to a local minimum.

The basic subset of the PnP is to use three points ($n = 3$) as reference points. The common aspect of almost all P3P methods is to use the law of cosines to form a system of three equations based on the distances between the 3D points in addition to the position/orientation of the camera. To solve the system of equations, different methods including zero-decomposition and the Sylvester resultant were used in [81] and [82].

A few works aimed not to use the law of cosines. In [83], the camera's pose was directly solved by a geometric approach in which finding the distances of the points was avoided. To elaborate, the aligning transformation was found directly, resulting in not considering the points in the camera frame. Authors in [84] introduced two intermediate frames to represent the points' arrangments. Then, the rays (projection of points onto planes) in different frames were used to calculate the rotation that aligns the rays.

Regardless of the elimination method, they all are time-consuming and lack accuracy [85]. To improve the accuracy and efficiency, the authors in [85] proposed an approach in which the position of the camera was first eliminated. Consequently, the equations include only the orientation of the camera.

Non-uniqueness in the P3P problem is one of the issues addressed in many works. In non-iterative approaches, most of the proposed works tried to first estimate the position of 3D points in the camera frame by finding the depth. Knowing the world coordinates, the position and orientation of the camera can be estimated to align 2D and 3D points. In most works, using large correspondence point sets is needed to reduce the sensitivity to the noise. In [82], the authors consider the triplets of points and form a four-degree polynomial, and then solve the unknown depth by the SVD method. Since the method is required to be applied on all n points, results in a $O(n^5)$ complexity. The ccomplexity of this method and other similar method is caused by using the interpoint distance constraints, resulting in a additional paramters.

In [86], the author aimed not to employ these constraints by eliminating the world to camera rotation/translation. Resulting in directly finding the depth of the points. Therefore, the complicity of

the method is $O(n^2)$. However, using only linear contraints, the proposed algotihm did not show anacceptable performance in the precense of noise.

In contrast to [86], in [87], the authors presented a non-iterative solution to the PnP problem that considered nonlinear constraints. The computational complexity of the proposed method was reported to be $O(n)$, compared to other works reported $O(n^5)$ or even $O(n^8)$. Other feature of this method was the ability of handling both planar and non-planar configuartions. The main idead behind the introduced method was to formulate the points as a weighted summation of fur virtual control points. As a result, the PnP problem was reduced to estimate the four control points in the camera frame.

2.8 Summary and research gap

There are different types of localization methods that take advantage of V2I communication. The majority of works in this context are vision-based or signal-based techniques using radar, ultrasonic, and Lidar sensors. Signal-based methods are reported to have better accuracy and performance. On the other hand, vision-based techniques are more reliable in some cases when signal-based methods have limited performance due to signal interference. Therefore, in order to take advantage of both approaches, cameras can be cooperatively used with Lidars in the infrastructure sensor nodes.

Although both types of vision-based and signal-based PL (mounted on the robot or mounted on the infrastructure) have been used for mobile robots, no works have been reported using infrastructure-based cameras and Lidars to improve vehicle PL in autonomous fleet operation and urban automated driving. Incorporating the measurements of the ISN, including cameras and Lidars, into the system could be a solution to achieve more accurate, more reliable localization, which is the research focus of this work. In this work, a fusion framework is proposed for vehicle PL that cooperatively uses ISN with onboard sensor data. In this framework, the information of camera-based PL, Lidar-based PL, vehicle motion model, and kinematic constraints are fused. Besides, the range-dependent nature of the sensors is incorporated into the system.

The previous works mostly focused on onboard camera-Lidar fusion. The onboard approach is computationally more complicated compared to the proposed infrastructure-based approach. The reason is that using fixed sensors avoids dealing with the complexity of time-variant orientation/translation matrices that existed in onboard systems.

A few works aimed to address the fixed infrastructure-based camera-Lidar fusion. In the most similar work [63] to this thesis, the method implemented likelihood maximization as the global filter to fuse onboard data with other sensor nodes. Comparing the proposed method in this thesis with [63], kinematic constraints are added in addition to the motion model to fuse the camera and Lidar data. Besides, range-dependent covariance scaling is implemented to incorporate the range-dependent accuracy of the ISNs into the system. This allows the proposed method to rely more on the more accurate source of information at each time step, resulting in a balanced PL performance in the overlapped field of view of ISNs.

Chapter 3
Infrastructure-based Sensor node

3.1 Introduction

In this chapter, the overall structure of the ISN node and its components are described. Each sensor node has a camera and Lidar with the same field of view as can be seen in Figure 3.1. Since the camera and the Lidar are mounted higher than the ground level, the ISN node provides a different FOV that can potentially improve PL. Both sensors are used to give the position of any object in their FOV. The sensors' data is fused at the node by incorporating the vehicle motion model and also kinematic constraints. Mathematical models of vehicle motion and sensors, and the fusion/estimation algorithm are also presented in this chapter.

The significant parameters affecting the accuracy of the camera and Lidar of the sensor node are studied. In this matter, one of the important parameters is the distance between the object and the sensors. Therefore, the fusion of the camera and Lidar data on each node based on their range accuracy is presented. The covariance scaling method is used to let the fusion engine rely on the camera and Lidar based on their range accuracy.

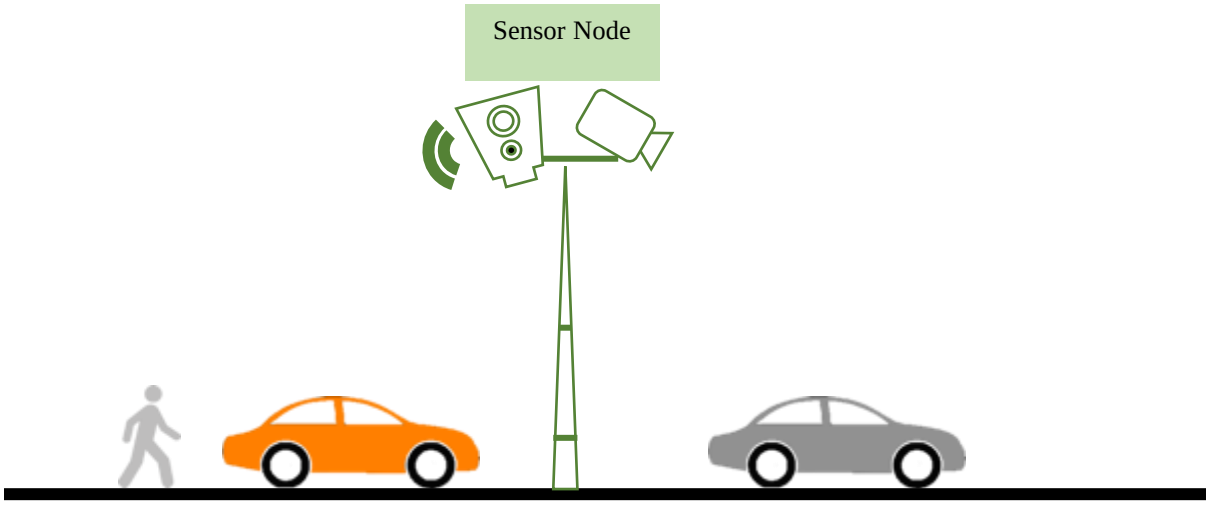


Figure 3.1: ISN node holds one camera and one Lidar

(a) First ISN (b) Second ISN

Figure 3.2. The overlapping covered area of two ISN cameras, simulated in MATLAB/Simulink

In Figure 3.2, two cameras have a FOV of $[80, 70]°$ are mounted so that they have a consistent total FOV of $[160, 70]°$. The Lidar has a FOV of $[360, 31]°$. Although the minimum FOV (common between camera and Lidar) is assumed to be the FOV of the node. Two ISNs are mounted along the road facing each other at a distance of almost 40 m. In this setup, when the vehicle is closer to the first ISN, the first ISN is more accurate to track and localize the vehicle (Figure 3.2.a). As the vehicle moves away from the first ISN, the accuracy of the first ISN decreases. However, on the other hand, the second ISN can detect the vehicle since the vehicle gets closer to it (Figure 3.2.b).

3.2 Overview of the sensor fusion at node

The main algorithm of the ISN is the fusion algorithm that fuses the camera and Lidar data to provide the estimated position and heading angle of all objects in the node's field of view. A Constrained Kalman Filter (CKF) is implemented as the fusion algorithm at the ISN. This method uses the motion model and the kinematic constraints in addition to the camera and Lidar data. Figure 3.3 illustrates the overview of the ISN node data fusion structure. This section is illustrated as the block called "fusion at the node" in Figure 1.2.

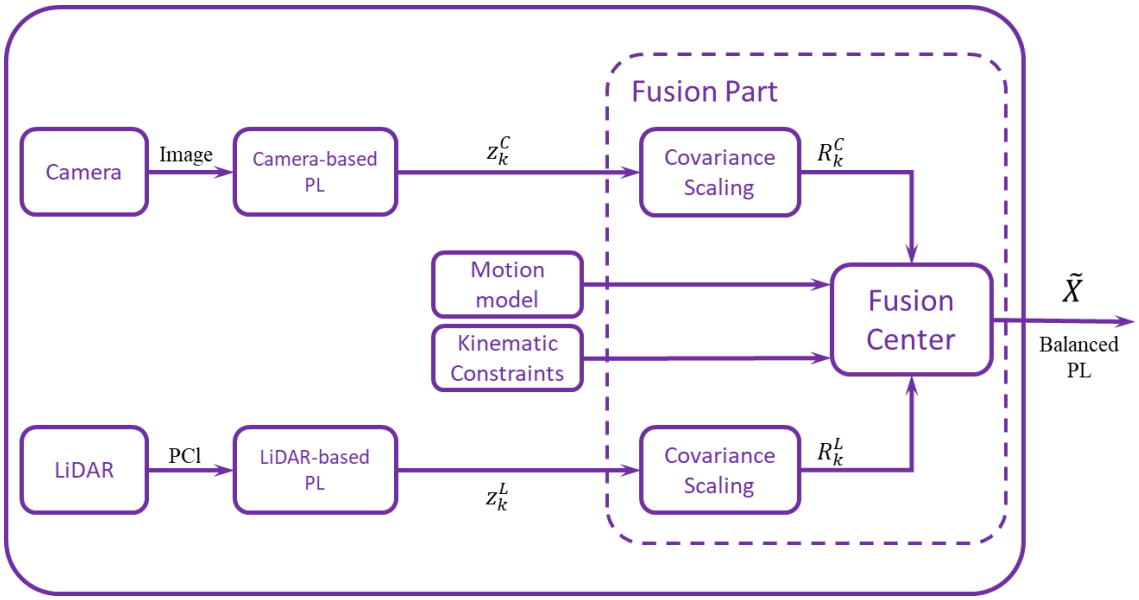


Figure 3.3: Structure of the ISN node

For the CKF algorithm, the dynamic model of the system is considered in the following form

$$X_k = FX_k + Gu_k + \omega_k \tag{3.1}$$

where X_k and u_k, respectively depict the state vector and the input vector at time step k. The state vector and the model matrices F and G will be thoroughly described in the next section which is the dynamic motion model.

The measurement model of the ith ISN is expressed as follows:

$$Z_k^{si} = H^{si}X_k + v_k^{si}, \quad Z_k^{si} \epsilon \mathbb{R}^p \tag{3.2}$$

where H^{si} represents the observation matrix of the ith ISN. Furthermore, It is assumed that the process and the measurement noises $\omega_k \sim \mathcal{N}(0, Q_k)$ and $v_k^{si} \sim \mathcal{N}(0, R_k^{si})$ are zero mean Gaussian white noise where Q_k and R_k^{si} are the process and measurement covariance of the ISN i respectively. Additionally, Z_k^{si} represent the measurements stacking the results of Lidar-based and camera-based positioning, respectfully z_k^L and z_k^C. Consequently, R_k^{si} consists of R_k^C and R_k^L which are the measurement covariance matrices regarding the camera-based and Lidar-based positioning. The details of how they are calculated through the covariance scaling will be covered in Section 3.4.

The estimation results are then broadcast based on the application. In the autonomous fleet operation, the output of each ISN node is used in a cloud-based PL, while in the urban driving application, each AV receives the broadcast information to fuse it with their onboard sensor data which will be explained later.

Although it is not the focus of this thesis, it is necessary to explain the camera-based and Lidar-based PL, since they are the inputs of the fusion algorithm. For the camera-based PL, the boundary box of the vehicle is extracted from the image. The boundary box is then projected onto the top-down view. The center of the projected box is assumed to be the location of the vehicle. For the Lidar-based PL, the raw point clouds are clustered for each vehicle. The clustered points are then used for a cuboid fitting. The center of the fitted cube is assumed to be the center of the vehicle.

The output of the camera-based and the Lidar-based PL is the position and the heading angle of the vehicle. The outputs are used in the covariance scaling and the fusion algorithm. In the fusion algorithm, which is KF, the vehicle motion model, and kinematic constraints are incorporated into the system.

3.3 Vehicle motion model

In addition to the camera and Lidar data, another input of the fusion at each node is the vehicle motion model. By doing this, the PL algorithm can take the dynamic motion of the vehicle into account This section includes dynamic models used for vehicle motion.

3.3.1 Constant velocity (considering heading angle)

While constant velocity models are not an accurate representation of the dynamic motion of a vehicle, they can be a reasonable approximation for certain scenarios, such as urban driving and intersections. In these scenarios, the maximum speed of the vehicle is typically limited to less than 40 km/h, and the vehicle's motion is often constrained by traffic rules and regulations, such as stop signs and traffic lights. As a result, the vehicle's acceleration and deceleration are typically low, and the vehicle's motion can be reasonably approximated as a constant velocity.

Furthermore, using a constant velocity model can help simplify the computational requirements and reduce the complexity of the perception and localization algorithms. This can be particularly important for real-time applications, where computational resources and processing time are limited.

The constant velocity model is suited for straight maneuvers. However, the heading angle of the vehicle is needed to be estimated as well since the vehicle's heading angle changes in most of the maneuvers including double lane change and turning. Therefore, the heading angle is required to be considered in the state vector. As a result, in the dynamic model of the system (3.1):

$$F = I^{3*3} \tag{3.3}$$

$$G_k = \begin{bmatrix} \Delta t. \cos \psi_{k-1}^0 & 0 \\ \Delta t. \sin \psi_{k-1}^0 & 0 \\ 0 & \Delta t \end{bmatrix} \tag{3.4}$$

In this model, the state vector is $X_k = [X_k^0 \quad Y_k^0 \quad \psi_k^0]^T$ which includes the global X position, Y position, and the heading angle of the ego vehicle (referred to as the subscript of *0*) at time step k and Δt is the sampling time.

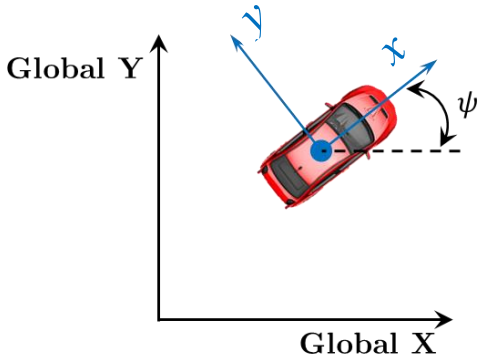


Figure 3.4: Illustration of the vehicle and the states of the model

Figure 3.4. shows the position and heading angle of the vehicle as well as the relative *x* and *y* positions in the vehicle's body coordinates. Besides, the inputs of the system are assumed to be the longitudinal velocity and yaw rate at the previous time step, as considered in $u_k = [V_{k-1}^0 \quad \dot{\psi}_{k-1}^0]^T$.

3.3.2 Combined longitudinal and lateral model

Two kinematic models are used to model longitudinal and lateral motion. For the latter, a kinematic bicycle model is illustrated in Figure 3.5. This model has some assumptions including approximately zero sideslip angle, small curvature path, and negligible lateral velocity compared to the longitudinal

velocity. In this model, lateral velocity is estimated using the longitudinal motion model of the previous states. By these assumptions, the nonlinear equations of motion of the vehicle are expressed as follows:

$$\dot{Y}^0 = V_{k-1}^0 \sin \psi_k^0 \tag{3.5}$$

$$\dot{\psi}^0 = \frac{V_{k-1}^0}{L} tan\, \delta \tag{3.6}$$

where L and δ respectively denote the wheelbase, and steering angle.

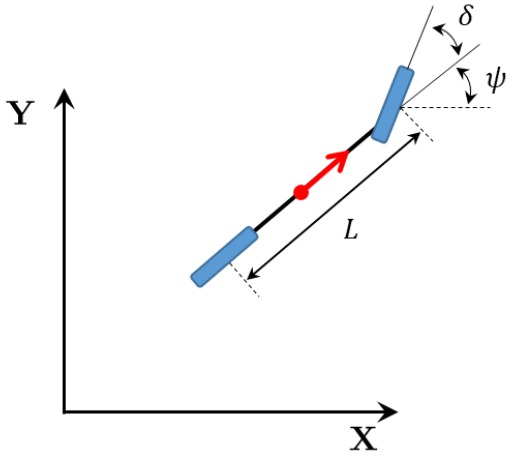


Figure 3.5: Kinematic bicycle model

To linearize this nonlinear model, it is assumed that the heading angle from the previous time step is used. Besides, the steering angle remains small during the maneuvers, meaning $\tan \delta \approx \delta$. As a result, the linear model is expressed as

$$\dot{Y}^0 = V_{k-1}^0 \sin \psi_{k-1}^0 \tag{3.7}$$

$$\dot{\psi}^0 = \frac{V_{k-1}^0}{L} \delta \tag{3.8}$$

The state-space of this model is represented as

$$\dot{X}_{la} = F_{la} x_{la} + G_{la} \delta \tag{3.9}$$

$$F_{la,k} = \begin{bmatrix} 0 & V_{k-1}^0 \\ 0 & 0 \end{bmatrix}, \qquad G_{la,k} = \begin{bmatrix} 0 \\ \frac{V_{k-1}^0}{L} \end{bmatrix} \tag{3.10}$$

where $X_{la} = [Y_k^0 \quad \psi_k^0]^T$ is the state vector.

Remark 1. *Discretization of the general continuous-time system $\dot{X} = FX + Gu$ with the measurement model of $Z = HX + Tu$ is performed by the zero-order hold (ZOH) method, also known as step-invariance [92]. One of the main advantages is its simplicity and ease of implementation, which allows for straightforward implementation in real-time applications.*

The input of the continuous-time system, $u_k = u(t_k)$, is held for a period of $t_k \leq t < t_{k+1}$ with the sampling time of Δt. As a result, the discrete-time system has respectively the system and state matrices of $F_q = e^{F(t)\Delta t}$ and $G_q = \int_0^{\Delta t} e^{F(t)\tau} G(t) d\tau$. Also, the output matrices are equal to the continuous-time output matrices.

3.3.3 Kinematic constraints

Motion models can predict the motion of the vehicle for each time step. However, there is still more room to improve the model. To elaborate, certain information about the system might be available, but it is not considered in the typical estimation algorithm. For instance, the geometry of the road or specific data on the vehicle dynamics can be exploited as an additional source of information to provide better performance. However, only constraints from the previous states of the vehicle are implemented.

In harsh maneuvers or even maneuvers like turning, kinematic constraints on the states may also be incorporated into the system. Taking the previous states into account, the states at the current step can be predicted.

As mentioned before, Constraint Kalman Filter (KCF) is the implemented algorithm to fuse the camera and Lidar data with the motion model. Depending on the type of constraints, certain methods can incorporate the constraints into the Kalman Filter formulation and its variations. If the linear constraints are supposed to be imposed on the linear KF, estimation projection, among other methods, has been proposed to implement the hard constraints.

In this context, another approach is to project the unconstrained state onto the constraint surface [88]. For the model in (3.1-3), the constraint can be calculated as:

$$\tilde{X}_k = \arg\min_x \left(X - \hat{X}_k\right)^T T \left(X - \hat{X}_k\right) \tag{3.11}$$

Subject to

$$DX = d \text{ and } \hat{D}X < \hat{d}$$

where X is the states of the system, defined in (3.1) and $\hat{X}_k$ is the estimated state at the time step k. Also, T is called the constraint matrix applying the constraint onto the state $\hat{X}_k$.

Defining $T = (P_k^+)^{-1}$,

$$\hat{X}_k^- = F\tilde{X}_{k-1} \tag{3.12}$$

$$\hat{X}_k = \hat{X}_k^- + K_k\left(Z_k^{si} - H^{si}\hat{X}_k^-\right) \tag{3.13}$$

$$\tilde{X}_k = \hat{X}_k - P_k^+ D^T (DP_k^+ D^T)^{-1}\left(D\hat{X}_k - d\right) \tag{3.14}$$

where Z_k^{si} and H^{si} respectively represent the measurements and the observation matrix of the ith ISN.

On the other hand, the pseudo-measurement method can be utilized to consider soft constraints in the system. In this case, constraints are going to be approximately satisfied. For this purpose, soft constraints are implemented as additional measurements by nonzero noise. The level of measurement covariance corresponding to these constraints can affect the weight of the constraints. Although, they are not real measurements.

For this purpose, the kinematic constraints are implemented into the system through the predicted state. The idea is that the fixed-sized sequence, n_p, of the past estimated states, $\{\tilde{X}\}_{k-n_p+1}^{k-1}$, is used to fit a curve. This fitted curve is in the state space and is considered a second-order polynomial function. Using this fitted curve, the predicted current state X_k^* is calculated based on the $\{\tilde{X}\}_{k-n_p+1}^{k-1}$ which is the sequence of the past estimated states. The predicted state, X_k^*, is then inserted into the estimation formulation to incorporate the constraints of the previous states into the estimation of the current state.

Therefore, the estimation can be formulated as an optimization problem, where the weighting of the constraints can be tuned by the covariance matrix T. As a result, the estimation problem can be written as:

$$\tilde{X}_k = argmin_{\mathrm{X}} \sum_{k=1}^{n} \left\|\mathrm{X} - \tilde{X}_k^-\right\|_{Q_k}^2 + \left\|Z_k^{si} - H^{si}\mathrm{X}\right\|_{R_k^{si}}^2 + \left\|\mathrm{X} - X_k^*\right\|_T^2 \tag{3.15}$$

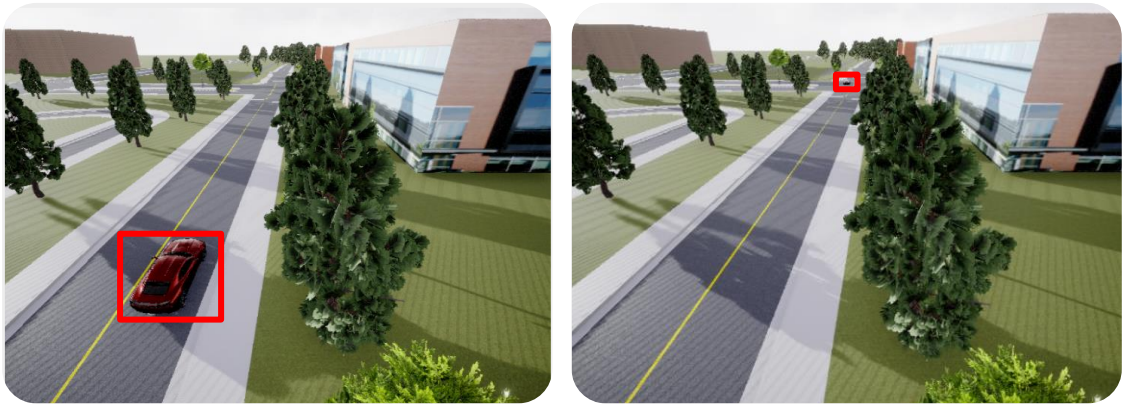
where Q_k, R_k^{si} are respectively the process and measurement covariance of the CKF discussed before and T is the weighting matrix regarding kinematics constraints.

3.4 Sensor uncertainty formulation

Vehicle navigation relies on accurate quantification of sensor noise or uncertainty to produce reliable state estimates. In practice, this uncertainty is often fixed for a given sensor and experiment, whether by automatic calibration or by manual tuning. Although a fixed measure of uncertainty may be reasonable in certain static environments, dynamic scenes frequently exhibit much uncertainty that corrupts a portion of the available observations. For visual sensors, these effects include, for example, distance, velocity, self-similar textures, variations in lighting, and motion blur.

3.4.1 Sensor noise/uncertainty

Sensor noise characterization is needed to be performed to obtain accurate estimates from navigation systems. Standard calibration techniques can provide some insight into sensor noise, but in practice, noise parameters are often manually tuned to optimize estimator performance for a given environment. For visual systems, a large body of literature has also focused on compensating for specific effects such as moving objects, lighting variations, and motion blur.



(a) Vehicle near the ISN (b) Vehicle far from the ISN

Figure 3.6: Bounding Box of the vehicle in different distances

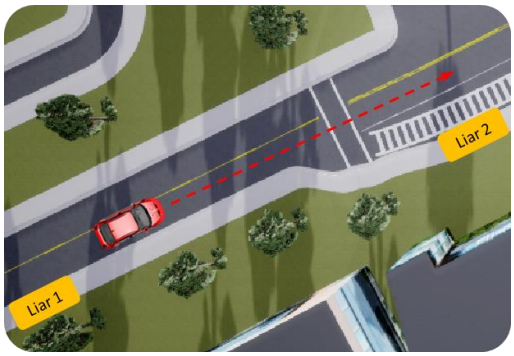
3.4.2 Correlation between sensor uncertainty and distance

For camera-based PL of an object, the theoretical analysis, as presented in [31], proves that the tracking performance depends on the containing pixels of the object that change with the distance of the object from the sensor. Figure 3.6 illustrates the camera images, simulated in MATLAB/Simulink. It shows

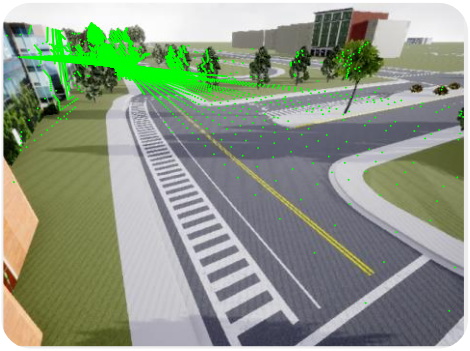
that the number of pixels containing the object is a lot larger when the vehicle is near the ISN (Figure 3.6.a) than when the object moves away from the camera (Figure 3.6.b).

For the Lidar-based PL of an object, the accuracy can be assumed to be dependent on the number of Lidar points falling onto the object. In a scenario, two Lidars are located along the road, as can be seen in A similar trend can be found in Lidar where the density of Lidar points decreases as the distance gets larger, as can be seen in Figure 3.7.a. For Lidar sensors, the accuracy can be assumed to be dependent on the number of Lidar points located on the objects. The density of the Lidar points goes down as the distance from the Lidar increases. The density of Lidar 1 can be seen in Figure 3.7.b. The density of points of Lidar 1 decreases as the range increases.

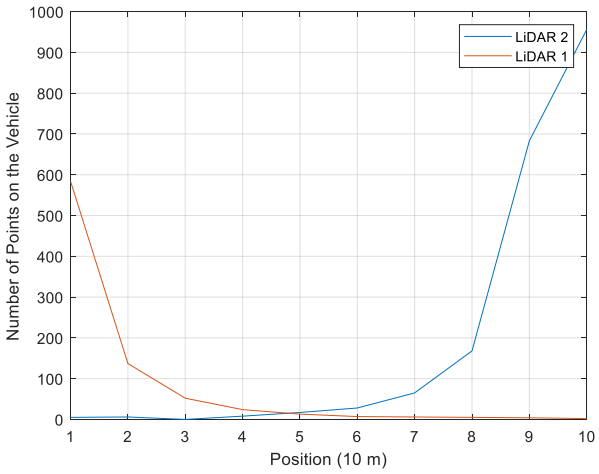
By clustering the Lidar points, the Lidar point falling onto the vehicle can be found. When the vehicle is near Lidar 1, the number of points from Lidar 1 that falls onto the vehicle is around 600. This number decreases to zero when the vehicle moves away. In large distances, where the number of Lidar points on the vehicle is low, there is a need to have another sensor node. This is how balanced Lidar points can be achieved between two sensor nodes, as can be seen in Figure 3.7.c.



(a) The location of the Lidars in the scenario



(b) The density of the Lidar 1 points



(c) Number of Lidar points falling onto the vehicle during the scenario

Figure 3.7: Location, and the density of Lidar 1 and 2

The theoretical analysis, as presented in [31], proves that the tracking performance of the camera depends on the pixel. Knowing that the containing pixels of the object change with the distance of the object from the sensor. The simulation results, which were presented before, are consistent with the fact that the positioning accuracy of the ISN has a distance-dependent nature.

Since the distance affects the number of pixels and Lidar falling points in both dimensions of the image frame, the measurement noise of the infrastructure-based sensors can be assumed to have a quadratic relationship with the distance of the object from the ISN. Therefore, v_k^{si} which is the element of the measurement noise covariance, R_k^{si}, can be described as:

$$v_k^{si} = a_n {d_{i,k}}^2 \quad (3.16)$$

where a_n is a constant parameter and the $d_{i,k}$ is the relative distance between the object and the sensor. It is noted that the relative distance is calculated in discrete time based on the estimated position of the object and the position of the ISN is fixed at a known location.

3.5 Wind-induced Motion Compensation

When the ISNs are first installed, the initial calibrated intrinsic and extrinsic parameters can be achieved by different methods. However, after the initial calibration, the ISNs are exposed to external disturbances like vibration resulting in time-varying extrinsic parameters [89]. For instance, if ISNs are

mounted high above the ground level on light posts, they could easily oscillate due to wind. Consequently, the extrinsic parameters (rotation matrix r_k^{si} and translation vector t_k^{si}) of the sensor node i at time k differ from the initial calibrated values (r_0^{si} and t_0^{si}). This deviation from the initial calibrated extrinsic parameters causes an error in the localization.

While this technique can be effective in compensating for the effect of external disturbances on any sensor motion, it is designed specifically for ISNs. Onboard sensors may be subjected to a variety of external disturbances. However, onboard sensors expose to external disturbances with much higher frequencies. For example, a vehicle may experience jolts or vibrations caused by uneven road surfaces or sudden acceleration or deceleration. Additionally, the motion model of a vehicle is different from that of a light post, with more complex and variable motion patterns.

The goal of this section is to develop a wind-induced motion compensation module to compensate for the effect of the motion of ISNs due to wind or other external disturbances. This module was mentioned in the overall block diagram of the cooperative PL, shown in Figure 1.2 as the "motion compensation" module. To elaborate, the extrinsic parameters of the sensors are estimated while the sensor nodes are moving. For this purpose, camera and Lidar data from the scene in addition to the motion model of the light post, where ISNs are mounted, are incorporated into the wind-induced motion compensation module.

Given a set of n points in the world coordinate and their corresponding 2D projections in the image frame in addition to the intrinsic parameters of the camera, the position (x^c, y^c, z^c) and orientation $(\zeta^c, \theta^c, \varphi^c)$ can be estimated. This problem is referred to as Perspective-n-Point (PnP) problem.

3.5.1 Camera-based Pose Estimation

This problem can be addressed through iterative or non-iterative methods. The basic subset of the PnP is to use three points $(n = 3)$ as reference points. A set of reference points can be extracted based on the feature of the scene, like the edges of buildings. An instance of the reference points extracted in the camera frame can be seen in Figure 3.8.

Figure 3.8: Example of reference points in the camera frame

The position of the reference points in world coordinates $P_i = [X_i, Y_i, Z_i]^T$ can be achieved through High Definition (HD) map. When the camera, along with the sensor node, oscillates due to wind, the same reference points can be captured in the new so-called disturbed image frame. Since the positions of the reference points in the world coordinates are available, a PnP algorithm can be implemented to estimate the rotation matrix $\hat{r}_k$ and translation vector $\hat{t}_k$ of the camera, as can be depicted in Figure 3.9.

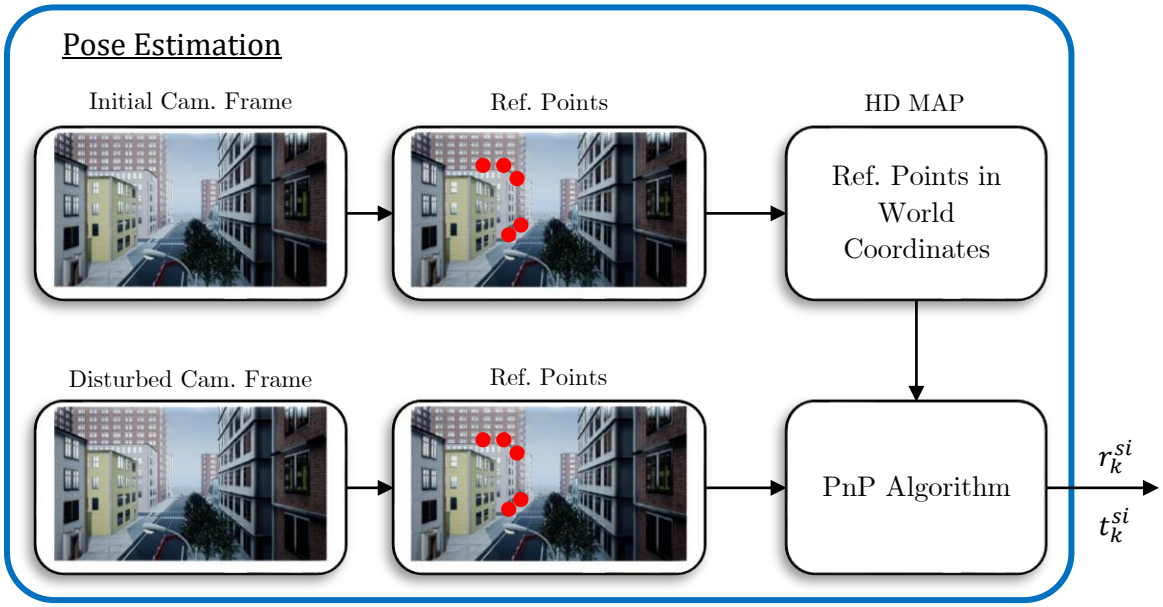


Figure 3.9: General framework of camera-based pose estimation

3.5.2 Lidar-based Pose Estimation

Lidar and camera sensors are fixed inside the sensor node module. Thus, when the sensor node oscillates, Lidar and camera sensors move together and their relative configuration does not change. As a result, the position and orientation of the sensor node can be estimated using the Lidar point cloud as well.

Figure 3.10 illustrates the framework of the Lidar-based pose estimation. The inputs of the framework include two Lidar Point Clouds (PCls). The initial Lidar PCl is achieved from the initial calibrated sensor node. The disturbed Lidar points are provided when the sensor node deviates from its initial calibrated extrinsic parameters due to wind. In these two Lidar PCls, certain landmarks from the scene, especially the building, can be used as a reference for comparing the disturbed Lidar PCl with the initial one. To elaborate, Lidar points that fall on a specific building can be filtered and treated as two sets of points. A point-set matching algorithm like the ICP algorithm uses two mentioned sets of points to estimate the relative configuration between the initial calibrated sensor node and the deviated one. This is the estimated position and orientation of the sensor node that deviated from the initial calibrated values.

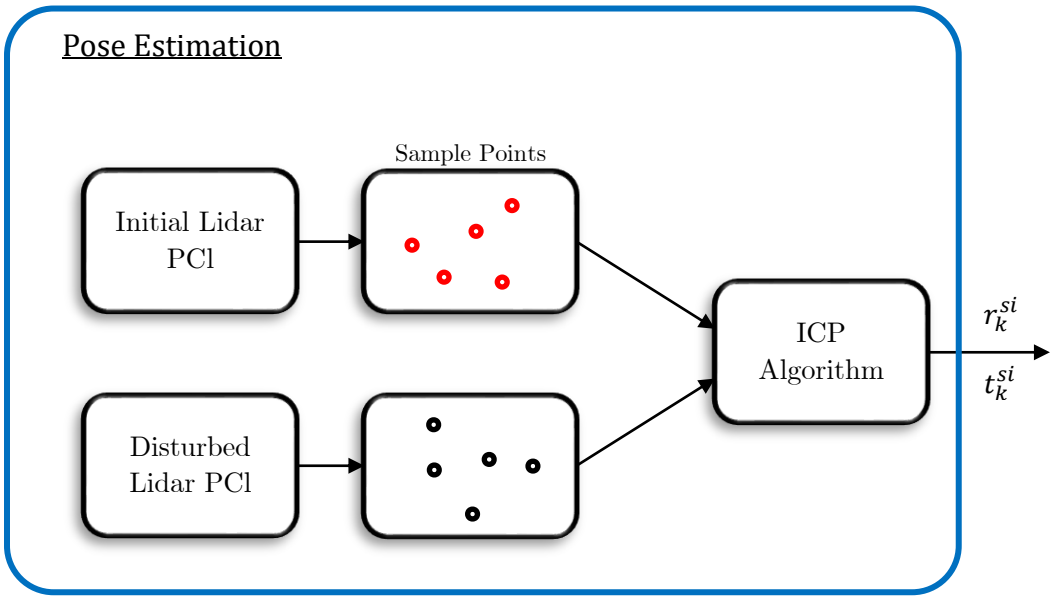


Figure 3.10: General framework of Lidar-based pose estimation

3.5.3 Pose Estimation using Camera/Lidar Augmentation

Using world coordinates of reference points, provided by Lidar, in addition to pixel coordinates of reference points can use both sensor data to estimate the extrinsic parameters of the sensor node. Figure 3.11 shows the framework of the pose estimation using both camera and Lidar data.

As mentioned before, the PnP algorithm requires a set of n points, their world coordinate, and also their corresponding 2D projections in the image frame. The PnP algorithm used Lidar data to have the position of the reference points in world coordinates. Furthermore, 2D projection of the same reference points in the disturbed frame is achieved from the camera. Given the reference points in world coordinates (from Lidar) and their 2D projection (in camera), the PnP algorithm can estimate the extrinsic parameter of the sensor node which is r_k^{si} and t_k^{si}.

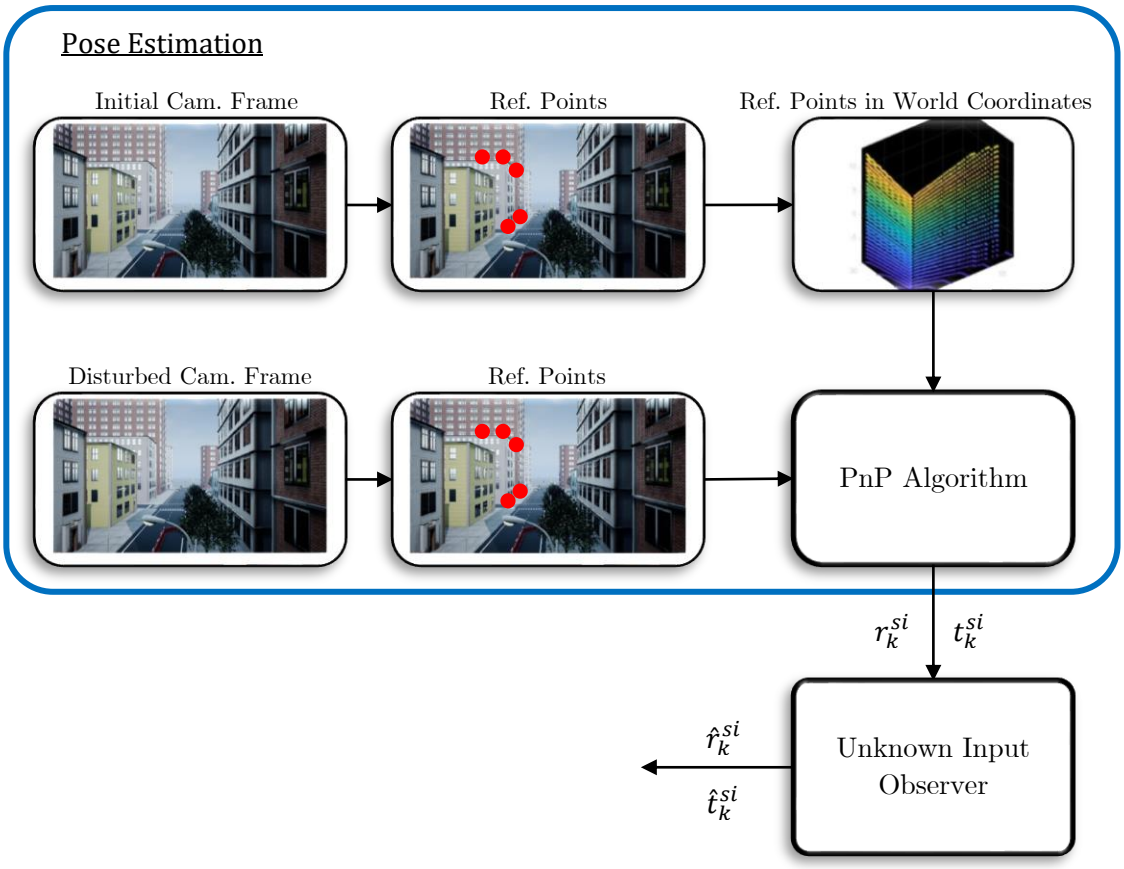


Figure 3.11: General framework of pose estimation using camera/Lidar data

The output of the pose estimation which is the estimated extrinsic parameters of the sensor node is then implemented as the measurements of the developed Unknown Input Observer (UIO). The dynamic motion model of the light post is incorporated into the UIO to improve the performance of the motion compensation in situations in which sensors lack accuracy.

3.6 Dynamic Motion Model of Light Post

ISNs are mounted at a high level from the ground on the light post. External load mostly due to wind makes the light post oscillate. In this context, the light post can be modeled as a vertical cantilever beam exposed to an external load. The proposed model of the light post consists of two degrees of freedom in X and Y directions (x^{si}, y^{si}) with corresponding pitch and roll angles $(\theta^{si}, \zeta^{si})$ where si refers to the ith sensor node.

To consider the two-DoF motion of the light post, the following state space model with the states $X^{si} = [x^{si} \quad \dot{x}^{si} \quad y^{si} \quad \dot{y}^{si}]^T$ is used:

$$\dot{X}^{si} = \bar{A}X^{si} + \bar{B}u \tag{3.17}$$

where $\bar{A}$ and $\bar{B}$ in eq. (3.18) are respectively 4×4 and 4×2 matrices. Furthermore, u is the input including disturbances in X and Y directions.

The sensor node is assumed to be mounted along the vertical axis of the light post at a height h. Therefore, when the light post oscillates, the ith sensor node moves to the position (x^{si}, y^{si}) with orientation of $(\theta^{si}, \zeta^{si})$. Since the height of the installation of the sensor node, h^{si}, is much larger than x^{si}, it is reasonable to assume that

$$\begin{cases} sin(\theta^{si}) \approx \theta^{si} \approx \dfrac{x^{si}}{h^{si}}, \\ cos(\theta^{si}) \approx 1 \rightarrow \Delta z^{si} \approx 0. \end{cases} \tag{3.19}$$

The same assumption is valid for the Y direction. Therefore, the assumption of decoupled motion model can be applied to the model of (3.17). As a result, $\bar{A}$ and $\bar{B}$ matrices can be decoupled into two DoFs in X and Y directions. Therefore they can be described as:

$$\begin{aligned} \bar{A} &= diag(\bar{A}_x, \bar{A}_y) \\ \bar{B} &= diag(\bar{B}_x, \bar{B}_y) \end{aligned} \tag{3.20}$$

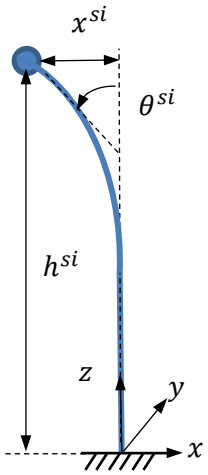


Figure 3.12: Dynamic model of light post as a vertical cantilever beam in the *x*-direction

As mentioned before, the motion model of the light post got decoupled in *X* and *Y* directions. Both DoF has a similar motion model. Therefore, the motion model of each DoF can be developed separately and then inserted into the model (3.18,19). Based on this assumption, the dynamic motion of the light post only in the *X* direction will be presented. Figure 3.12 illustrates the motion model of the light post in the *X* direction.

Based on eq. (3.19), the motion is within the linear region. Thus, the linear dynamic motion of the light post can be derived based on the Euler-Bernoulli theory [90]. For this purpose, the dominant natural frequency of the light post is required. Figure 3.13 illustrates the first two modes of the vertical cantilever beam.

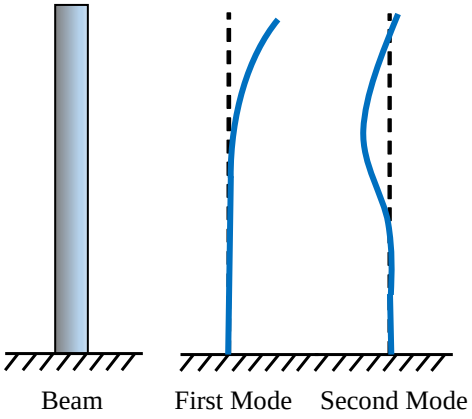


Figure 3.13: First two modes of a vertical cantilever beam

In the present study, the dynamic motion model is formulated using the first mode of oscillation which contributes to a significant part of the overall response of the light post, especially in low-level external excitations. Besides, exciting higher modes of oscillation requires much more energy than what is typically present in regular weather winds. Therefore, the first mode of oscillation is typically the most important mode to consider. Thus, it is reasonable to assume that the first mode of oscillation of the light post is the most significant mode of vibration to consider in the model, and neglecting higher modes will not significantly affect the results.

According to [90], the first natural frequency of the light post is as follows:

$$\omega_{x,1}^{si} = \frac{3.516}{(h^{si})^2}\sqrt{\frac{E^{si}I_x^{si}}{m^{si}}} \tag{3.21}$$

where m^{si} is the total mass of the light post, E^{si} is the elastic modulus and I_x^{si} is the moment of inertia.

To determine the mentioned model parameters of the light post, certain assumptions are made due to the variability of the nominal parameters across different light posts. To elaborate, it is assumed that the light posts have a similar structure and are made of aluminum. More specifically, it is assumed that the light posts can be modeled as hollow cylinders made of aluminum. Using this assumption, the elastic modulus and density of aluminum are respectively considered to be $70\ GPa$ and $2.7\ g/cm^3$.

While the nominal parameters may vary between light posts, this modeling approach can calculate reasonable model parameters that can be used to develop a dynamic motion model for the light posts. By using a consistent modeling approach, it is possible to account for the variability in nominal parameters and develop a more accurate and reliable model for the behavior of the light posts.

Whoever, it is worthwhile to note that these values are nominal and may not precisely reflect the properties of all light poles in practice. There may be some variability in the material properties of different light poles due to manufacturing differences, age, wear and tear, and other factors. As such, there is some degree of uncertainty associated with the model parameters.

To address this uncertainty, it is important to consider a range of values for the model parameters reflecting the expected variability in the properties of different light poles. This can be done by conducting sensitivity analyses or by obtaining data on the material properties of a representative sample of light poles. Another approach is to estimate the parameters.

Using the first frequency, the dynamic motion of light post in the x-direction can be written as:

$$\begin{bmatrix} \dot{x}^{si} \\ \ddot{x}^{si} \end{bmatrix} = \bar{A}_x \begin{bmatrix} x^{si} \\ \dot{x}^{si} \end{bmatrix} + \bar{B}_x u_x \quad (3.22)$$

where x^{si} and $\dot{x}^{si}$ are respectively *x* position and *x*-component of the velocity of the ith sensor node. u_x is the load inserted by the wind in the *X* direction. Besides, $\bar{A}_x$ and $\bar{B}_x$ are defined by:

$$\bar{A}_x = \begin{bmatrix} 0 & 1 \\ -\left(\omega_{x,1}^{si}\right)^2 & -2\xi^{si}\omega_{x,1}^{si} \end{bmatrix} \quad (3.23)$$

$$\bar{B}_x = \begin{bmatrix} 0 \\ \dfrac{1}{m^{si}} \end{bmatrix} \quad (3.24)$$

in which ξ^{si} is the damping ratio. The damping ratio is assumed to be only mass-proportional which gives:

$$\xi^{si} = \frac{\alpha}{2\omega_{x,1}^{si}} \quad (3.25)$$

For the under-damping case, which works for this work, $1 < \alpha < 2\omega_{x,1}^{si}$.

3.7 Unknown Input Observer

In the motion model of the light post, the input (wind load) is not available. Therefore, to estimate the position/orientation of the sensor node, it is required to implement a UIO. UIO decouples state estimation error from the unknown input. Considering an unknown disturbance to the model (3.17), the dynamic motion model can be rewritten as:

$$\begin{cases} \dot{X}^{si} = \bar{A}X^{si} + \bar{B}u \\ Y = \bar{C}X^{si} \end{cases} \quad (3.26)$$

where $X^{si} = [x^{si} \quad \dot{x}^{si} \quad y^{si} \quad \dot{y}^{si}]^T$, $\bar{B}$ is the input matrice related to u, the unknown input vector. Besides, the measurements include position (x, y) with orientation of (θ, ζ). Thus, the observation matrix is:

$$\bar{C} = diag\left(\bar{C}_x, \bar{C}_y\right) \quad (3.27)$$

in which

$$\bar{C}_x = \bar{C}_y = \begin{bmatrix} 1 & 0 \\ \frac{180}{\pi h^{si}} & 0 \end{bmatrix}. \quad (3.28)$$

Adopting the assumption of $\bar{B}$ being full column ranked, a UIO can be designed as follows:

$$\begin{cases} \dot{z} = \bar{F}z + KY \\ \hat{X}^{si} = Z + HY \end{cases} \quad (3.29)$$

where z, the state of the observer, has the same size as X^{si}. The dynamic of the estimation error gives:

$$\begin{aligned} \dot{e} = X^{si} - \hat{X}^{si} &= (\bar{A} - H\bar{C}A - K_1\bar{C})X - K_2\bar{C}X - (\bar{A} - HCA - K_1\bar{C})\hat{X} \\ &+ (\bar{A} - H\bar{C}\bar{A} - K_1\bar{C})\hat{X} - Fz + (\bar{B} - H\bar{C}\bar{B})u \\ &= (\bar{A} - H\bar{C}\bar{A} - K_1\bar{C})e - [F - (\bar{A} - H\bar{C}\bar{A} - K_1\bar{C})]z \\ &- [K_2 - (\bar{A} - H\bar{C}\bar{A} - K_1\bar{C})]y - (H\bar{C} - I)\bar{B}u \end{aligned} \quad (3.30)$$

in which, $K = K_1 + K_2$. By satisfying the following conditions:

$$(HC - I)\bar{B} = 0$$

$$T = I - H\bar{C}$$

$$F = \bar{A} - H\bar{C}\bar{A} - K_1\bar{C}$$

$$K_2 = FH \quad (3.31)$$

$$rank(\bar{C}\bar{B}) = rank(\bar{B})$$

$$(\bar{C}, A_1) \textit{is detectable, where}$$

$$A_1 = (I - H\bar{C})\bar{A}$$

The dynamics of the error is given by:

$$\dot{e} = Fe \quad (3.32)$$

which makes the state estimation error approach zero asymptotically, regardless of the unknown input [91], if F is stable. The discretization is done by the same method, described in Section 3.3.2.

Figure 3.14: Simulated scenario in MATLAB/Simulink Automated Driving.

3.8 Simulation results

In this section, the results of implementing ISN node data in the PL application are presented. The ISN node provides position estimation for the vehicle performing different scenarios. The mentioned scenarios with camera-Lidar sensors are simulated in 3D by the Automated Driving Toolbox of MATLAB/Simulink.

An instance of the simulated scenario is illustrated in Figure 3.14 which shows the ego vehicle performing a constant velocity maneuver. The camera and Lidar sensors are mounted at a height of 10 m from the ground level.

3.8.1 Sensor and Simulation Parameters

The sampling time of the simulation was set to be $T_s = 10\,ms$, however, all sensors including the camera and Lidar were simulated to operate at a 10 Hz frequency. Thus, the time step of the proposed algorithm was assumed to be $\Delta t = 100\,ms$.

There are several factors to consider when choosing a sampling rate for sensors, including vehicle speed, processing requirements, and algorithmic accuracy requirements. This thesis is part of a larger project that integrates modules such as control, path planning, decision-making, perception, and localization. For the overall system to operate efficiently and effectively, a sampling rate of 10 Hz was chosen for each of these modules to meet specific computational requirements. Consequently, a 10 Hz sampling rate was selected for all sensors, including the simulated camera and Lidar sensors.

Additionally, the application of infrastructure-aided perception and localization in this project focuses on urban driving scenarios, such as intersections, residential/commercial complexes, and resorts. Vehicles in these scenarios are limited to a maximum speed of less than 40 kph. With vehicle dynamics, the frequency range of interest depends on vehicle speed and acceleration. Vehicle motion is relatively slow and the frequency at which the vehicle's motion changes is low for urban driving scenarios with speeds less than 40 kph. In urban driving scenarios, the sampling rate of 10 Hz allows accurate perception and localization of the vehicle's dynamic motion, allowing for accurate perception and localization, in addition to effectively operating other modules.

It is assumed that all sensors are operating at the same sampling rate. However, fusing measurements from different sensors with different sampling rates can be handled by KF. In order to do this, the KF first uses the measurements from each sensor to estimate the current state of the system at its own sampling rate. Then, the filter incorporates these estimates into a larger state estimate that is updated at a common sampling rate.

In the simulation, the range and the horizontal resolution of the Lidar were respectively 0.03 m and 0.2 degrees. The focal length of the camera and the image size of the camera were respectively $[1144, 1144]$ and $[1200, 1920]$ in pixels, as should be defined in the Simulink environment. For the onboard sensor, GNSS/INS was used.

Based on the results reported in [93], the position accuracy of $[1, 1, 0.8]$, the attitude (roll, pitch, yaw) accuracy of $[0.014, 0.013, 0.247]^{\circ}$, and the velocity accuracy of $0.03\ m/s$ were set as the parameter of the GNSS/INS. The following results are based on the estimation error which is the difference between the estimated value and the ground truth, as can be seen in Figure 3.15.

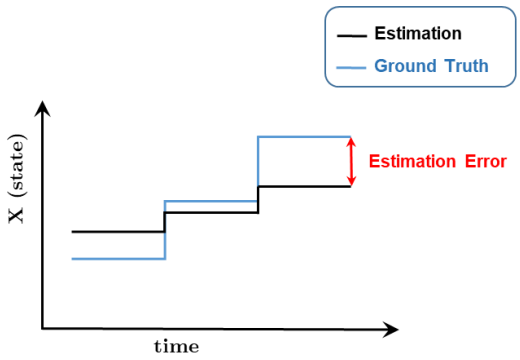
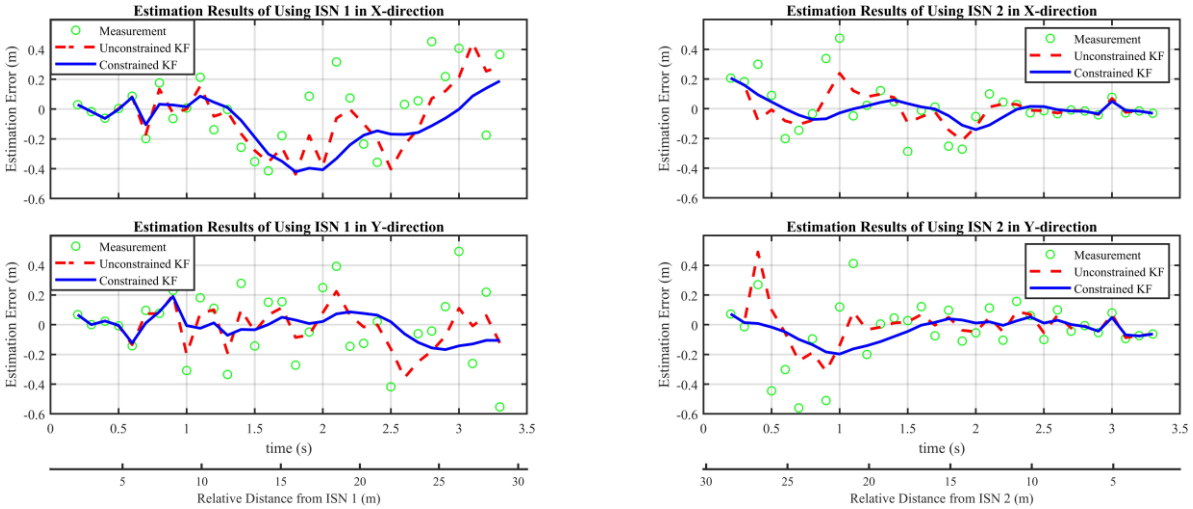


Figure 3.15: The visualization of the estimation error



(a) Estimation Error of ISN 1 (b) Estimation Error of ISN 2

Figure 3.16: Estimation results of (a)ISN 1 and (b) ISN 2 in *X* and *Y* directions for scenario 1.

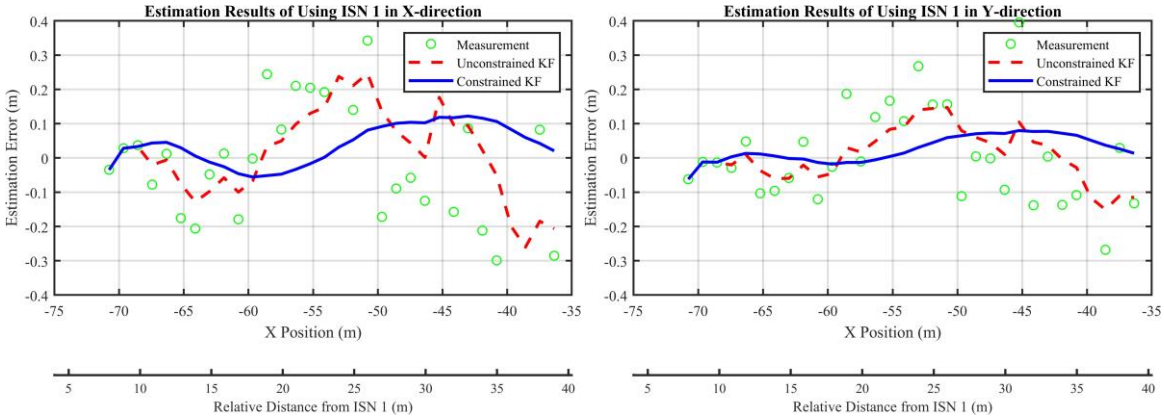
3.8.2 Scenario 1 – Constant Velocity

In this scenario, the ego vehicle is moving at a constant velocity of $v = 10\ [m/s]$ from the initial position of $[-10.2, 17.4]^T\ m$. Two ISNs are located at $[-10, 17, 10]^T$ and $[10, -5, 10]^T\ m$ respectively. Both sensor nodes have a camera and Lidar that can detect the ego vehicle.

Figure 3.16 illustrates the estimation error of ISN 1 and ISN 2 with/without using the motion model and kinematic constraints. Both Lidar-based and camera-based PL in addition to the kinematic constraints and motion model are used as input for the linear least square optimization. Measurements, denoted in green circles, represent the case in which camera-based and Lidar-based localization are implemented without considering the motion model and kinematic constraints.

For this case, the maximum estimation error is around 0.5 m for both the x and y directions. In the unconstrained case (denoted in dashed red line), by fusing the motion model with the measurements, the root mean squared error (RSME) is decreased from 0.33 m to 0.2 m in the x-direction and from 0.29 m to 0.13 cm in y-direction while not having any improvements in the maximum error significantly.

By incorporating the kinematic constraints (denoted in blue), the RMSE slightly reduces from 0.20 m to 0.19 m in the x direction and from 0.13 m to 0.08 m in the y direction. However, the maximum error was significantly decreased from 0.5 m to 0.2 m that is because of imposing more constraints on the position of the vehicle with respect to its previous position.



(a) Estimation Error in *X*-direction (b) Estimation Error in *Y*-direction

Figure 3.17: Estimation result of the ISN 1 for turning scenario

Note that in this scenario, the vehicle starts from a position near ISN 1 and as it keeps moving, it gets closer to ISN 2. As predicted, the estimation error increases as the distance of the object from the sensor node gets larger.

3.8.3 Scenario 2 – Turning Right

In this scenario, the vehicle performs a turning maneuver at a constant speed of $v = 10\ [m/s]$. One ISN broadcasts the estimated position of the vehicle so that the vehicle receives and fuses it with its onboard sensor data, GNSS/INS in this case.

Figure 3.17 shows the error of the developed algorithm in which the constraints are applied compared with the cases in which only sensor measurements or measurements with a motion model are used individually. The red line presents the error when only ISN is used. The results show that the error is dependent on the relative distance of the object from the sensor node.

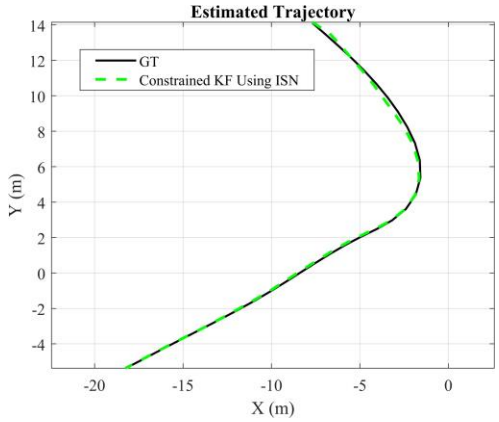


Figure 3.18: Estimated Trajectory of the vehicle performing scenario 2.

The estimated trajectory can be seen in Figure 3.18. By using ISN in addition to the motion model and the kinematic constraints, the proposed algorithm shows an excellent performance in terms of accuracy and consistency.

The overall error statistics of the ego vehicle localization using different sensor nodes can be seen in Figure 3.19. The mean estimation error is acceptable in all cases. However, fusing both sensor nodes decrease the deviation of the estimation at least by 50%. The reason is that each sensor has a relatively smaller estimation error in short ranges. Fusing both sensor nodes allows the algorithm to rely more on the more accurate sensor node at each time resulting in a balanced estimation regardless of the distance of the object from each individual sensor node.

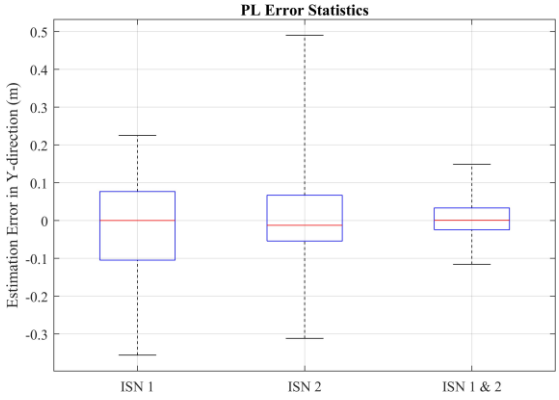


Figure 3.19: Estimation error statistics using different sensor nodes for scenario 2.

3.8.4 Motion Compensation Results

As mentioned before, the sensor node mounted on the light post can oscillate due to wind. This results in a deviation of the extrinsic parameters of the sensors. These oscillations happen in different frequencies depending on the wind speed. In this section, the simulation results of the pose estimation in different frequencies are provided.

Figure 3.20 illustrates the estimation results of the *x* and *y* position of the sensor node in the case that the wind loading applies in 1 and 0.5 Hz modes. The wind load varies from -1.5 to 1 kN resulting in a position oscillation with the altitude of 10 and 15 cm respectively in the *x* and *y* directions.

Using only sensor data (depicted by the red line), the x estimation has a maximum error of 4 cm which is about 40% of the maximum altitude. In the y direction, similar error statistics can be seen where the y estimation has a maximum error of 6 cm which is about 60% of the maximum altitude. The green dashed line shows the estimation results of the unknown input observer which takes advantage of the motion model of the light post. Results depict a significant improvement in pose estimation by incorporating the motion model into the estimation algorithm.

The proposed algorithm can be applied to estimate the roll and pitch angles as well. Although, these angles (roll and pitch) are smaller than their related position of x and y respectively. Based on the results provided in Figure 3.21, roll and pitch angles can be estimated with good accuracy for the wind loading f 0.5 and 1 Hz modes. The maximum estimation error is less than 2% of the maximum altitude which verifies the performance of the algorithm in orientation estimation.

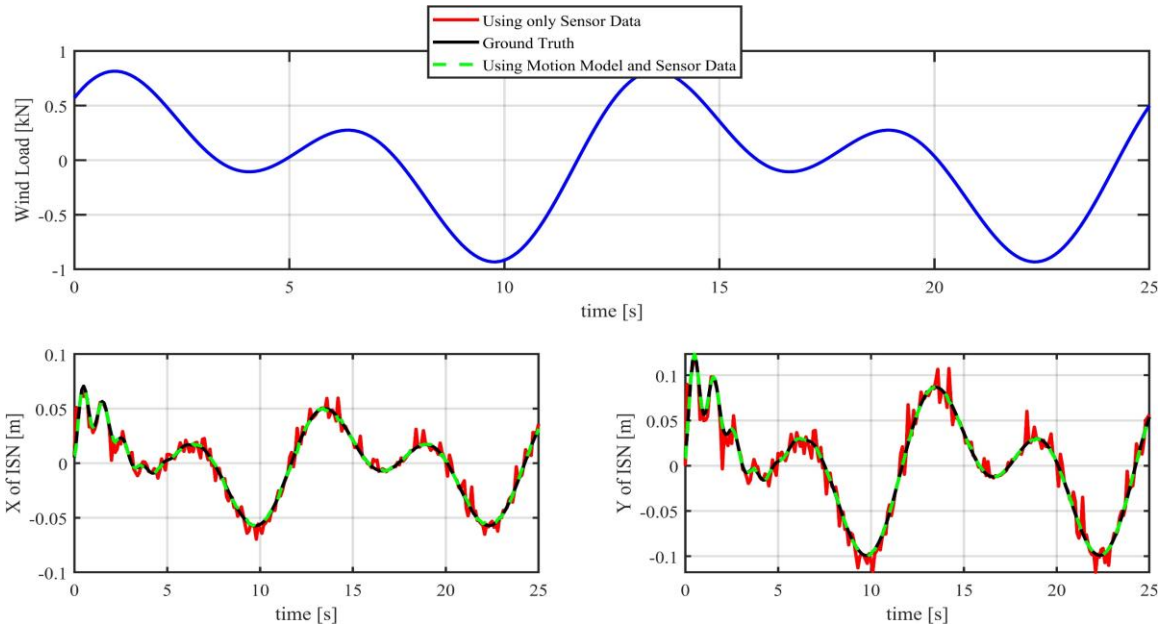


Figure 3.20: Pose estimation for the wind loading applies in 0.5 and 1 Hz modes

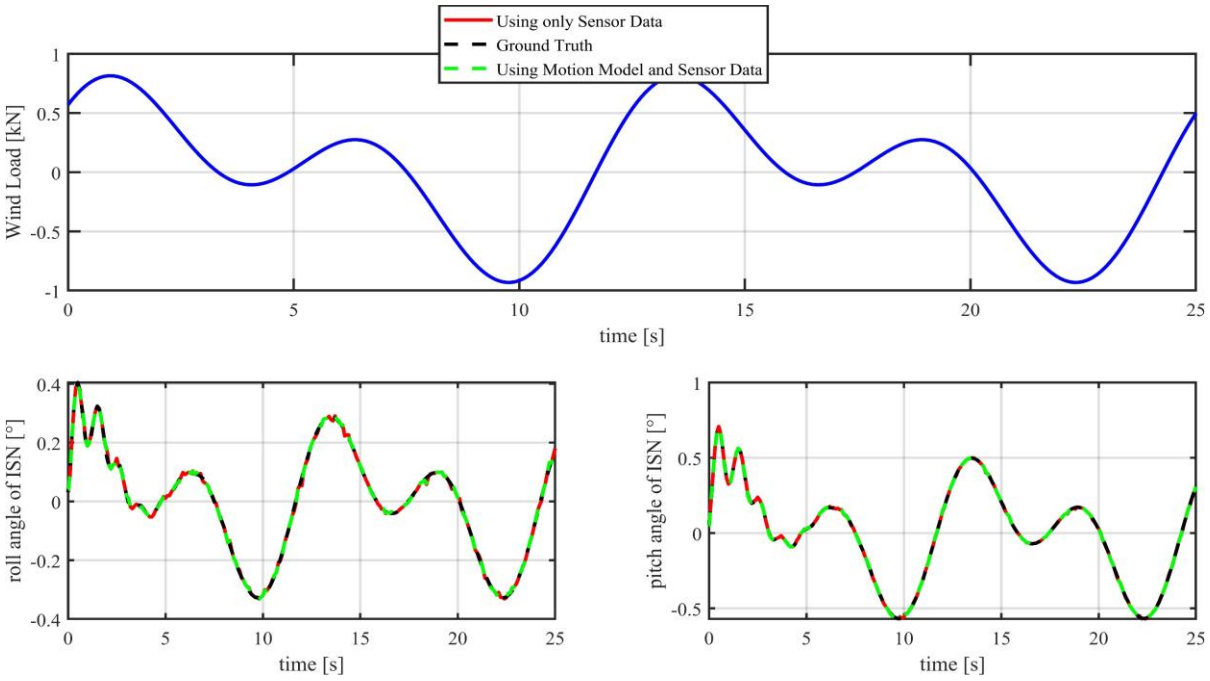


Figure 3.21: Orientation estimation for the wind loading applies in 0.5 and 1 Hz modes

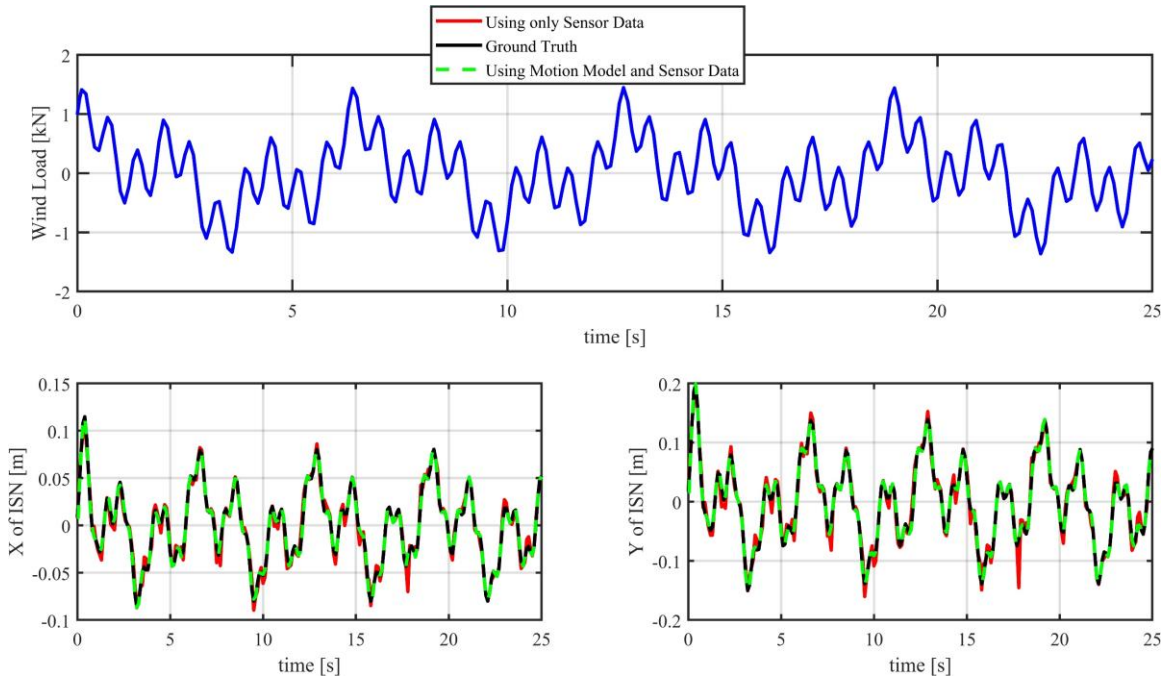


Figure 3.22: Position estimation for the wind loading applies at 1, 3, and 10 Hz modes

Wind loading can be applied in larger frequencies. Consequently, the light post and the sensor node move faster. Figure 3.22 illustrates the motion of the sensor node in a case when the wind loading is at 1, 3, and 10 Hz. Using the sensor data, the maximum estimation error is around 3 and 7 cm for the x and y directions, respectively. On the other hand, the estimation error using the motion model of the light post is almost negligible in both directions. The results show the importance of the motion model in the pose estimation of the sensor node, in high-frequency oscillations.

Orientation estimation, including the roll and pitch angle estimation of the sensor node, can be performed using the sensor data with/without the motion model. Figure 3.23 shows the performance of the algorithm in order to estimate the roll and pitch angle in a high-frequency situation.

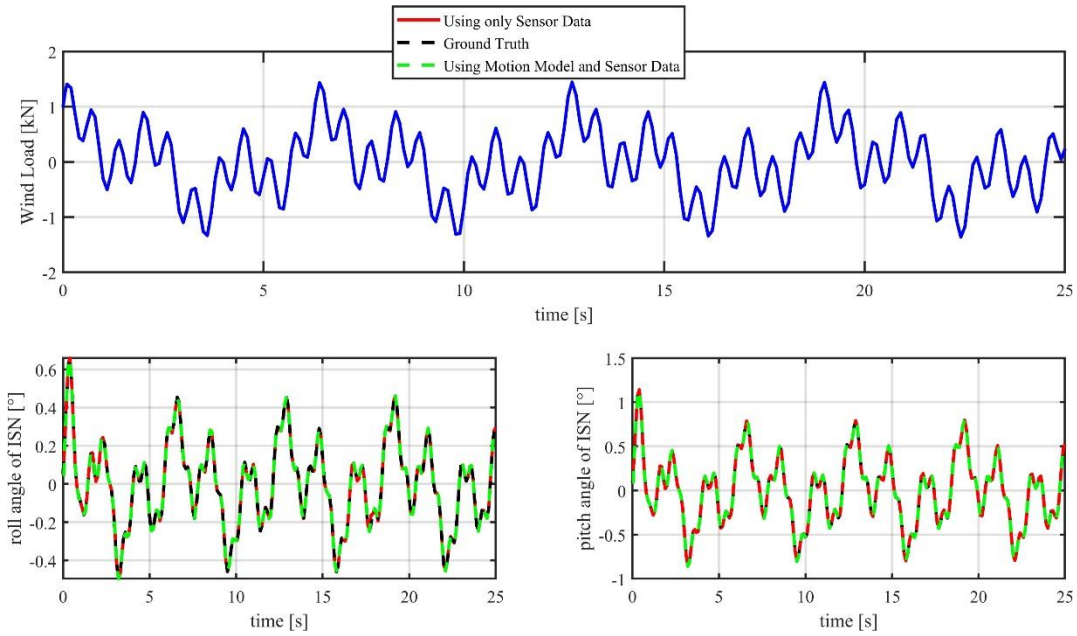


Figure 3.23: Orientation estimation for the wind loading applies at 1, 3, and 10 Hz modes

3.9 Summary

This chapter presents the overall structure of the ISN node. The proposed approach improves the camera-Lidar fusion by considering vehicle motion models, the distance-dependent covariance scaling, and kinematic constraints. This novel approach enhances perception and localization accuracy in autonomous vehicles, as previous similar work did not consider the last two items. For this purpose, different motion models were also presented to implement in the proposed approach. To elaborate, the constant-velocity and kinematic bicycle models are presented for vehicle motion. Incorporating the vehicle motion model and the kinematic constraints, the results show improved accuracy in localization, particularly in turning and double-lane change scenarios. As expected, the accuracy of the ISN is dependent on the distance of the object from the sensor node which is considered in the proposed algorithm.

Furthermore, the chapter proposes a motion compensation module to address the problem of time-varying extrinsic parameters of ISNs, which are caused by external disturbances like the wind. The chapter describes how sensor data, including Lidar and camera, are used to estimate the extrinsic parameters of the sensor node using an unknown input observer that incorporates the motion model of the light post. The proposed algorithm shows reliable and accurate performance in estimating the

extrinsic parameters of the sensor node. Results presented in different frequencies demonstrate the effectiveness of the proposed algorithm in various scenarios, including constant velocity and turning maneuvers.

Chapter 4
Infrastructure-aided PL for urban driving

4.1 Introduction

Chapter 3 introduced the ISN, its components, and its general structure. The output of the ISN is the estimated position and attitude of all objects in its FOV. The output of the ISN node is then broadcast so that any AV can receive and cooperatively use it with its onboard sensor data. This chapter presents the overall framework of the cooperative PL using ISN for urban driving. The cooperative PL is formulated as a fusion problem in which infrastructure data is used in addition to the onboard sensor data to improve the PL performance in terms of overall accuracy. In this context, the AV is first to find the correspondence between the information broadcast by ISNs and its onboard information. It is important to note that information refers to each object's estimated position. The data correspondence framework will be presented in the next section.

4.2 Data Correspondence

As mentioned before, the infrastructure-based sensor node (ISN) provides the position of each object in the FOV with its ID. This information, as the output of ISN, is broadcast so any autonomous vehicle (AV) can receive it. On the other hand, the AV (ego vehicle) also has the positioning awareness of its surrounding environment, meaning that each AV knows the relative position of objects in its coordinate in addition to its assigned ID.

The issue is that the ID assigned to the objects in ISN’s and ego vehicle's coordinates might be different. Therefore, correspondence between them is required for the ego vehicle to use the information broadcast by the ISN for fusing it with its onboard data.

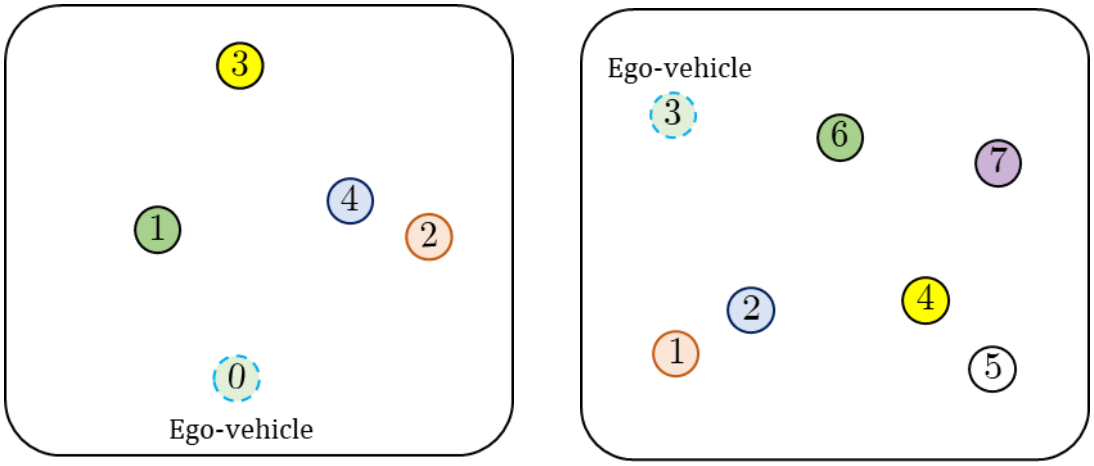
The problem of finding the correspondence between the position of objects in ISN’s and ego vehicle's coordinates can be formulated as point set matching. This section aims to introduce the point set matching method to correspond objects detected in ISN and ego vehicle's coordinates. This module is mentioned in the overall block diagram of the cooperative PL in Figure 1.2 as the data correspondence block.

4.2.1 Rigid Registration for Data Correspondence

Under a rigid transformation (translation and rotation), the distance between any pair of points is preserved. Therefore, with a rigid transformation, the relative positions of objects detected in the ego vehicle's coordinates can be aligned to global positions in ISN's coordinates. As a result, matching the objects in ISN's coordinates with those measured by the ego vehicle is a rigid point set registration problem. Each dynamic object is one point that can be put in either the first or second set, depending on whether it is measured in the ego vehicle's or an ISN's coordinate system.

To formulate the problem, it is assumed that the first point set is $N_k = \{p^G_{1,k}, \dots, p^G_{N,k}\}$ where $p^G_{n,k} \in \mathbb{R}^2$ is the estimated position of object n by the ISN in the world coordinates at time step k. The second point set is $M_k = \{0, p^{rel}_{0,k}, \dots, p^{rel}_{M,k}\}$ where $p^{rel}_{m,k} \in \mathbb{R}^2$ is the estimated positions of object m by the ego vehicle which is in the onboard relative coordinate at time step k. In other words, point set M_k consists of the relative positions of M objects seen by the ego vehicle. The element 0 is related to the ego vehicle itself. By doing this, an imaginary point, related to the ego vehicle, is added to the point set M_k.

Figure 4.1 illustrates the position of the objects measured by the ego vehicle and the ISN, stored in M_k and N_k. It can be seen that when the ISN and ego vehicle detect an object, the ID of the same object detected in the ISN and the ego vehicle is NOT the same. For example, object 2 in the ISN is detected by ID number 4 in the ego vehicle.



(a): Objects in the ego vehicle's Coordinates (b): Objects in ISN's Coordinates

Figure 4.1: Position of objects in different coordinates

Table 4.1: ID correspondence between the objects in the ISN and the-vehicle

ID in the ego vehicle's coordinates	ID in the ISN's coordinates
0	3
1	6
2	1
3	4
4	2
-	5
-	7

The correspondence between the objects in the ISN and the ego vehicle can be seen in Table 4.1. It is also illustrated by the color of the objects in Figure 4.1. The main goal of this section is to find correspondence $C_k = \{c_{m,n}\}_k$ where $c_{m,n}$ determines the cprrepondence between $p_{m,k}^{rel}$ and $p_{n,k}^{G}$ which will be addressed later in this section.

One of the sub-results is to find the point $p_{n,k}^{G}$ from the point set N_k that corresponds to the ego vehicle. Knowing this, the global position of the ego vehicle can be found. In other words, the ego vehicle can identify itself among all objects seen in the ISN. In this framework, an important fact is that the local positions of the objects are transformed to the global positions measured by ISN by rotation matrix and translation vector through the heading angle of the ego vehicle. Therefore, finding the optimal transformation that aligns the point set M_k onto N_k provides the estimation of the heading angle of the ego vehicle, ψ_k^0.

The framework of the data correspondence is depicted in Figure 4.2. The output of the ISN including the global position of objects $p_{n,k}^{G} = \begin{bmatrix} x_{n,j}^{G} & y_{n,j}^{G} \end{bmatrix}^T$, is broadcast. Then, the AV receives it and uses the broadcast information in addition to its local measurements $p_{m,k}^{rel} = \begin{bmatrix} x_{m,k}^{rel} & y_{m,k}^{rel} \end{bmatrix}^T$, in the point set matching part. In this part, unknown correspondence and transformation are calculated. Performing the algorithm, the ego vehicle can find itself among all other objects and also know the position of even the objects that cannot see by themselves due to occlusion.

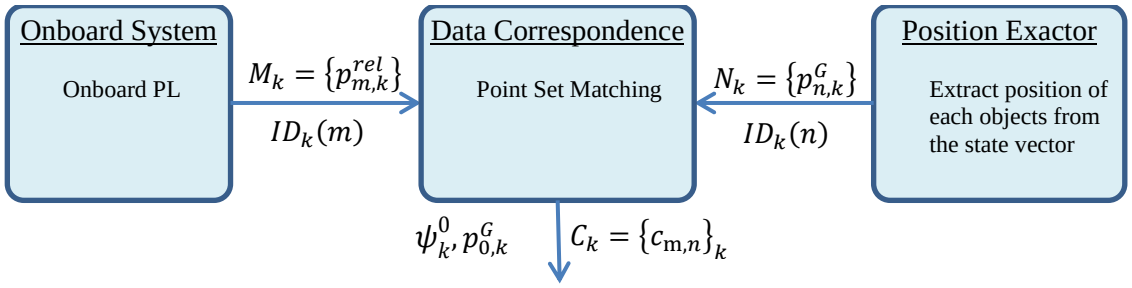


Figure 4.2: Block diagram of data correspondence module

The first output of the data correspondence module is correspondence between the estimated position of the objects in two coordinates of the ISN and the onboard AV's system. It means the IDs of the objects are unique in both coordinates.

Another output is the global position and heading angle of the AV in ISN's coordinate. To elaborate, the rotation and translation of the second point set with respect to the first point set is the orientation and translation of the onboard coordinates with respect to the ISN's coordinates. Since the ISN are stationary, the ISN coordinates can easily be transformed into global coordinates.

4.2.1.1 Occlusion cases

In occlusion cases, objects are obstructed from the view of onboard sensors. As can be seen in Table 4.1, two objects can NOT be seen by the ego vehicle. This situation frequently happens in urban driving, especially in crowded intersections where some objects are occluded by some other objects. As a result, the ego vehicle is not aware of the presence of some objects which is a critical safety issue. As mentioned in Chapter 1, this is one of the main motivations of this work. On the other hand, ISNs can detect almost all objects since they have a bird's-eye view. As a result, $N > M$ in occlusion cases. From data correspondence perspective, inequality between the size of M_k and N_k is not an issue, since the correspondence can be performed between two point sets with different sizes. Even if one of the point set is a subset of the other point set, the correspondence between the points can be done by the following approaches.

In the fusion context, the ego vehicle has a less number of measurements, compared to the ISN. Therefore, occlusion cases should be addressed properly. In these cases, some objects can not be seen by the ego vehicle, in other words, some objects are missing in the ego vehicle's measurements.

Therefore, the number of objects detected by the ego vehicle, stored in M_k, is less than the detected objects by the ISN, stored in N_k. Since the positions of objects are implemented as measurements, the missing objects (occluded objects) should be addressed.

For this purpose, after finding the ID correspondence between point sets N_k and M_k, for the points that do not have any correspondence, dummy points are added so the number of points in the ISN and the ego vehicle's coordinates becomes equal. For instance, if M_k contains 5 points including the ego vehicle, itself and N_k has 6 points, which means that one object is occluded and missing from the ego vehicle's sensors. To address the occlusion issue, one dummy point will be added to M_k corresponds to the missing object. In the ego vehicle's onboard sensor node, this dummy point will be implemented as a dummy measurement with a significantly large measurement covariance. As a result, regarding the missing objects, the cooperative PL almost entirely relies on the ISN which is less likely exposed to occlusion.

Consequently, by fusing broadcast information by the ISNs with onboard sensor data, the ego vehicle will have a total position awareness of its surrounding, even in occlusion cases. The details of how the proposed method addresses the occlusion issue will be provided later in this chapter.

4.2.2 Point Set Matching for Data Correspondence

The inputs of the data correspondence module are two point sets. The first point set $N_k = \{p_{1,k}^G, \dots, p_{N,k}^G\}$ includes the estimated positions of the objects by the ISN in the world coordinates. The second point set $M_k = \{0, p_{0,k}^{rel}, \dots, p_{M,k}^{rel}\}$ involves the estimated positions of the objects by the ego vehicle which are in the onboard relative coordinate.

Algorithm 4.1 below presents the procedure of point set matching. In this point set matching, two point sets $N_k = \{p_{1,k}^G, \dots, p_{n,k}^G\}$ and $M_k = \{0, p_{1,k}^{rel}, \dots, p_{m,k}^{rel}\}$ are assumed to be given and the goal is to find the transformation, rotation matrix r_k^{MN} and translation vector t_k^{MN}, that minimizes $\sum_{(m,n)\in C_k} \| p_{n,k}^G - r_k^{MN} p_{m,k}^{rel} - t_k^{MN} \|$. The binary correspondence C_k between M_k and N_k, is unknown. The binary correspondence in the ICP method implies that if $p_{m,k}^{rel}$ corresponds to $p_{n,k}^G$, then $c_{\mathrm{m},n} = 1$, otherwise $c_{\mathrm{m},n} = 0$.

The main idea behind Algorithm 4.1 is to start from an initial guess of rotation, and translation shown by $M_{k,0}$, $r_{k,0}^{MN}$ and $t_{k,0}^{MN}$ for the first point set, and to compute the new rotation and translation matrices

$r_{k,1}^{MN}, t_{k,1}^{MN}$. Then, the new $r_{k,1}^{MN}, t_{k,1}^{MN}$ are applied to the first point set $M_{k,0}$ to find the transformed point set $M_{k,1}$. This loop continues until $r_{k,j}^{MN}, t_{k,j}^{MN}$ respectively converges to I and $\vec{0}$ where I is the identity matrix of 2×2 size. This condition means the two point sets are aligned and they are not going to be rotated nor translated, as a result C_k is found. In this context, subscript j is the iterator of the while loop, described in Algorithm 4.1.

In the algorithm, j_{max} is used as the maximum number of iterations. The maximum number of iterations to achieve convergence depends on the size of the point set. In this work, the number of objects detected by the sensors is the size of the point set. If the number of objects is assumed not to exceed 100, j_{max} is considered 100.

Algorithm 4.1: Iterative closest point

// **Input:** Point sets $\boldsymbol{N_k}$ and $\boldsymbol{M_k}$, initial guess $\boldsymbol{r_{k,0}^{MN}, t_{k,0}^{MN}}$, convergence criteria $\boldsymbol{\varepsilon}$, and maximum number of iterations $\boldsymbol{j_{max}}$

// **Output:** The optimal transformation $\boldsymbol{r_{k,*}^{MN}, t_{k,*}^{MN}}$ that best aligns $\boldsymbol{M_k}$ onto $\boldsymbol{N_k}$ with correspondence matrix $\boldsymbol{C_k}$

Initialize with guess $\boldsymbol{M_{k,0} = M_k}$

for $\boldsymbol{j = 1\ to\ j_{max}}$ **do**

Find the closest points in $\boldsymbol{M_{k,j}}$ for each point in $\boldsymbol{N_k}$ using the nearest neighbor search.

Determine the binary correspondence $\boldsymbol{C_k = \{c_{m,n}\}_k}$ based on the closest point in $\boldsymbol{M_{k,j}}$ to $\boldsymbol{N_k}$.

Compute the registration $\boldsymbol{r_{k,j}^{MN}, t_{k,j}^{MN}}$ between $\boldsymbol{M_{k,j}}$ and $\boldsymbol{N_k}$.

Apply for the registration $\boldsymbol{M_{k,j+1} = r_{k,j}^{MN} M_{k,j} + t_{k,j}^{MN}}$.

// Check for convergence:

Compute the $\boldsymbol{error}$ between $\boldsymbol{M_{k,j+1}}$ and $\boldsymbol{N_k}$ using Euclidean distance.

if $\boldsymbol{error < \varepsilon}$

return $\boldsymbol{r_{k,j}^{MN}, t_{k,j}^{MN}}$

end if

end for

return $\boldsymbol{r_{k,j}^{MN}, t_{k,j}^{MN}}$

In each iteration, $error$ is calculated using a distance metric, Euclidean distance in this work, between every point in two point sets $M_{k,j+1}$ and N_k. Therefore, the Euclidean norm is used to calculate $error$. Convergence criteria, ε, is then compared with $error$ to check for the convergence of the algorithm. In this work, each point in the point sets represents the position of the object. Considering the order of position estimation of the objects which is around 0.1 m. Therefore, it is reasonable to consider ε to be 0.1. it is required to mention that an empirical test should be conducted to determine the mentioned parameters for a particular application by taking into account the specific characteristics of the point clouds and the application's requirements.

The performance of Algorithm 4.1 is dependent on the initial guess. Local minimum and initial guess are two main limitations of this method. As mentioned in Chapter 2, Many variants of ICP have been proposed that affect all phases of the algorithm from the selection and matching of points to the minimization strategy. The next section presents the point set matching to address the above-mentioned issue to make the algorithm the engine of the data correspondence module.

4.2.3 Coherent Point Drift for Data Correspondence

The point set matching method implemented in this work is adapted from the Coherent Point Drift (CPD) algorithm in [78]. The methodology of the method will be presented in this section. Then, the rigid point set matching algorithm will be presented. The algorithm introduced in this section was mentioned in Figure 4.2 as the "point set matching" block.

The same as the previous section, the point set $N_k = \{p_{1,k}^G, \dots, p_{N,k}^G\}$ includes the estimated positions of the objects broadcast by the ISN. In other words, $p_{n,k}^G = \begin{bmatrix} x_{n,j}^G & y_{n,j}^G \end{bmatrix}^T$, is the estimated position of object n in the global coordinate. Point set $M_k = \{0, p_{0,k}^{rel}, \dots, p_{M,k}^{rel}\}$ involves the estimated position of the objects by the ego vehicle in the onboard relative coordinate. Therefore, $p_{m,k}^{rel} = \begin{bmatrix} x_{m,k}^{rel} & y_{m,k}^{rel} \end{bmatrix}^T$ is the estimated relative x and y position of object $m \in \{0, \dots, M\}$ in the onboard coordinates.

In this method, the correspondence matrix, C_k, contains a probabilistic assignment of correspondence between pair of points, based on its likelihood of being a true correspondence. Using a probabilistic correspondence matrix, this approach is more robust to noise and outliers, as mentioned in [78].

In this method, $p_{m,k}^{rel}, m \in \{0, \dots, M\}$ is considered as the centroids of GMM and $p_{n,k}^{G}, n \in \{1, \dots, N\}$ as the data points generated by it. GMM has a probability density function as follows:

$$p\left(p_{n,k}^{G}\right) = \sum_{m=0}^{M} p(m) p\left(p_{n,k}^{G} \middle| m\right) \tag{4.1}$$

where $p\left(p_{n,k}^{G} \middle| m\right) = \frac{1}{2\pi\sigma^2} exp^{-\frac{\left\| p_{n,k}^{G} - p_{m,k}^{rel} \right\|^2}{2\sigma^2}}$ is the Gaussian distribution centered on a point $p_{m,k}^{rel}, m \in \{0, \dots, M\}$.

In addition to $p\left(p_{n,k}^{G} \middle| m\right)$, an additional uniform distribution $p\left(p_{n,k}^{G} \middle| M+1\right) = \frac{1}{N}$ is added to the mixture model. This assumption takes accounts for noise and outliers. Considering the isotropic covariance of $\sigma^2 = \frac{1}{2N(M+1)} \sum_{n=1}^{N} \sum_{m=0}^{M+1} \left\| p_{n,k}^{G} - p_{m,k}^{rel} \right\|^2$ and using w for the weight of the uniform distribution, the GMM density function can be written as:

$$p\left(p_{n,k}^{G}\right) = w \frac{1}{N} + (1-w) \sum_{m=0}^{M} \frac{1}{M} p\left(p_{n,k}^{G} \middle| m\right) \tag{4.2}$$

The location of the GMM centroids can be reformulated by using a set of parameters $\mathcal{T}$ which consists of r_k^{MN}, t_k^{MN}. The parameters $\mathcal{T}$ can be then estimated by maximizing the likelihood or equivalently by minimizing the negative log-likelihood function:

$$E(\mathcal{T}, \sigma^2) = -\sum_{n=1}^{N} \log \sum_{m=0}^{M} p(m) p\left(p_{n,k}^{G} \middle| m\right) \tag{4.3}$$

The correspondence probability between two points $p_{m,k}^{rel}$ and $p_{n,k}^{G}$ is defined as the posterior probability of the GMM centroid given the data point as $P\left(m \middle| p_{n,k}^{G}\right) = P(m) p\left(p_{n,k}^{G} \middle| m\right) / p\left(p_{n,k}^{G}\right)$.

The expectation maximization (EM) algorithm is used to find $\mathcal{T}$ and σ^2. Initial guess on parameters is used to calculate the posterior probability distributions $P^{old}\left(m \middle| p_{n,k}^{G}\right)$ which has the following form:

$$P^{old}\left(m \middle| p_{n,k}^{G}\right) = \frac{exp^{-\frac{1}{2}\left\| \frac{p_{n,k}^{G} - \mathcal{T}\left(p_{m,k}^{rel}, \theta^{old}\right)}{\sigma^{old}} \right\|^2}}{\sum_{m=0}^{M+1} exp^{-\frac{1}{2}\left\| \frac{p_{n,k}^{G} - \mathcal{T}\left(p_{m,k}^{rel}, \theta^{old}\right)}{\sigma^{old}} \right\|^2} + c} \tag{4.4}$$

where $c = 2\pi\sigma^2 \frac{w}{1-w}\frac{M+1}{N}$. Minimizing the expectation of the complete negative log-likelihood function, Q, the new parameter values will be found. The last two steps will be iterated until convergence:

$$Q(\mathcal{T}, \sigma^2) = -\sum_{n=1}^{N}\sum_{m=0}^{M+1} P^{old}\left(m\middle|p_{n,k}^{G}\right)\log\left(P^{new}(m)P^{new}\left(p_{n,k}^{G}\middle|m\right)\right) \tag{4.5}$$

Rewriting (4.5) based on $\mathcal{T}$ and σ^2 gives:

$$Q(\mathcal{T}, \sigma^2) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{m=0}^{M+1} P^{old}\left(m\middle|p_{n,k}^{G}\right)\left\|p_{n,k}^{G} - \mathcal{T}\left(p_{m,k}^{rel}\right)\right\|^2 + + N_p \log\sigma^2 \tag{4.6}$$

where $N_p = \sum_{n=1}^{N}\sum_{m=0}^{M+1} P^{old}\left(m\middle|p_{n,k}^{G}\right) \leq N$

4.2.3.1 Rigid point set registration

In rigid point set registration, the transformation of the location of the GMM centroids is defined as follows:

$$\mathcal{T}\left(p_{m,k}^{rel}; r_k^{MN}, t_k^{MN}\right) = r_k^{MN} p_{m,k}^{rel} + t_k^{MN} \tag{4.7}$$

where r_k^{MN} is the rotation matrix and t_k^{MN} is the translation vector. Substituting $\mathcal{T}$ into (4.6), the objective function Q becomes:

$$\begin{aligned} &Q\left(r_k^{MN}, t_k^{MN}, \sigma^2\right) \\ &\qquad = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{m=0}^{M+1} P^{old}\left(m\middle|p_{n,k}^{G}\right)\left\|p_{n,k}^{G} - \left(r_k^{MN} p_{m,k}^{rel} + t_k^{MN}\right)\right\|^2 + N_p \log\sigma^2 \end{aligned} \tag{4.8}$$

The exact minimization solution of the objective function (4.8) can be obtained under certain conditions. It is proved in [78] that if function Q has the form of $tr(A^T R)$, the optimal rotation matrix r_k^{MN} that maximizes $Q = tr\left(A^T r_k^{MN}\right)$ is $r_k^{MN} = UCV^T$, where UCV^T is the singular value decomposition (SVD) of A, $VV^T = I$, $UU^T = I$, and $C = diag\left(1, 1, \ldots, det(UV^T)\right)$. Therefore, $r_{k,j}^{MN}$ can be calculated by computionally light algebraic equation.

The procedure of rigid point set registration can be seen in Algorithm 4.2. The inputs of the algorithm are two sets of points M_k and N_k, defined at the beginning of Section 4.2.3. The output is then the

transformation between the ISNs and the AV's coordinates, $r_{k,*}^{MN}, t_{k,*}^{MN}$, to give the correspondence between the estimated positions of the objects in the mentioned coordinates.

Algorithm 4.2: Rigid point set registration

// **Input:** Point sets $\boldsymbol{N_k}$ and $\boldsymbol{M_k}$, initial guess $\boldsymbol{r_{k,0}^{MN}, t_{k,0}^{MN}}$, convergence criteria $\boldsymbol{\varepsilon}$, and maximum number of iterations $\boldsymbol{j_{max}}$

// **Output:** The optimal transformation $\boldsymbol{r_{k,*}^{MN}, t_{k,*}^{MN}}$ that best aligns $\boldsymbol{M_k}$ onto $\boldsymbol{N_k}$ with the probability of correspondence, $\boldsymbol{C_k}$

Initialize with $\boldsymbol{M_{k,0} = M_k}$ and $\boldsymbol{0 \leq w \leq 1}$

for $\boldsymbol{j = 1\ to\ j_{max}}$ **do**

// Compute correspondence:

$$\boldsymbol{C_{mn} = \frac{exp^{-\frac{1}{2\sigma^2}\left\| p_{n,k}^{G} - \left(r_{k,j}^{MN} p_{m,k}^{rel} + t_{k,j}^{MN} \right) \right\|^2}}{\sum_{m=0}^{M+1} exp^{-\frac{1}{2\sigma^2}\left\| p_{n,k}^{G} - \left(r_{k,j}^{MN} p_{m,k}^{rel} + t_{k,j}^{MN} \right) \right\|^2} + 2\pi\sigma^2 \frac{w}{1-w} \frac{M+1}{N}}}$$

// Find the new registration:

Solve for $\boldsymbol{r_{k,j+1}^{MN}, t_{k,j+1}^{MN}}$

// update point set:

$$\boldsymbol{M_{k,j+1} = r_k^{MN} M_{k,j} + t_k^{MN}}$$

// Check for convergence:

Compute the $\boldsymbol{error}$ between $\boldsymbol{M_{k,j+1}}$ and $\boldsymbol{N_k}$ using Euclidean distance.

if $\boldsymbol{error < \varepsilon}$

return $\boldsymbol{r_{k,j}^{MN}, t_{k,j}^{MN}}$

end if

end for

return $\boldsymbol{r_{k,j}^{MN}, t_{k,j}^{MN}}$

4.3 Estimation formulation for Fusion

After the data correspondence, the position information of objects from ISNs can be fused with onboard information. The estimation methods which apply to this problem are reviewed in this section and a comparison is provided to select the suitable approach.

As mentioned before, there are two general approaches to address the data fusion problem in the localization application, optimization-based and filter-based. To choose a suitable approach for this application, it is necessary to review these methods and compare their strengths and weaknesses to decide which approaches to take as the fusion method.

4.3.1 Optimization-based methods

Optimization-based methods formulate the localization problem as an optimization problem using Least Squares, Maximum a Posteriori (MAP), or Maximum Likelihood method [94]. Since optimization-based methods can explicitly use nonlinear models, they are more accurate due to iterative linearization. Besides, state and noise constraints can be easily incorporated into the optimization formulation compared to the filter-based methods.

However, optimization-based methods have a serious drawback in real-time applications due to their computational complexity. They become increasingly complex in terms of computation as they incorporate observed measurements and the previously calculated states into the next step of the optimization algorithm. In addition, solving an optimization problem is dependent on a number of factors, such as initial guesses and the number of iterations, etc. Therefore, the calculation time cannot be predicted [95]. Computation complexity can be reduced or bound by adopting some techniques, although none of them took advantage of considering the constraints on state variables and process uncertainties, which is one of the important benefits of optimization-based solutions.

4.3.2 Filter based methods

The Bayesian filtering techniques including Particle Filter (PF), and Kalman Filter (KF), and their variations are usually used to estimate the states of the ego vehicle. These techniques incorporate noise and vehicle dynamics by using linear/non-linear or Gaussian/non-Gaussian assumptions. KF and its different variations like Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are widely implemented for vehicle localization [50], [48].

The results show that they are computationally light and can be easily implemented [95]. In linear systems with Gaussian noise, KF yields a recursive solution to the unconstrained MAP which means under the above conditions, there is no justification to use optimization methods such as MAP since they require more computations to provide the same result. This is more crucial for infrastructure-aided localization in which communication issues like delay exacerbate the real-time computation problem.

In [48], the authors propose a combined PF and KF to estimate the vehicle's position. To do this, the algorithm uses GPS measurements and also signal strength indicators (SSI) as input to a Markov-based localization algorithm. The information is transmitted from RSUs to the target vehicle by V2I and V2V communication. First-order Taylor approximation is used to formulate the motion of the vehicle.

A distributed interactive multiple-model KF is implemented to fuse local information and the PF algorithm is used to fuse the transmitted information. In another work [44], both V2V and V2I have been used to improve the local estimation of the vehicle's position in tunnels. PF-based data fusion has been implemented to fuse cooperative information with onboard sensor data. The kinematic bicycle model is used to model the vehicle dynamics.

4.4 Cooperative PL

This section presents the framework of the cooperative PL that the AV uses to fuse the broadcast information by the ISN with its onboard data. Figure 4.3 illustrates the structure of the cooperative PL. The inputs of the cooperative PL are local estimations of ISNs and also the local estimation of the ego vehicle.

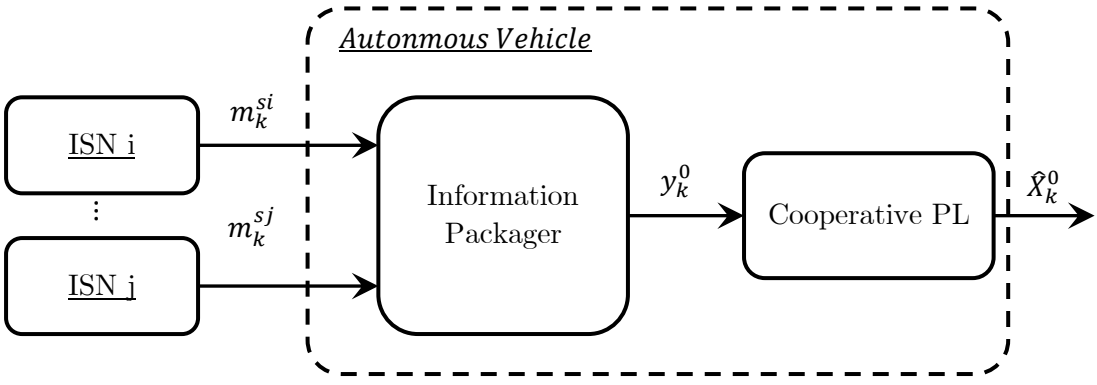


Figure 4.3: Block diagram of cooperative PL

As mentioned before, the main objective is to use data from the ISN node which includes the camera and Lidar in addition to onboard sensor data to improve the overall estimation performance. In the KCF context, the local estimators, which use sensor data, can exchange data with other neighboring nodes to provide cohesive estimates.

Since the accuracy of the estimation provided by each local estimator can hypothetically change over time during the tracing, it is expected to have a better performance using infrastructure data in addition to onboard sensor data.

As mentioned in Chapter 3, the output of ISNs is the estimated positions of objects, stored in N_k, and their corresponding measurement covariance $R_{i,k}$. In the context of cooperative PL, each sensor node i has the following measurement model:

$$Z_k^{si} = H^{si} X_k + v_k^{si}, \quad Z_k^{si} \epsilon \mathbb{R}^{2\times N} \tag{4.9}$$

where $Z_k^{si} = N_k$ and $v_k^{si} \sim \mathcal{N}\left(0, R_k^{si}\right)$.

Then each sensor node i forms the following information package to broadcast:

$$u_k^{si} = \left(H^{si}\right)^T {R_{i,k}}^{-1} Z_k^{si} \tag{4.10}$$

$$U_k^{si} = \left(H^{si}\right)^T {R_{i,k}}^{-1} H^{si}$$

where H^{si} is the observation matrix of ith sensor with the flowing measurement model:

For the infrastructure sensor, all measurements (positions of all targets) are in the global coordinate, thus:

$$H^{si} = \text{diag}(\begin{bmatrix} H_{t1}^{si} & \cdots & H_{tj}^{si} \end{bmatrix}) \tag{4.11}$$

Assuming for object j, the state vector is $X_k^j = \begin{bmatrix} X_k^j & Y_k^j & \psi_k^j \end{bmatrix}^T$, H_{tj}^{si} represents the observation matrix corresponding to the jth object with the following form:

$$H_{tj}^{si} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.12}$$

Measurements obtained by onboard sensors include positions of the ego and other vehicles. The position of the ego vehicle is in global coordinates, while others are relative to the ego vehicle. Therefore measurement model for the ego vehicle has the following form:

$$H^{AV} = \begin{bmatrix} H_{t1}^{AV} & \cdots & H_{tj}^{AV} \end{bmatrix}^T \quad (4.13)$$

where H_{tj}^{AV} is the observation matrix of the ego vehicle's measurement corresponding to the jth object with the following form:

$$H_{tj}^{AV} = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (4.14)$$

Then, each sensor node broadcast the information package $m_k^{si} = \{u_k^{si}, U_k^{si}, R_k^{si}\}$. Each AV receives the broadcast information from all the sensor nodes and forms the following matrices:

$$y_k^0 = \sum_{i \in I} u_k^{si} \quad (4.15)$$

$$S_k^0 = \sum_{i \in I} U_k^{si}$$

where I is the number of ISNs that have V2I communication with the AV and the AV can receive the information from them. The AV then can perform the cooperative PL algorithm using Kalman-Consensus Filter (KCF) method proposed in [96].

The position and the velocity of targets in the global coordinate frame are considered the state X_k and $\hat{X}_k^{si}$ is referred to as the local estimates of sensor i with the estimated covariance matrix of P_k^{si}. To simplify the notations, the assumption is that two targets are tracked by two sensors in this framework, although the number of sensors and targets can be extended to any number.

4.4.1 Kalman- Consensus Filter

KCF is implemented as the main algorithm of the "Cooperative PL" module. In this context, the dynamic model of the targets is assumed to have the following form:

$$X_{k+1} = FX_k + Gu_k + \omega_k; \quad (4.16)$$

$$X_0 \epsilon \mathcal{N}(\bar{X}_0, P_0)$$

where X_k and ω_k are respectively the states and input noise of the process at time k, and X_0 is the initial state with a Gaussian distribution.

The states of the target ate tracked by multiple sensor nodes. Each track has a linear measuring model:

$$Z_k^{si} = H^{si} X_k + v_k^{si} \tag{4.17}$$

where H^{si} and v_k^{si} are respectively observation model and observation noise of sensor node i. It is assumed that $\omega_k \sim \mathcal{N}(0, Q_k)$, $v_k^{si} \sim \mathcal{N}\left(0, R_k^{si}\right)$.

The goal is to have cohesive estimations $\hat{X}_k^{si}$, calculated by each sensor node, using data exchange between neighboring sensor nodes.

Assume that ω_k and v_k^{si} are zero-mean white Gaussian noise with the following statistics:

$$\begin{aligned} E[\omega_k {\omega_k}^T] &= Q_k \delta_{kl} \\ E\left[v_k^{si} {v_k^{si}}^T\right] &= R_k^{si} \delta_{kl} \end{aligned} \tag{4.18}$$

The observability matrix can be written as [97]:

$$\mathcal{O}^n = [{H_t}^T \quad (H_t F)^T \quad \dots \quad (H_t F^{n-1})^T]^T \tag{4.19}$$

where $F = I$ based on the motion model (3.3). The system is observable since $rank(\mathcal{O}^n) = n$.

For the mentioned dynamic system, KCF has the following distributed estimation algorithm.

$$\begin{cases} \hat{X}_k^{si} = \bar{X}_k^{si} + K_k^{si}\left(Z_k^{si} - H^{si}\bar{X}_k^{si}\right) + C_i \sum_{j \in N_i} \left(\bar{X}_k^{sj} - \bar{X}_k^{si}\right) \\ K_k^{si} = P_k^{si} {H^{si}}^T \left(R_k^{si} + H^{si} P_k^{si} {H^{si}}^T\right)^{-1} \\ M^i = F P_k^{si} F^T + K_k^{si} R_k^{si} {K_k^{si}}^T \\ {P_k^{si}}^+ = F M^i F^T + G_k Q_k {G_k}^T \\ {\bar{X}_k^{si}}^+ = F \hat{X}_k^{si} \end{cases} \tag{4.20}$$

Algorithm 4.3: Cooperative PL algorithm of the ego vehicle

// **Input:** Information Packages $\boldsymbol{m_k^{si}} = \{\boldsymbol{u_k^{si}}, \boldsymbol{U_k^{si}}, \boldsymbol{R_k^{si}}\}$ from all neighboring sensor nodes $\boldsymbol{N^0}$, and their correspondence matrix $\boldsymbol{C_k}$ with the ego vehicle.

// **Output:** Cohesive estimation $\widehat{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}}$, estimated by the ego vehicle.

Receive the broadcast information by sensor nodes $\boldsymbol{m_k^{si}} = \{\boldsymbol{u_k^{si}}, \boldsymbol{U_k^{si}}, \boldsymbol{R_k^{si}}\}$.

// **Update the state of the filter:**

$$\boldsymbol{P_k^0} = \boldsymbol{F M^0 F^T} + \boldsymbol{G_k Q_k G_k}^{\boldsymbol{T}}$$

$$\bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}} = \boldsymbol{F}\widehat{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}} + \boldsymbol{G_k u_k}$$

// **Fuse information matrices and vectors:**

$$\boldsymbol{y_k^0} = \sum_{\boldsymbol{i}\in\boldsymbol{N^0}} \boldsymbol{u_k^{si}}\ ,\ \boldsymbol{S_k^0} = \sum_{\boldsymbol{i}\in\boldsymbol{N^0}} \boldsymbol{U_k^{si}}$$

// **Compute the cooperative state estimate:**

$$\widehat{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}} = \bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}} + \boldsymbol{M^0}\left(\boldsymbol{y_k^0} - \boldsymbol{S_k^0}\bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}}\right) + \boldsymbol{\gamma P_k^0}\sum_{\boldsymbol{i}\in\boldsymbol{N^0}}\left(\bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{si}} - \bar{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}}\right)$$

$$\boldsymbol{M^0} = \left(\left(\boldsymbol{P_k^0}\right)^{-\boldsymbol{1}} + \boldsymbol{y_k^0}\right)^{-\boldsymbol{1}}$$

$$\boldsymbol{\gamma} = \boldsymbol{\epsilon}/\left(\left\|\boldsymbol{P_k^0}\right\| + \boldsymbol{1}\right),\ \left\|\boldsymbol{P_k^0}\right\| = \boldsymbol{tr}\left(\left(\boldsymbol{P_k^0}\right)^{\boldsymbol{T}}\boldsymbol{P_k^0}\right)^{-\boldsymbol{1}}$$

return $\widehat{\boldsymbol{X}}_{\boldsymbol{k}}^{\boldsymbol{0}}$

Algorithm 4.3. shows the cooperative PL algorithm. The exchanged data includes m_k^{si} where $i \in N^0$ are the neighboring nodes of the ego vehicle, referred to as the subscript *0*.

In the algorithm, ϵ is a small constant that is set to be equal to the sampling time, based on the stability conditions, described in the next section.

4.4.1.1 Stability Analysis

KCF described in eq. (4.20) can be implemented on the system with the model and measurements of noise covariance of $Q_k = \mathbb{E}[\omega_k, {\omega_k}^T]$ and $R_k^{si} = \mathbb{E}\left[v_k^{si}, {v_k^{si}}^T\right]$. It is assumed that the model and measurement noises, described in the previous section, are uncorrelated and have zero means.

Based on the theorem proved in [98], the error dynamics of KCF (4.20) are globally asymptotically stable under the following sufficient conditions.

The main assumption for the approximation of KCF is considering the consensus gain to be of the order of ϵ. Therefore:

$$C_i = O(\epsilon) \tag{4.21}$$

where ϵ is a small positive constant which is chosen to of the order of the time step of the process model. This assumption was the reason that any C_i factors were eliminated from M_i in (4.20). In fact, all C_i factors was set to zero in M^i, resulting in a scalable covariance propagation.

The chosen consensus gain C_i to satisfy (4.20) has the following form:

$$C_i = \gamma A_k^i B_k^i \tag{4.22}$$

where γ consequently is positive and sufficiently small and

$$A_k^i = I - K_k^{si} H^{si} \tag{4.23}$$

$$B_k^i = F M^i F^T + G_k Q_k {G_k}^T + P_k^{si} U_k^{si} P_k^{si} \tag{4.24}$$

Assuming the information matrix $U_k^{si} = \left(H^{si}\right)^T {R_k^{si}}^{-1} H^{si}$ to be positive definite for all time step k, (4.21) implies that the error dynamics of the filter of (4.20) is globally asymptotically stable with a Lyaounov function $V(\eta) = \eta^T M^{-1} \eta$ where η is the estimation error.

To guarantee that the cooperative PL system in this work is globally asymptotically stable, the mentioned sufficient conditions are examined as follows:

In the cooperative PL, R_k^{si} is supposed to have a quadratic relationship with distance, as mentioned before in equation (3.16). Furthermore, all the measurements are independent and uncorrelated. Therefore, R_k^{si} is positive definite. Given H^{si} in (4.12) and (4.14), the information matrix U_k^{si} is positive definite for any sensor node i at all time steps.

The other condition is A_k^i and B_k^i in (4.22). Both can be easily formed by having H^{si}, K_k^{si} and U_k^{si} well defined, as mentioned in equations (4.23, 24).

The last sufficient condition is regarding γ. For the cooperative PL system, γ is supposed to be smaller than an upper bound γ^* with the following definition:

$$\gamma^* = \left(\frac{\lambda_{min}(\Lambda_k)}{\lambda_{min}(\hat{F}^T \hat{L} B_k \hat{L} \hat{F})}\right)^{\frac{1}{2}} \quad (4.25)$$

where $\Lambda_k = diag(\Lambda_k^1, \dots, \Lambda_k^i)$, $\Lambda_k^i = M^{i-1}\left(M^{i-1} + F^T\left(G_k Q_k {G_k}^T + P_k^{si} U_k^{si} P_k^{si}\right)^{-1} F\right)^{-1}$. Furthermore, $B_k = diag(B_k^1, \dots, B_k^i)$, $\hat{F} = I_m \otimes F$, $\hat{L} = I_m \otimes L$ where I_m is an identity matrix, $\otimes$ represents the Kronecker product, and L is the graph Laplacian of the (sensor) network.

The parameter γ is set as $\gamma < \gamma^*$ to guarantee the stability of the consensus in the cooperative system of this work.

4.5 Simulation results

In this section, the results of implementing infrastructure-based camera data in vehicle PL is presented. The ISN node provides the PL for the vehicle performing different maneuvers. The next section will provide examples of implementing the ISN in addition to the vehicle-based sensors. Similar to the previous chapter, the results are based on the estimation error which is the difference between the estimated value and the ground truth.

4.5.1 Constant velocity motion

Figure 4.4 illustrates the configuration of the ISN node and two vehicles moving with constant velocities. The ISN node (s2) is placed at a known position in the global coordinate. In this scenario, The ISN module includes Lidar and camera sensors which are mounted at 10 m height from the ground level. Another measurement is obtained by the onboard sensor (s1) which is moving with the ego vehicle (the red one).

Figure 4.4: Constant velocity scenario

The Gaussian noise is added to both measurements. The variance is assumed to have a quadratic relationship with the distance between the objects and the sensors, as mentioned in equation (3.16). The initial error covariance is assumed to be $P_0 = I$.

Measurements include the *x* and *y* positions of the vehicles. Therefore, the measurement noises can be found by calculating the difference between the position measurement values and the ground truth. Figure 4.5. shows the norm of measurement noises for both vehicles, as solid red and blue lines. Furthermore, the considered measurement of covariance R_k^{si} are also illustrated as dashed red and blue lines.

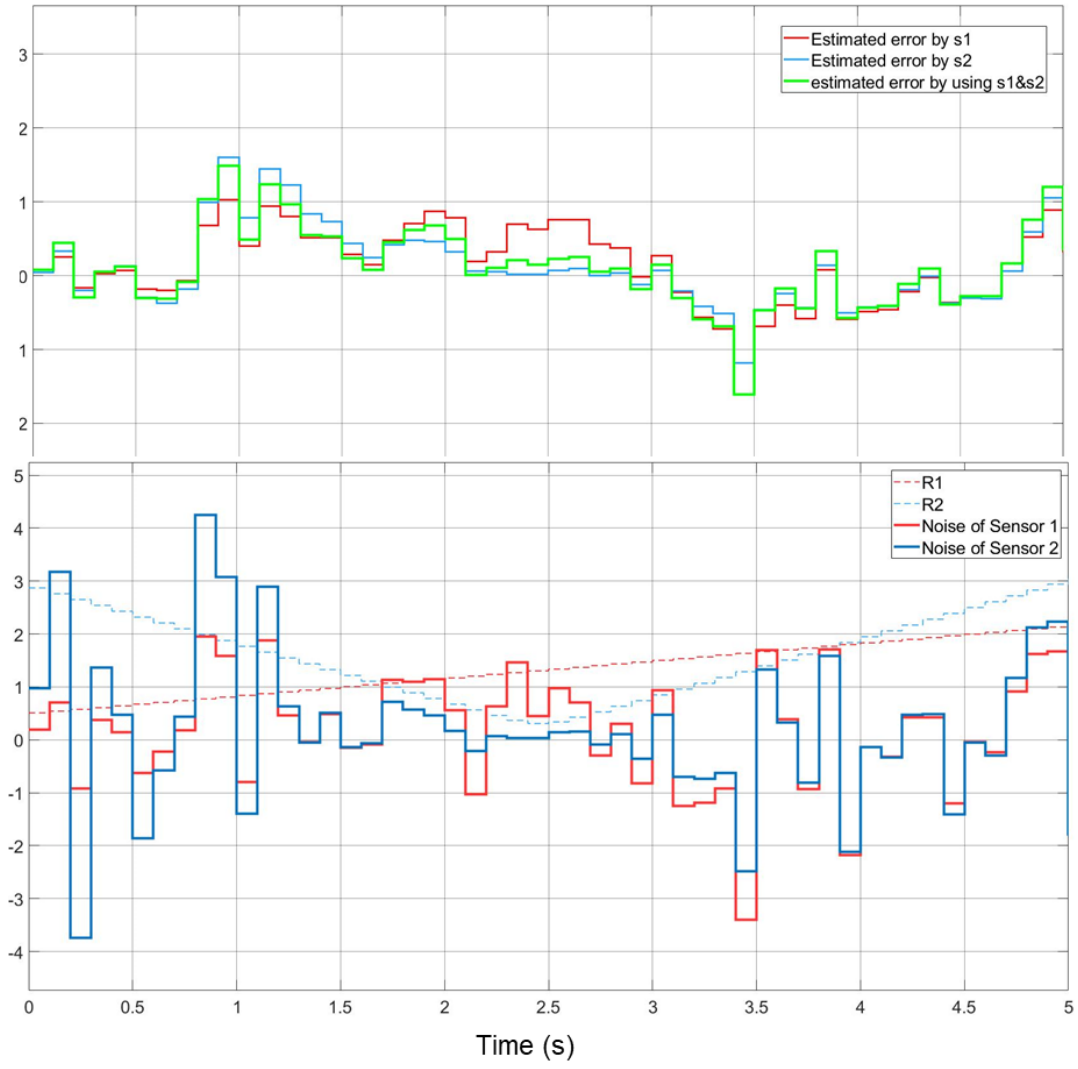


Figure 4.5: Noise and measurement covariance illustration

Using the cooperative PL system, described in Section 4.4, the fusion of the ISN data with onboard data can be carried out. Figure 4.5. the figure also shows the estimation error for different cases of using each sensor individually and also using them together. It is needed to mention that the estimation error is the difference between the estimated position of the vehicle and ground truth data of its location before adding any noise.

By fusing infrastructure data with onboard sensor measurement, the overall accuracy of the estimation has been improved. When the ego vehicle, the red vehicle, was at a far distance from the infrastructure-based camera, the estimated value is as in the case of using an onboard sensor alone. However, when the vehicle comes into the field of infrastructure sensors, the estimation error decreases. As it moves away from the ISN node, the estimation error goes up.

4.5.2 Cooperative PL Using One Sensor Node

In this configuration, one sensor node is used in addition to the onboard sensor. The configuration of the sensors can be seen in Figure 4.6. The onboard sensors including GPS and a camera are mounted on the ego vehicle (Vehicle 1 in the figure). There is also an ISN node mounted at a known location to provide the position of both the ego and the other vehicles. The initial locations of the vehicles are respectively at [-16 -3.7] and [-25 3.2] m. The ego vehicle is moving at a constant velocity of 20 m/s while the other one is moving faster at 50 m/s.

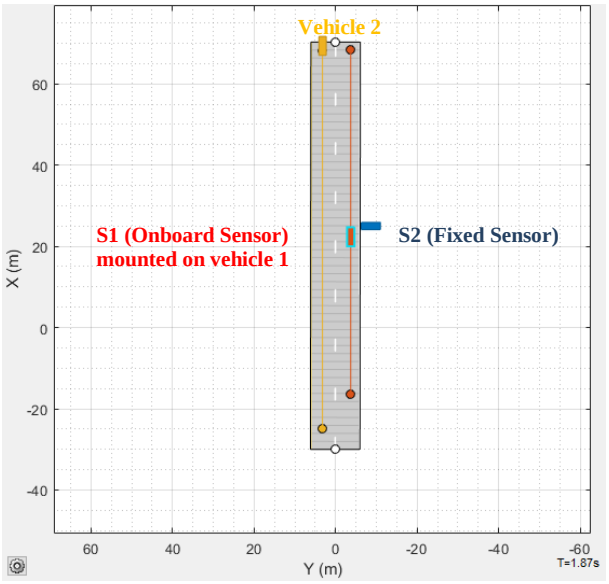


Figure 4.6: Sensor Configuration in the third scenario, simulated in MATLAB

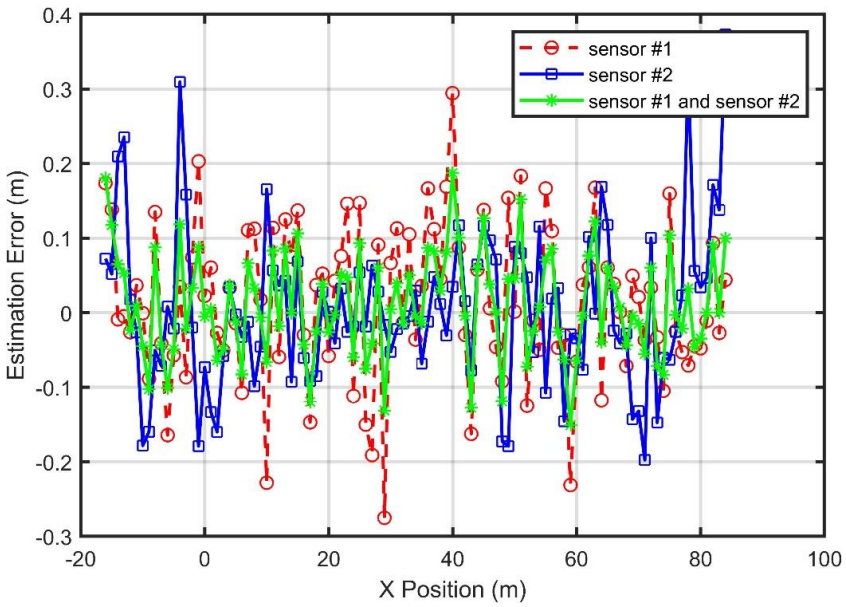


Figure 4.7: Estimation error for localizing the ego vehicle

Implementing the KCF for this scenario, the estimation error for the position of the ego vehicle, using both sets of sensors including s1 and s2 is illustrated in Figure 4.7 and compared with using each sensor individually.

The overall error statistics of the ego vehicle localization using different sensor nodes can be seen in Figure 4.8. Based on the results, the mean estimation error is acceptable in all cases. However, fusing both sensor nodes decreases the deviation of the estimation.

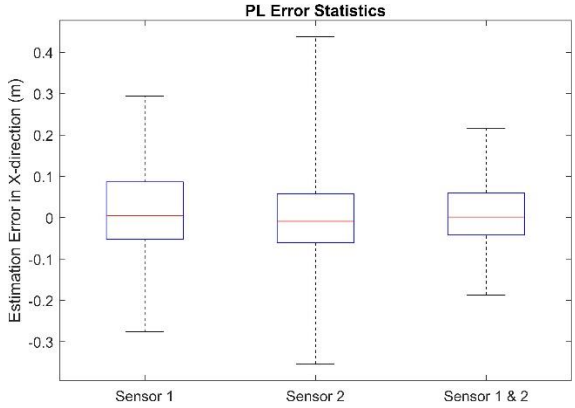


Figure 4.8: Estimation error statistics using different sensor nodes

Another important point to mention is that when the vehicle is close to the ISN node, it is more accurate. When the vehicle is far from the ISN node, the fusion algorithm relies more on the onboard sensor node which is more accurate at that time. Fusing both sensors' data, the overall estimation error is shown to be decreased.

The onboard sensor (GPS) has an almost uniform noise level performance during the maneuver, leading to a constant estimation performance (red line). On the other hand, a fixed sensor (s2) has a noise level that is dependent on the distance. This is the reason that at the beginning and also at the end of the maneuver where the distance of the object is relatively larger, the estimation in the case of using only the fixed sensor (blue line) lacks in performance. Incorporating both sets of sensor data in the KCF algorithm shows overall better performance, compared with using only one of them. Generally speaking, when the measurements of sensor 2 (fixed sensor) have lower noise levels than sensor 1's (onboard sensor), the localization algorithm weighs the sensor 2 data more, as can be seen in the x-position of 20 m. In far distances, where the fixed sensor lacks accuracy, the algorithm relies on the onboard sensor, like GPS, more.

A similar trend can be seen in localizing other vehicles. Figure 4.9 shows that using the fixed sensor in addition to the onboard sensors results in a better overall performance.

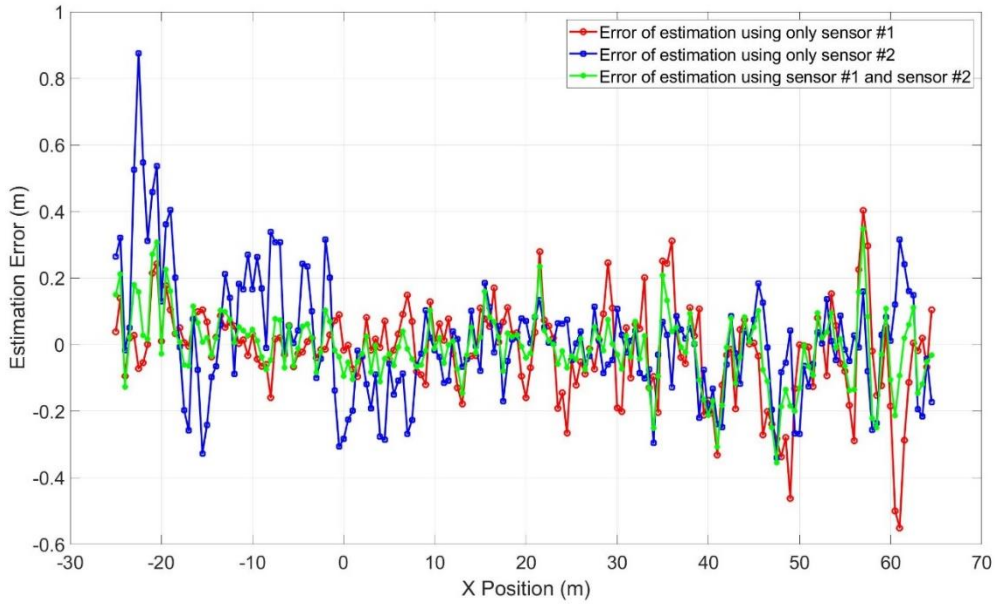


Figure 4.9. Estimation error of the other vehicle's position

Figure 4.10: Sensor locations in the second scenario, simulated in MATLAB

4.5.3 Cooperative PL Using Two Sensor Nodes

In this scenario, two ISN data are fused with onboard sensor data. Two similar sensors are located at different, but known locations, as illustrated in Figure 4.10. the distance between the sensor nodes is about 40 m. The vehicle starts moving from [-16 -3.7] m with a constant velocity of 30 m/s. The main goal of the configuration is to have a balance localization performance everywhere on the line because as the distance of the object from the sensor gets larger, the accuracy of the position estimation decreases. However, with such a configuration, the mentioned undesirable effect is avoided.

Figure 4.11 illustrates the result of the above scenario. The results satisfy the goal of having a balance localization performance during the motion, despite the distance-dependent accuracy of implemented sensors.

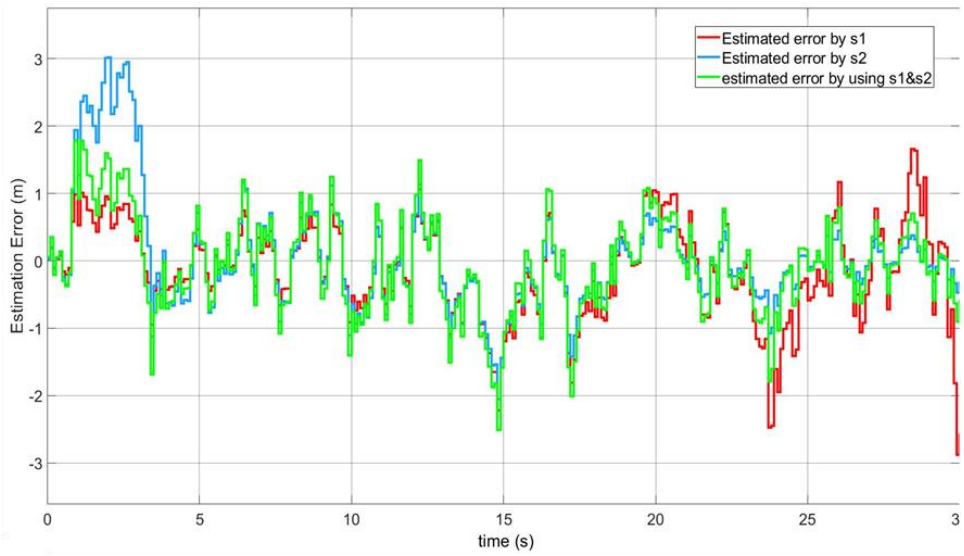


Figure 4.11: Estimation error of using two fixed sensors

Sensor 1 (shown as the red line) has a better accuracy at first since the vehicle is close to the sensor while sensor 2 lacks performance (the blue line) as a result of the far distance from the sensor. However, the implemented algorithm shows a better overall performance thanks to the covariance scaling method.

4.5.4 Constant Acceleration maneuver

In this scenario, the vehicles start moving at different velocities and perform content acceleration maneuvers. The initial locations of the vehicles are respectively [-16 -3.7] and [-25 3.2] m. The ego vehicle is stationary and then starts accelerating at a rate of 10 m/s^2 . The initial velocity of the other vehicle is 40 m/s and decelerates at the rate of 10 m/s^2. Figure 4.12 shows the longitudinal position of the vehicles in the global coordination.

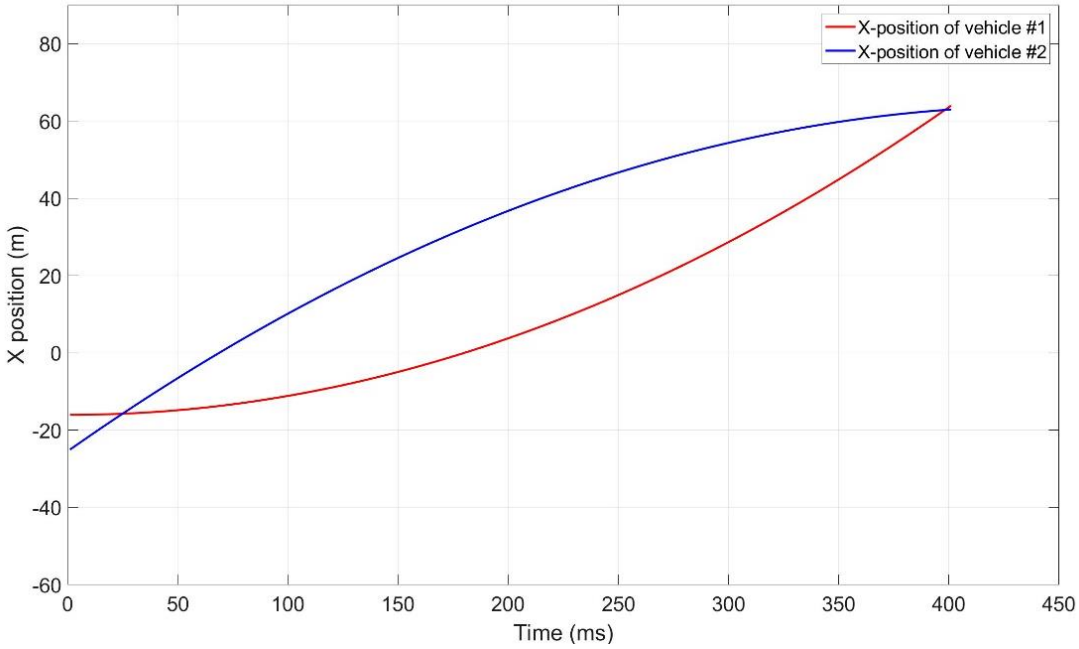


Figure 4.12: Longitudinal position of the vehicles in the constant acceleration scenario

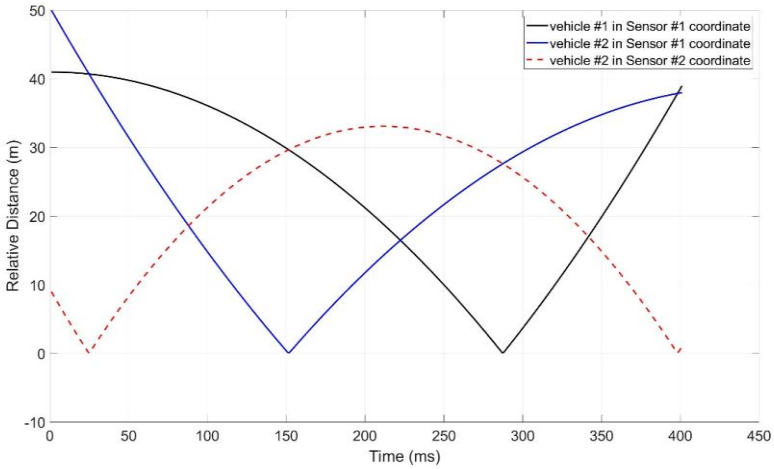


Figure 4.13: The relative distances of the vehicles in sensor coordinates

As mentioned before, the model takes the relative distance of the objects into account since the accuracy of sensors is dependent on the distance. The relative distances of each vehicle in sensor coordinates are illustrated in Figure 4.13.

Figure 4.14 illustrates the estimation error of positioning the vehicle using both sensors together or individually. The results show that based on the distance of the object from the sensor, the PL error of each sensor varies. However, using both sensor data, the overall estimation error of the proposed method is relatively smaller, even in harsh accelerating/decelerating maneuvers.

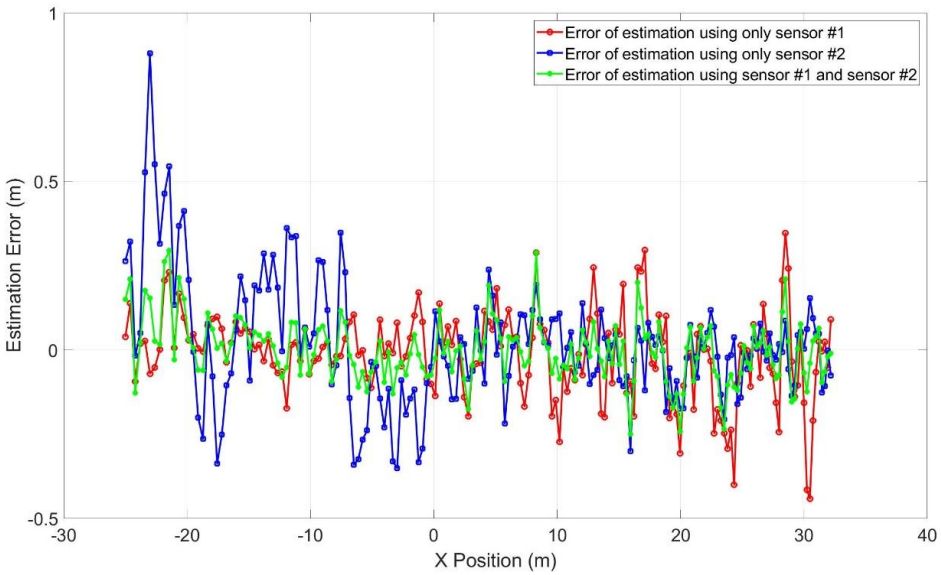


Figure 4.14: Estimation results of the vehicles

This is true even for fusing distance-dependent sensor data with another source of data with a relatively constant noise level like GPS. Estimation errors in the case of using only GPS data (red line) are approximately constant over the maneuvers. Fusing infrastructure data with GPS data can improve the overall performance in terms of PL accuracy.

4.5.5 Data Correspondence Simulation Results

In this section, the simulation results of implementing point set matching algorithms for the purpose of data correspondence are provided. The estimated detected objects by the ISN and the AV have different IDs. Point set matching algorithms are used to find the correspondence between the detected objects in different coordinates of the ISN and the AV.

For this purpose, the estimated position of 5 objects including the ego vehicle is supposed to be given at a specific time, as can be seen in Figure 4.15. The ego vehicle is represented by a dashed circle, while other objects are shown with solid circles.

The IDs of the objects are different in each coordinate. However, it can be seen that the correspondence between the ID can be found by detecting the relative geometry of the points with each other. Table 4.1 shows the true correspondence between the IDs of the objects in the ISN's (sensor 2) and the AV's (sensor 1) coordinates.

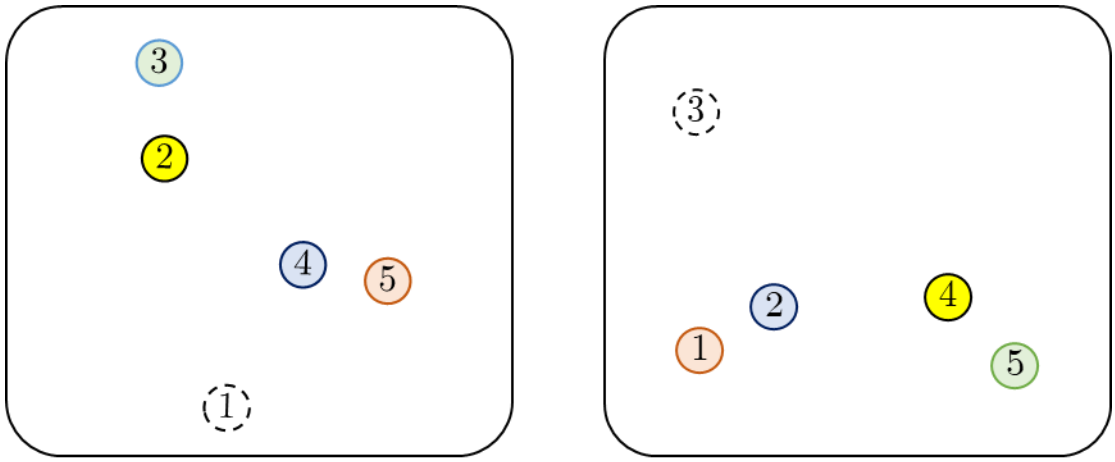


Figure 4.15: Estimated position of the objects including the ego vehicle in different coordinates

The ID of 1 in sensor 1 (AV's coordinates) corresponds to the AV itself. In the ISN's (sensor 2) coordinates, the AV is assigned with the ID of 3. The goal is to apply the point set matching algorithm provided in Algorithms 4.1 and 4.2 to find the correspondence between the IDs of the objects in onboard and global coordinates.

Figure 4.16 illustrates the estimated position of objects in the coordinates of sensors 1 and 2. It can be seen that there is a registration between sensors 1 and 2.

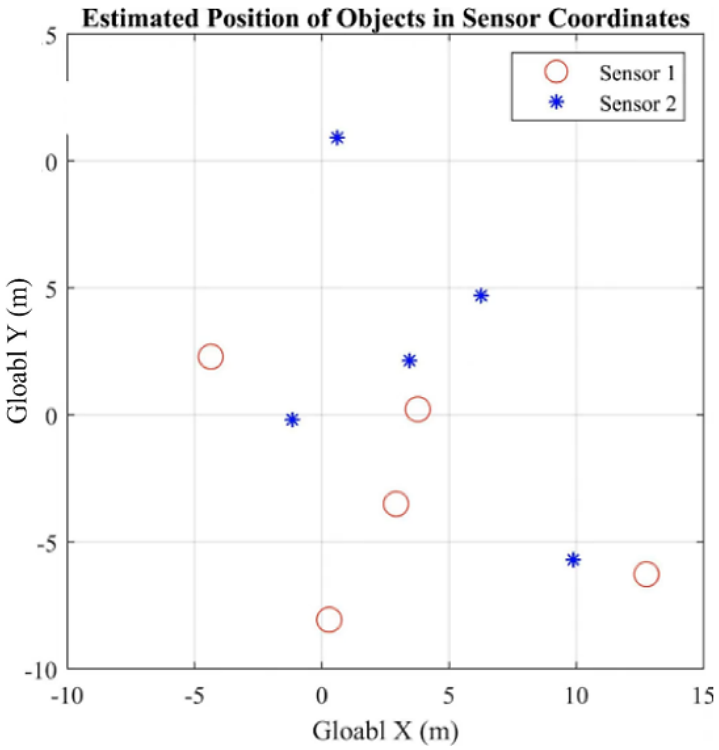


Figure 4.16: Estimated position of objects in coordinates of sensor 2 (ISN) and sensor 1 (ego vehicle)

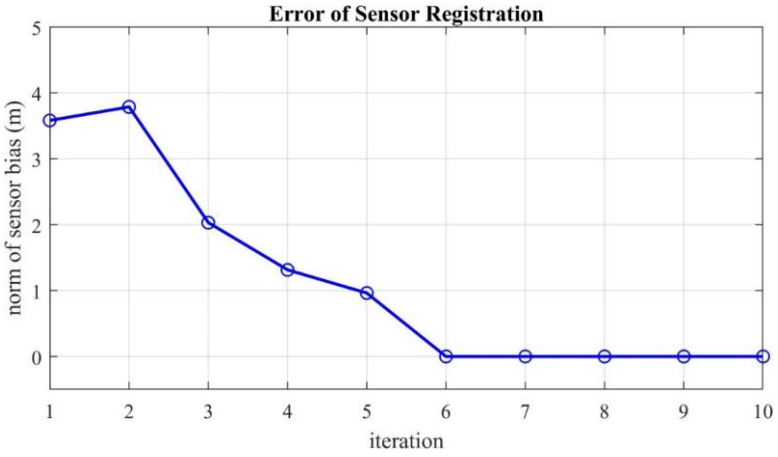


Figure 4.17: Sensor registration by using the ICP algorithm

Figure 4.17 depicts the estimation error of sensor registration by using the ICP algorithm. It shows that after only 6 iterations the sensor registration becomes zero, meaning that both sensors are aligned with each other after 6 iterations. Consequently, the correspondence between all the objects in two sensor coordinates is found.

Figure 4.18 shows the position of the objects after applying the ICP after 6 iterations. It can be seen that the sensor registration is successfully achieved. All the objects are estimated at the same position in the ISN's and the AV's coordinates. Consequently, the correspondence between the objects can easily be achieved.

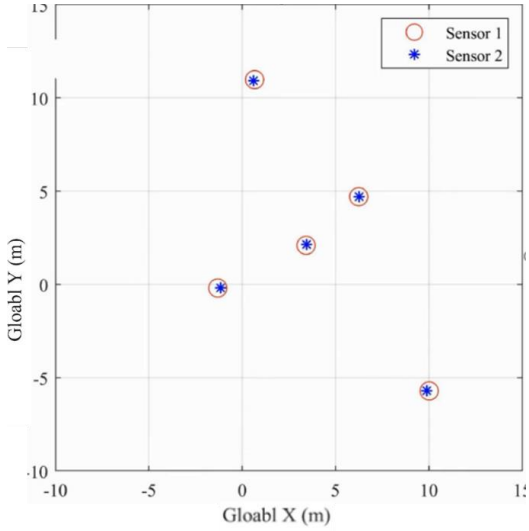


Figure 4.18: Estimated position of objects in different coordinates after convergence

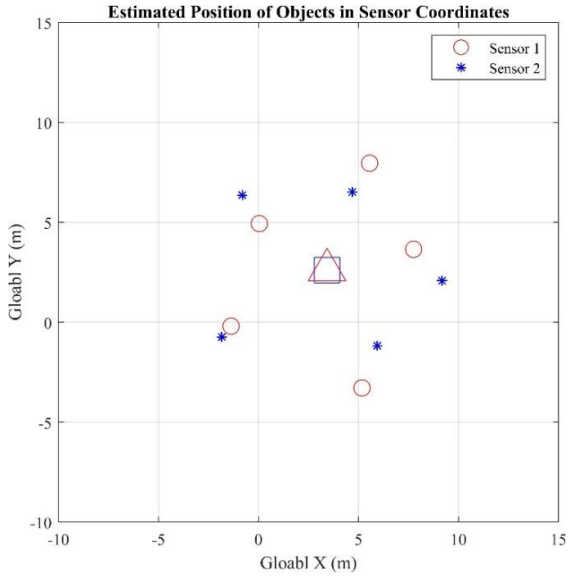


Figure 4.19: Example of the local minimum issue

Figure 4.19 shows one example in which the local minimum issue causes an error in the sensor registration. The square and the triangle represent the origin of the calculated sensor translation which are at the same location. It means that the ICP algorithm converges, however, the sensor registration is not correct.

Figure 4.20 shows the estimated heading angle of the AV in the scenario in which the local minimum issue occurred. The ICP algorithm estimated the heading angle of 26 degrees, while the ground truth is -47.257 degrees.

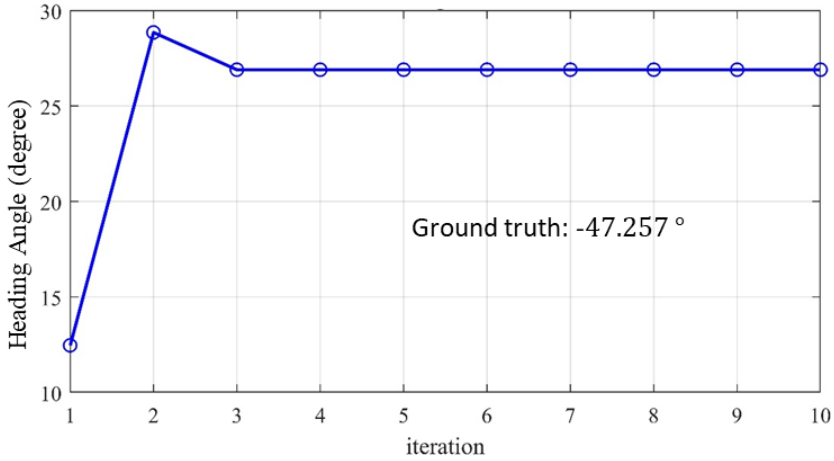


Figure 4.20: Estimated heading angle in a “local minimum issue” case

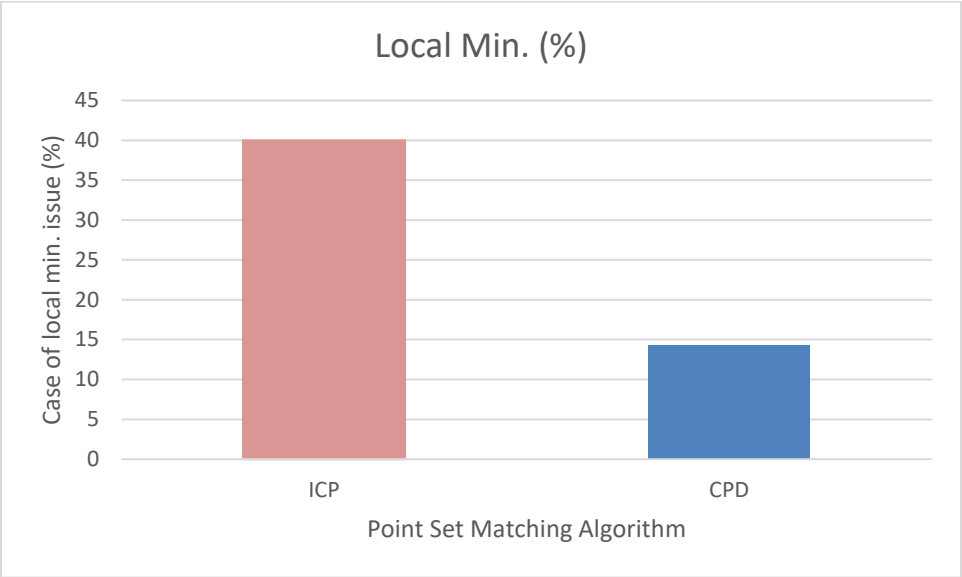


Figure 4.21: Statistic results of the local min. issue for two point set matching algorithms

As mentioned before in Chapter 2, the main issue of the ICP is that its performance is very sensitive to the initial guess. Therefore, it can converge to different local minimums with different initial guesses. This is referred to as the local minimum issue. That was the main motivation to present and implement the CPD approach in Algorithm 4.2.

Figure 4.21 illustrates the case in which the ICP and CPD suffer from the local minimum issue. Both algorithms start from the initial guess of $r_0^{MN} = I$ and $t_0^{MN} = [0 \quad 0]^T$ for the rotation and translation matrices, respectively.

CPD shows significantly better performance to address the local minimum issue in only 15% of cases, while the ICP suffers from the local minimum issue in 40% of the simulation cases. However, a 15% chance of a mistake in the correspondence of the objects is not negligible for the cooperative PL application.

Therefore, another assumption can be taken into account in the initial guess to improve the performance of the CPD algorithm. The assumption is to use the estimated rotation and translation matrices at the previous time step as the initial guess of the algorithm at the current time step.

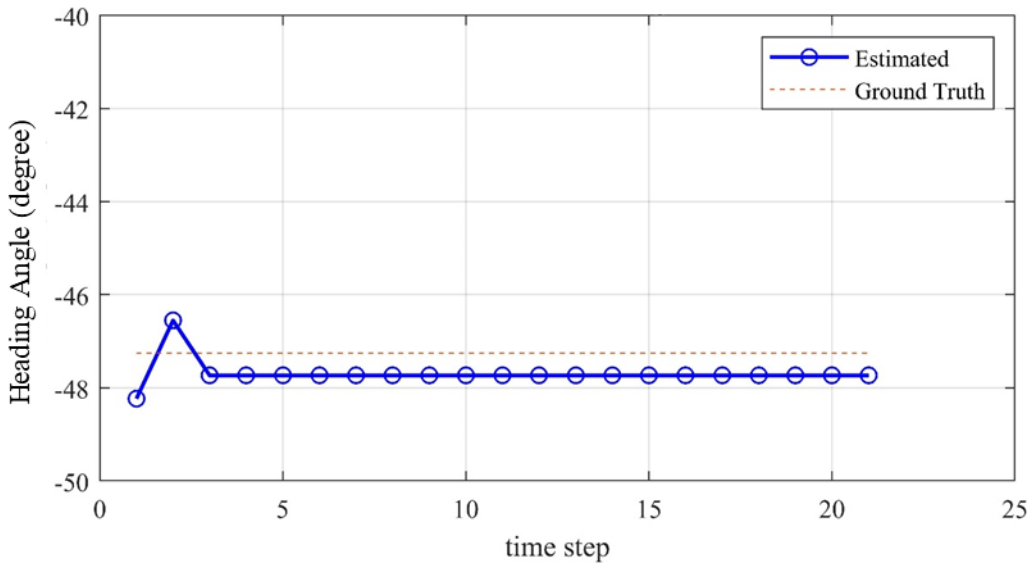


Figure 4.22: Estimated heading angle of the AV using the improved CPD algorithm

By considering this assumption,

$$r_{k,0}^{MN} = \hat{r}\left(\psi_{k-1}^{0}\right) \qquad (4.27)$$

$$t_{k,0}^{MN} = t_{k-1}^{MN}$$

Using this assumption, there is no local minimum issue reported in any time step of the simulated scenario. Besides, the estimated heading angle of the AV is successfully estimated, as can be seen in Figure 4.22.

4.6 Summary

This chapter serves as a contribution to the field of autonomous driving by presenting a framework for cooperative PL that addresses the issue of occlusion, which is commonly encountered in urban driving scenarios. Specifically, this work proposes a solution that fuses infrastructure-based sensor node data with onboard sensor data, thus improving the overall perception and localization of the autonomous vehicle.

In this chapter, the framework of the cooperative PL, its components, and the implementation results were presented. In the cooperative PL, the AV received the broadcast information package from the

sensor nodes. The received package included the estimated positions of the objects in the world coordinates.

The AV also had the estimated position of the objects in its own coordinates. The IDs of the objects were different in the AV's system and the received package. Therefore, the data correspondence algorithm was developed in Section 4.2 to find the correspondence between the AV's estimations and the sensor nodes' estimations.

After the correspondence was achieved, a data fusion algorithm could be performed. KCF was described and used as a data fusion algorithm to incorporate the infrastructure data into the localization of autonomous vehicles. The measurements of sensors were fused through KCF. This method requires less computation to provide the estimation, compared to the optimization-based methods. This is more crucial for the infrastructure-aided PL in which communication issues like delay exacerbate the real-time computation problem. Finally, the observability analysis and the stability criterion for the stable error dynamics of KCF were provided.

This chapter also presented the result of simulating different maneuvers in MATLAB/Simulink. The main purpose of implementing a fixed sensor in addition to onboard sensors in order to improve the overall localization performance was achieved, as the results showed. Despite the distance-dependent accuracy of cameras and Lidar, the KCF incorporated the sensor data in a way that the estimation relied more on the sensor which had a less noise level at that time.

Chapter 5
Experimental Studies

5.1 Introduction

Evaluating the algorithm's performance based on real scenarios is another goal of this thesis. Using the experimental data provide insights into the strength and also limitations of the algorithm. This chapter provides the experimental results for implementing the developed algorithm on real data in real scenarios.

The ISN units include two cameras and a Lidar with the same FOV of the two cameras, a computing module, and a modem. The node was mounted on a scissor lift, as can be seen in Figure 5.1. The ISN was then lifted to 10 m to provide an almost bird's-eye view.

Lidar used for the experimental data is a 32-beam RS-Helios 1615 which offers a 31° vertical FoV and uniform 1° vertical resolution. It also has 360° horizontal FoV with a 0.2° horizontal resolution.

Furthermore, The Basler daA1920-160uc is used for both cameras which has 0.05° (1920 pixels) and 0.03° (1200 pixels) horizontal and vertical resolution, respectively. Besides, their FoVs are $[90, 76]°$.

Figure 5.1: ISN, including two cameras and a Lidar, mounted on a scissor lift

Figure 5.2: WATonoBus, an autonomous shuttle bus with Vehicle to Infrastructure (V2I) connectivity

The autonomous shuttle (WATonoBus) bus can receive the information broadcast by the sensor node. the sensor node is depicted by the red circle in Figure 5.2 WATonoBus is equipped with V2I connectivity so it can receive the information and fuse it with the onboard sensor data. This experiment will evaluate the performance of the approach in improving the safety of the AV in urban intersections. WATonoBus along with another vehicle and a pedestrian performed an intersection scenario as can be seen in Figure 5.3.

Figure 5.3: Intersection scenario with WATonoBus, another vehicle, and a pedestrian

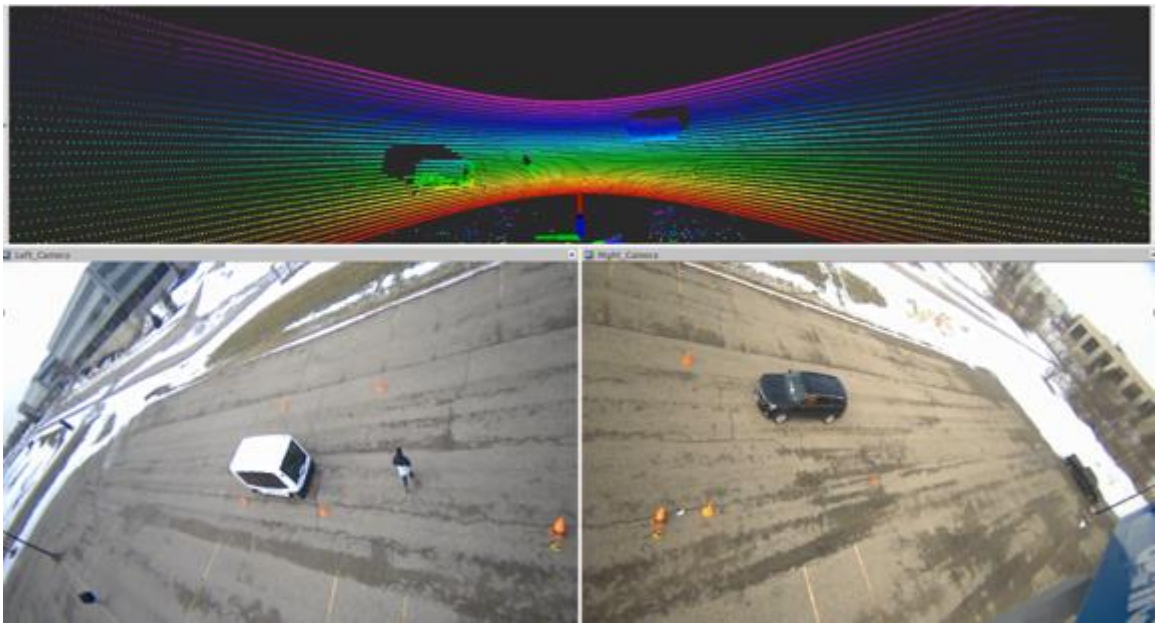


Figure 5.4: Covered area by the ISN's cameras and Lidar

Figure 5.4 shows the covered area by the ISN. The mounted ISN can cover about 40m along the road with a minimum of 10 m across. Figure 5.5 illustrates the image frames of the cameras before and after the image stitching. It is worthwhile to mention that the image frames shown in Figure 5.5 are cropped and they are not the whole frame of the cameras.

(a) Before image snitching (b) After image snitching

Figure 5.5: Image frames of the ISN's cameras

5.2 Experimental Results

The scenario of interest in this section simulates the urban intersection situation where the AV is surrounded by other dynamic objects including other vehicles and pedestrians. In these situations, the AV has a limited FOV since its onboard sensors are exposed to occlusion.

Figure 5.6 illustrates the estimated trajectories of all dynamic objects by the ISN. The objects include WATonoBus, a pedestrian, and a vehicle which are respectively, referred to as objects 1 to 3. In this scenario, the dynamic objects are all in the ISN's FOV which is located at the origin. The measurement is recorded in $10\ Hz$, therefore, the time step is considered $100\ ms$.

As can be seen in Figure 5.6, WATonoBus, represented in blue, starts from the initial position of $[4.4, 20.5]$ m which is a relatively far distance from the ISN (located at $[0, 0]$), and then keeps getting closer to it. Another vehicle, shown in red, moves from the other side of the ISN starting from the initial position of $[9.2, -14.1]$ m.

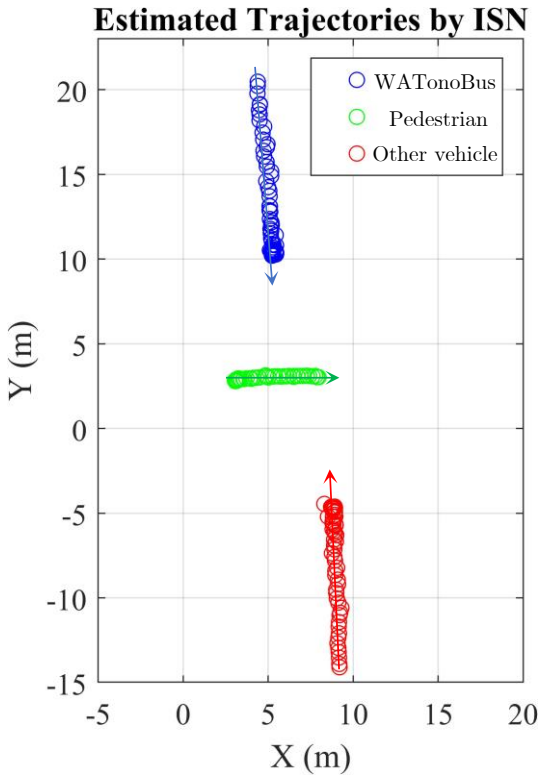


Figure 5.6: Estimated trajectories of dynamic objects by the ISN

At first, the pedestrian is located at position $[3, 2.8]$ m and stays there with little movement. Both vehicles then stop for the pedestrian to cross in front of them.

Figure 5.7 shows the estimated position of the pedestrian and the other vehicle by WATonoBus. WATonoBus estimated the position of the objects in its own coordinate. For the sake of visualization, the x and y-axis of the ISN's and WATonoBus' coordinates are assumed to be the same for the estimated positions to be consistent for the diagrams.

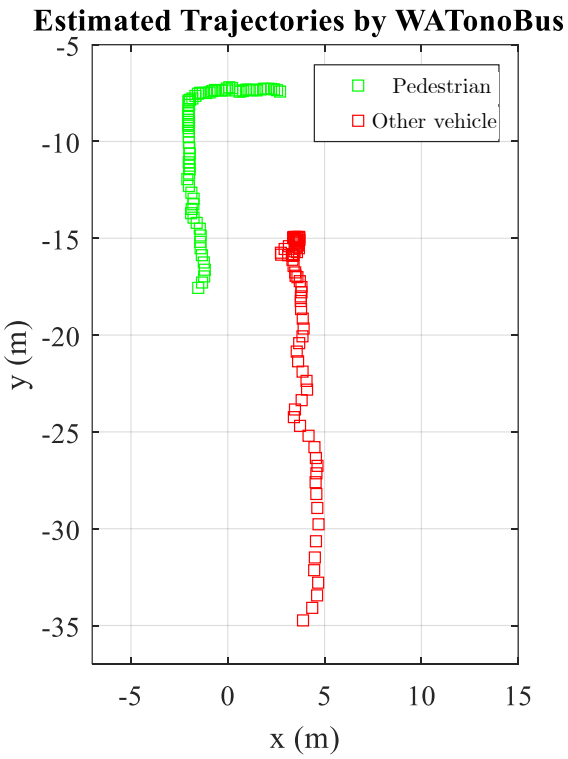
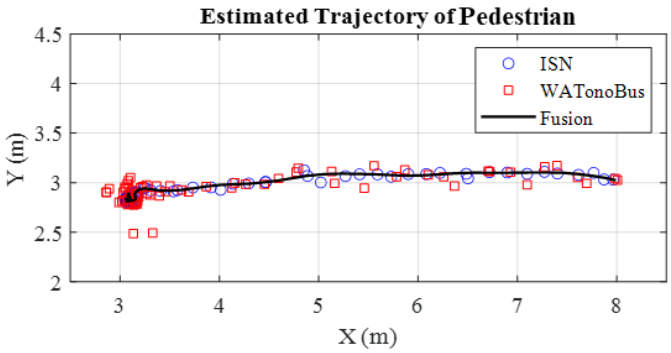


Figure 5.7: Estimated trajectories of dynamic objects by WATonoBus

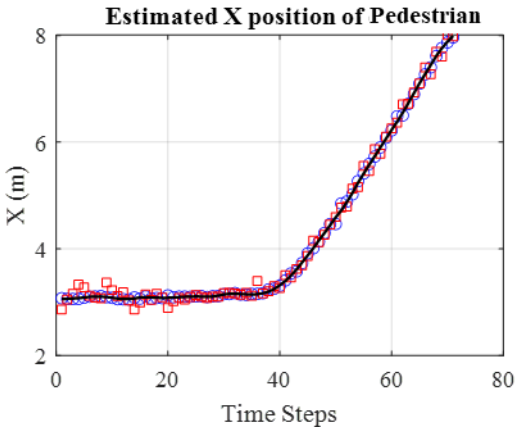
After the estimation of the dynamic objects by the ISN, it broadcasts the estimated positions of the objects. WATonoBus then receives the estimation information broadcast by the ISN. Both ISN and WATonoBus estimated the positions of dynamic objects in their own coordinates. Therefore, WATonoBus ran the correspondence algorithm to identify its estimated position among other ones in the received information by using Algorithm 4.2. When the estimated position of WATonoBus is found, other objects' estimated positions can be calculated in the onboard coordinates. All estimated values either by the ISN or by WATonoBus are in onboard coordinates and ready for the next step which is

fusing them in the cooperative PL by using Algorithm 4.3. WATonoBus then performs the cooperative PL using its onboard estimation and the ISN's. The cooperative PL is carried out through the proposed algorithm in Chapter 4. The reliability of each node (ISNs or WATonoBus) is taken into account based on the scaled measurement covariance method which is considered dependent on the distance of each object from the nodes.

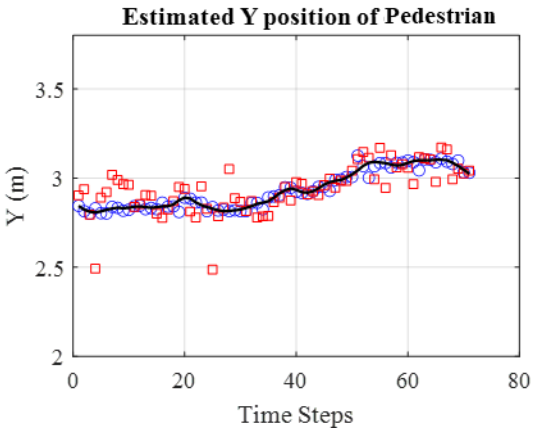
Figure 5.8 depicts the estimated trajectory of the pedestrian (object 2) using the ISN, WATonoBus, and the fusion of both nodes. The estimation results of WATonoBus, represented by red squares, are not reliable at first when the pedestrian is not moving. This is because, during the first time steps, WATonoBus is located around $[4.4, 20.5]$ which is far from the pedestrian which is located around $[3, 2.8]$ m.



(a) Estimated trajectory



(b) Estimated X position

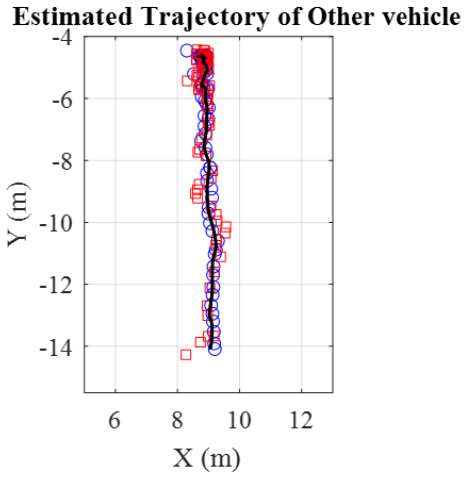


(c) Estimated Y position

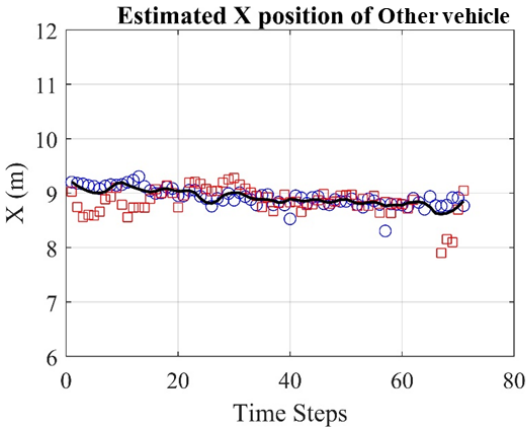
Figure 5.8: Estimated position of object 2 by the ISN and WATonoBus

However, the pedestrian, located at [3, 2.8], is near the ISN in the XY plane and therefore the ISN's estimation is more accurate and consequently more reliable. That is why the fusion results (black line) rely more on the ISN estimation at the first thirty time steps.

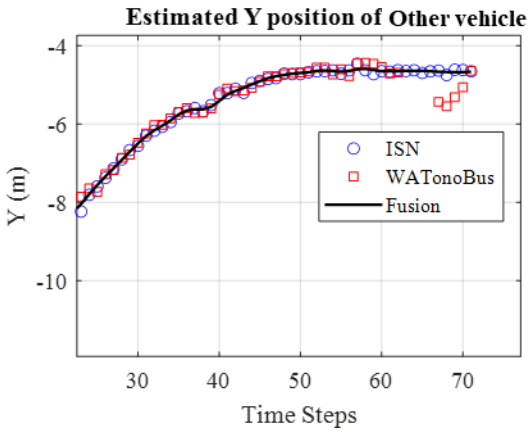
Figure 5.9 shows the estimated trajectory of the other vehicle by the ISN, WATonoBus, and the fusion of both nodes. Similarly, the red squares illustrate the estimated position of the other vehicle (object 3) by the onboard sensor of WATonoBus.



(a) Estimated trajectory



(b) Estimated X position



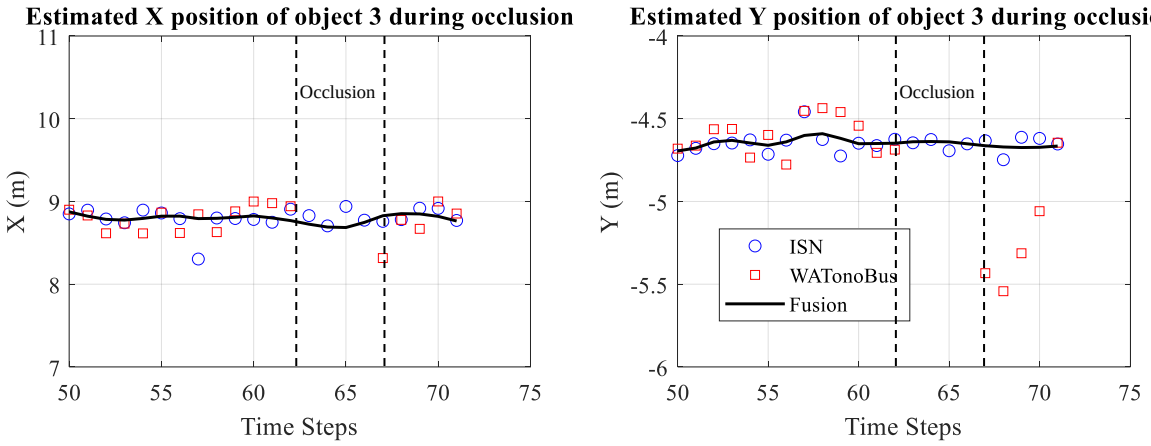
(c) Estimated Y position

Figure 5.9: Estimated position of the other vehicle by the ISN and WATonoBus

The other vehicle and WATonoBus start from a distance of almost 40 m from each other. Since the distance between the other vehicle from WATonoBus is even longer than with the pedestrian, the accuracy of the onboard estimation in the first half of the scenario (the first 35 time steps) is not acceptable.

Using distance-dependent measurement covariance for the cooperative PL, the fusion estimation puts more weight on the closer node to the object. As the other vehicle and WATonoBus get closer to each other, the measurement covariance of the onboard node (WATonoBus) decreases, and consequently, its weight in the consensus increases.

However, starting from the 60th time step, the WATonoBus estimation deviates from the ISN's estimation, as can be seen in Figure 5.10. This is because the pedestrian crosses in front of WATonoBus at that moment, obstructing the onboard sensor. As a result, the other vehicle cannot be detected by WATonoBus and there is no estimation regarding the other vehicle in the onboard system. In other words, WATonoBus could not see the other vehicle for a few time steps due to occlusion. This happens frequently in applications for urban driving, particularly at intersections when certain dynamic objects occlude other objects from AVs' onboard sensors.

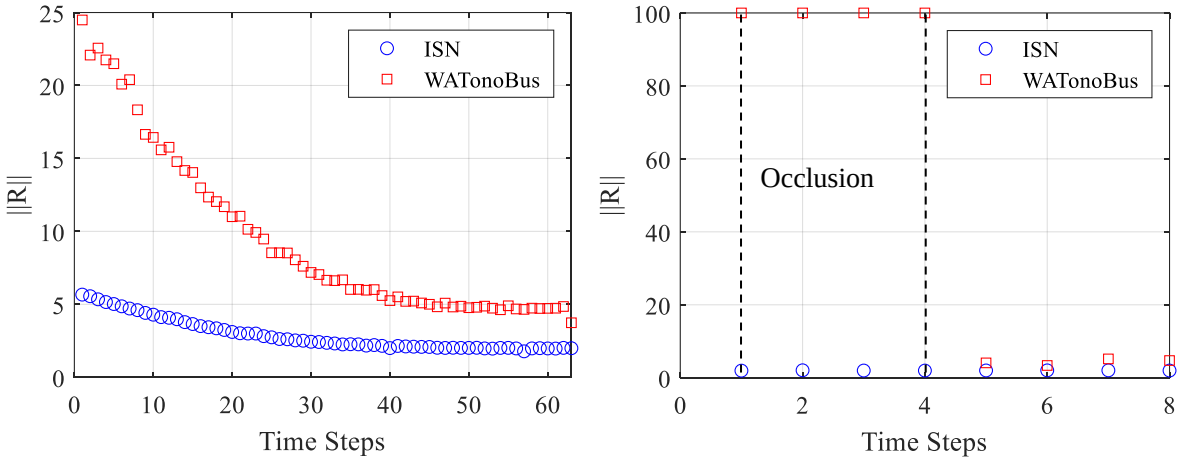


(a) Estimated X position (b) Estimated Y position

Figure 5.10: Estimated position of the other vehicle during occlusion

This issue can be dealt with in the proposed method in Section 4.4. by tuning the measurement covariance corresponding to the missing objects. To elaborate, when WATonoBus receives the position information of the objects, it performs the data association algorithm to correspond the estimated position by the ISN with the estimated positions of the objects by itself. In cases where the objects detected by the ISN are more than the detected objects by WATonoBus, imaginary measurements with large measurement covariance are considered in onboard coordinates (WATonoBus) to represent the occluded objects.

The measurement covariance of the other vehicle can be seen in Figure 5.11. The measurement covariance of the ISN is smaller since object 3 is always closer to the ISN than to WATonoBus. During the occlusion, a large number is assigned to the measurement covariance of the other vehicle by WATonoBus. After WATonoBus detects the other vehicle, the other vehicle is not missing in the estimation result by the onboard system. Thus, the covariance measurement follows the standard procedure which is dependent on the distance of the object from the node.



(a) Before occlusion (b) Including Occlusion

Figure 5.11: Measurement covariance regarding the other vehicle

As a result, the cooperative PL algorithm can be still applied to fuse the ISN and onboard estimations. Since a very large number is considered as the measurement covariance of the undetected object, the fusion result will not rely on the node that does not detect an object due to occlusion and is almost entirely dependent on more reliable nodes.

Therefore, the developed cooperative method using the ISN allows the WATonoBus to overcome occlusion in the urban intersection. In other words, using the proposed algorithm, it has a full position awareness of its surroundings, including any occluded objects that cannot be detected by its onboard sensors due to occlusion.

5.3 Summary

In this chapter, the experimental setup and real-time implementation of the developed algorithm were presented. The chapter utilized WATonoBus and an infrastructure-based sensor node mounted on a scissor lift to simulate an intersection scenario. The performance of the developed approach in real-time implementation was evaluated using the real data obtained from these two sensor nodes.

The chapter highlights the issue of occlusion in autonomous driving, which can cause dangerous situations and even collisions when objects are occluded from the onboard sensors. To address this issue, the proposed approach incorporates another source of information that does not suffer from occlusion, which is the position information of the occluded vehicles provided by the infrastructure-based sensor node. By fusing this information with onboard sensor data, the proposed approach enables the autonomous vehicle to be aware of occluded objects and improves overall perception and localization performance. The results of the experiments demonstrate the effectiveness of the proposed approach in real-world scenarios.

Chapter 6
Conclusions and Future Work

6.1 Conclusions

This thesis provided a deeper insight into the application of ISN nodes in cooperative PL for autonomous driving. Cooperative PL can be used in improving added safety to autonomous vehicles in complex urban driving such as intersections and also in autonomous fleet operations. In the former application, ISN data is cooperatively incorporated into the vehicle-based PL. This can improve the accuracy and reliability of the PL system, particularly in urban intersections where occlusion is common. On the other hand, in the latter application, the fleet vehicles are directly controlled via global PL and control systems.

Needless to say that the proposed framework can also be used for fusing any distance-dependent sensor data with onboard sensor data with a relatively constant noise level. Fusing the broadcast information by the ISNs with onboard sensor data can improve the overall localization performance and provide safer autonomy in complex urban driving environments such as intersections as well as for autonomous fleet operations

The motion compensation module was successfully used to estimate the sensor node's position and orientation in the presence of external disturbances such as wind. Although camera/Lidar-based pose estimation could estimate the position and orientation of the sensor node, its performance, especially in the x and y positions could be improved. The simulation results showed that the maximum estimation error of 8 cm. Furthermore, the approach was capable to estimate the orientation of the sensor node with negligible error. The reason was that the small level of deviation in the orientation could cause a large deviation in the position deviation due to the height of the sensor node.

Incorporating the dynamic motion of the light post into the motion compensation module successfully improved the performance of the pose estimation. Since an unknown external disturbance was inserted by the wind, an unknown input observer was implemented to estimate the position and orientation of the sensor node. The observer matrices were designed to make the state estimation error approaches zero asymptotically, regardless of the unknown input. It was shown that using the motion model of the light post by applying the unknown input observer was capable to improve the pose estimation. The simulation results showed a negligible estimation error.

The distance-dependent nature of the infrastructure-based sensors was addressed by the measurement covariance scaling approach. In this method, the measurement covariance was considered a quadratic function of the distance of the object from the sensor node. This assumption was consistent with the dependency of the density of pixels and point clouds to distance, mentioned in the literature and also shown by simulations. Applying this method, the cooperative PL algorithm could rely more on the sensor node that provides more accurate estimations at each time step.

The developed data correspondence method could link the estimations of the ISNs and AVs. The data association was formulated as a point set matching problem in which the estimated position objects were considered as point sets. The performance of the typical point set matching algorithm was very sensitive to the initial guess, which was undesirable. Therefore, the rigid registration was used as the point set matching algorithm. The results showed that the correspondence between the objects in ISNs' coordinates and the ego vehicle's coordinates could be found successfully.

Applying the proposed cooperative on experimental data showed that fusing ISN data with onboard sensor data could improve the reliability of the Pl algorithm in occluded situations. The results showed that due to occlusion, the WATonoBus failed to detect all dynamic objects in some time steps. ISN, however, was able to detect and estimate the positions of all the objects and broadcast the position information of all objects including the occluded objects to the WATonoBus so it could use it to cover its limited FoV. Thanks to the cooperative PL method, the WATonoBus could have a position awareness of their surrounding regardless of the occlusion limitation in urban intersections.

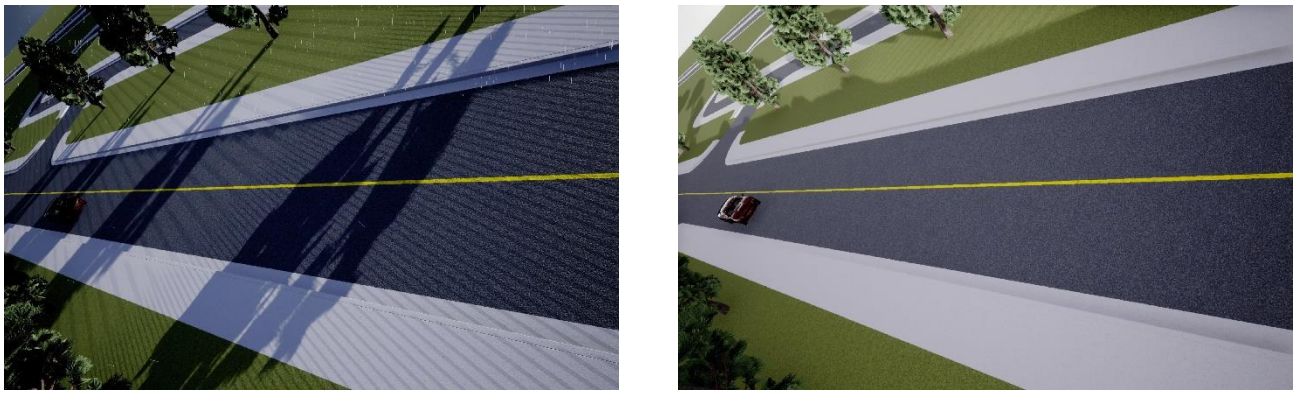
6.2 Future Work

The work done in this thesis can be further extended in the following directions:

- **Improving the sensor uncertainty model and covariance estimation**

Implementing a linear correlation between the measurement covariance and the distance showed that the concept of taking the uncertainty model into account can be used to have a balanced localization performance. Thus far, only the distance was used to adjust the sensor covariance. Other factors such as lighting and weather conditions also impact the sensor accuracy. Figure 6.1 illustrates two virtual images from the same scene but in different weather and lighting conditions. As seen in the figures, the accuracy of PL using each of the images will be different. This especially when the ISN data are fused with the vehicle onboard sensor data become more important as any fusion method relies on the data

reliability information. Utilizing a feature-based model that can predict the sensor uncertainty is considered one of the future tasks.



(a) Rainy weather (b) Different light condition

Figure 6.1: Images captured by the simulated camera in (a) rainy weather and (b) a different light condition

- **Improving the fusion method**

As mentioned before, the main goal of this thesis is to improve PL performance by using ISNs. Fusing infrastructure data with onboard data was done in the context of the Klaman-consensus filter. However, it can be extended to other techniques. To elaborate, the correlation between different sensor nodes can be taken into account, while this correlation is difficult to determine in general distributed networks [13] resulting in non-optimal local. Another approach can be using an inverse covariance filter, as proposed in the information consensus filter (ICF) [96]. This method works with the effect of the correlation instead of solving the correlation problem between the local estimates, and sensor nodes. Other techniques like Covariance Intersection (CI) can deal with this problem by considering bounding limits for the cross-covariance.

Another approach is to use factor graphs as a fusion algorithm instead of KCF. Factor graphs can model complex systems with nonlinearities and uncertainties. They provide a flexible and intuitive framework for modeling a wide range of sensor measurements and vehicle dynamics. They also allow for the fusion of heterogeneous sensor data and can handle missing or noisy data more robustly.

Any AV that receives the broadcast information from the ISN node, uses the data association algorithm to find its related position to fuse it with the onboard sensor data. Therefore, the input of the data association algorithm will be the broadcast information by the ISN data. The output of the

algorithm is the assigned data for each object including the ego vehicle. The assigned data will be the input of the fusion algorithm.

References

[1] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Inf. Fusion*, vol. 14, no. 1, pp. 28–44, 2013, doi: 10.1016/j.inffus.2011.08.001.

[2] E. Waltz and J. Llinas, *Multisensor data fusion*, vol. 685. Artech house Boston, 1990.

[3] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.

[4] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, 1997, vol. 3068, pp. 182–193.

[5] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*. Artech House, 2013.

[6] R. P. S. Mahler, "' Statistics 101' for multisensor, multitarget data fusion," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 53–64, 2004.

[7] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1696–1710, 2006.

[8] F. Castanedo, "A review of data fusion techniques," *Sci. World J.*, vol. 2013, 2013, doi: 10.1155/2013/704504.

[9] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[10] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused-track covariance," *IEEE Trans. Aerosp. Electron. Syst.*, no. 6, pp. 803–805, 1986.

[11] X. R. Li, Y. Zhu, J. Wang, and C. Han, "Optimal Linear Estimation Fusion - Part I: Unified Fusion Rules," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2192-2208+2323, 2003, doi: 10.1109/TIT.2003.815774.

[12] J. Ajgl, M. Simandl, and J. Dunik, "Millman ' s Formula in Data Fusion," *10th Int. PhD Work. Young Gener. Viewp.*, 2009.

[13] L. P. Yan, B. S. Liu, and D. H. Zhou, "The modeling and estimation of asynchronous multirate multisensor dynamic systems," *Aerosp. Sci. Technol.*, vol. 10, no. 1, pp. 63–71, 2006.

[14] S. McLaughlin, V. Krishnamurthy, and R. J. Evans, “Bayesian network model for data incest in a distributed sensor network,” in *Proceedings of the 7th International Conference on Information Fusion*, 2004, vol. 1, pp. 606–613.

[15] L. Chen, P. O. Arambel, and R. K. Mehra, “Fusion under unknown correlation-covariance intersection as a special case,” in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*, 2002, vol. 2, pp. 905–912.

[16] S. J. Julier and J. K. Uhlmann, “Non-divergent estimation algorithm in the presence of unknown correlations,” *Proc. Am. Control Conf.*, vol. 4, pp. 2369–2373, 1997, doi: 10.1109/acc.1997.609105.

[17] A. R. Benaskeur, “Consistent fusion of correlated data sources,” in *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, 2002, vol. 4, pp. 2652–2656.

[18] Y. Zhou and J. Li, “Data fusion of unknown correlations using internal ellipsoidal approximation,” *IFAC Proc. Vol.*, vol. 41, no. 2, pp. 2856–2860, 2008.

[19] S. Sun, H. Lin, J. Ma, and X. Li, “Multi-sensor distributed fusion estimation with applications in networked systems: A review paper,” *Inf. Fusion*, vol. 38, pp. 122–134, 2017, doi: 10.1016/j.inffus.2017.03.006.

[20] D. C. Carmer and L. M. Peterson, “Laser radar in robotics,” *Proc. IEEE*, vol. 84, no. 2, pp. 299–320, 1996.

[21] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system,” *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, 1992.

[22] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor location sensing using active RFID,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, 2003, pp. 407–415.

[23] O. C. Barawid Jr, A. Mizushima, K. Ishii, and N. Noguchi, “Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application,” *Biosyst. Eng.*, vol. 96, no. 2, pp. 139–149, 2007.

[24] J. R. Guerrieri *et al.*, “RFID-assisted indoor localization and communication for first responders,” in *2006 First European Conference on Antennas and Propagation*, 2006, pp. 1–

6.

[25] I. H. Li, M. C. Chen, W. Y. Wang, S. F. Su, and T. W. Lai, “Mobile robot self-localization system using single webcam distance measurement technology in indoor environments,” *Sensors (Switzerland)*, vol. 14, no. 2, pp. 2089–2109, 2014, doi: 10.3390/s140202089.

[26] T. Sato, M. Sugimoto, and H. Hashizume, “An extension method of phase accordance method for accurate ultrasonic localization of moving node,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 92, no. 12, pp. 953–963, 2009.

[27] A. R. Rababaah, “MOBILE ROBOT LOCALIZATION VIA EFFICIENT CALIBRATION TECHNIQUE OF A FIXED REMOTE CAMERA,” *Int. J.*, vol. 24, no. 1, pp. 23–32, 2012.

[28] I. Rekleitis, D. Meger, and G. Dudek, “Simultaneous planning, localization, and mapping in a camera sensor network,” *Rob. Auton. Syst.*, vol. 54, no. 11, pp. 921–932, 2006, doi: 10.1016/j.robot.2006.05.009.

[29] J. H. Shim and Y. I. Cho, “A mobile robot localization via indoor fixed remote surveillance cameras,” *Sensors (Switzerland)*, vol. 16, no. 2, pp. 1–13, 2016, doi: 10.3390/s16020195.

[30] P. Chakravarty and R. Jarvis, “External cameras & a mobile robot: A collaborative surveillance system,” *Proc. 2009 Australas. Conf. Robot. Autom. ACRA 2009*, 2009.

[31] C. C. Hsu, M. C. Lu, and K. W. Chin, “Distance measurement based on pixel variation of CCD images,” *ICARA 2009 - Proc. 4th Int. Conf. Auton. Robot. Agents*, pp. 324–329, 2009, doi: 10.1109/ICARA.2000.4803985.

[32] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, “Mapping and localization with RFID technology,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, 2004, vol. 1, pp. 1015–1020.

[33] G. Baatar, M. Eichhorn, and C. Ament, “Precise indoor localization of multiple mobile robots with adaptive sensor fusion using odometry and vision data,” *IFAC Proc. Vol.*, vol. 19, pp. 7182–7189, 2014, doi: 10.3182/20140824-6-za-1003.02345.

[34] A. Idrees, Z. Iqbal, and M. Ishfaq, “An efficient indoor navigation technique to find optimal route for blinds using QR codes,” in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 690–695.

[35] P. Furgale, J. Anderson, and J. Baltes, “Real-time vision-based pattern tracking without predefined colors,” in *Proceedings of the Third International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS)*, 2005.

[36] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, “Indoor localization of mobile robots through QR code detection and dead reckoning data fusion,” *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 6, pp. 2588–2599, 2017, doi: 10.1109/TMECH.2017.2762598.

[37] W. Xiao-Long, W. Chun-Fu, L. Guo-Dong, and C. Qing-Xie, “A robot navigation method based on RFID and QR code in the warehouse,” *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, pp. 7837–7840, 2017, doi: 10.1109/CAC.2017.8244199.

[38] A. C. Rice, A. R. Beresford, and R. K. Harle, “Cantag: an open source software toolkit for designing and deploying marker-based vision systems,” in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM’06)*, 2006, pp. 10-pp.

[39] M. S. Ahmed, R. Saatchi, and F. Caparrelli, “Vision based object recognition and localisation by a wireless connected distributed robotic systems,” *Electron. Lett. Comput. Vis. Image Anal.*, vol. 11, no. 1, pp. 54–67, 2012, doi: 10.5565/rev/elcvia.486.

[40] S. Lee, G. S. Tewolde, J. Lim, and J. Kwon, “Vision based localization for multiple mobile robots using low-cost vision sensor,” *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2015-June, pp. 280–285, 2015, doi: 10.1109/EIT.2015.7293353.

[41] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, “A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, 2018, doi: 10.1109/JIOT.2018.2812300.

[42] A. Amini, R. M. Vaghefi, J. M. De La Garza, and R. M. Buehrer, “Improving GPS-based vehicle positioning for Intelligent Transportation Systems,” *IEEE Intell. Veh. Symp. Proc.*, pp. 1023–1029, 2014, doi: 10.1109/IVS.2014.6856592.

[43] R. Parker and S. Valaee, “Vehicular node localization using received-signal-strength indicator,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 6 I, pp. 3371–3380, 2007, doi: 10.1109/TVT.2007.907687.

[44] G. M. Hoang, B. Denis, J. Harri, and D. T. M. Slock, “Robust data fusion for cooperative

vehicular localization in tunnels," *IEEE Intell. Veh. Symp. Proc.*, pp. 1372–1377, 2017, doi: 10.1109/IVS.2017.7995902.

[45] A. J. Alami, K. El-Sayed, A. Al-Horr, H. Artail, and J. Guo, "Improving the Car GPS accuracy using V2V and V2I Communications," *2018 IEEE Int. Multidiscip. Conf. Eng. Technol. IMCET 2018*, pp. 1–6, 2019, doi: 10.1109/IMCET.2018.8603032.

[46] E. I. Adegoke, J. Zidane, E. Kampert, C. R. Ford, S. A. Birrell, and M. D. Higgins, "Infrastructure Wi-Fi for connected autonomous vehicle positioning: A review of the state-of-the-art," *Veh. Commun.*, vol. 20, p. 100185, 2019, doi: 10.1016/j.vehcom.2019.100185.

[47] S. Davar, A. Mohammadi, and K. N. Plataniotis, "Event-Triggered Particle Filtering Via Diffusion Strategies for Distributed Estimation in Autonomous Systems," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2018-April, pp. 6578–6582, 2018, doi: 10.1109/ICASSP.2018.8462150.

[48] Y. Wang, X. Duan, D. Tian, M. Chen, and X. Zhang, "A DSRC-based vehicular positioning enhancement using a distributed multiple-model Kalman filter," *IEEE Access*, vol. 4, pp. 8338–8350, 2016, doi: 10.1109/ACCESS.2016.2630708.

[49] A. H. Sakr and G. Bansal, "Cooperative localization via DSRC and multi-sensor multi-Target track association," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 66–71, 2016, doi: 10.1109/ITSC.2016.7795533.

[50] A. Edelmayer, M. Miranda, and V. Nebehaj, "Cooperative federated filtering approach for enhanced position estimation and sensor fault tolerance in ad-hoc vehicle networks," *IET Intell. Transp. Syst.*, vol. 4, no. 1, pp. 82–92, 2010, doi: 10.1049/iet-its.2009.0017.

[51] A. Khattab, Y. A. Fahmy, and A. Abdel Wahab, "High Accuracy GPS-Free Vehicle Localization Framework via an INS-Assisted Single RSU," *Int. J. Distrib. Sens. Networks*, vol. 2015, pp. 10–18, 2015, doi: 10.1155/2015/795036.

[52] X. Zhong, R. Rabiee, Y. Yan, and W. P. Tay, "A particle filter for vehicle tracking with lane level accuracy under GNSS-denied environments," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2018-March, no. Mm, pp. 1–6, 2018, doi: 10.1109/ITSC.2017.8317659.

[53] H. Lu, S. Zhang, Y. Dong, and X. Lin, "A Wi-Fi/GPS Integrated System for Urban Vehicle

Positioning," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 1663–1668, 2010, doi: 10.1109/ITSC.2010.5625268.

[54] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau, and Y. Dupuis, "Survey on Cooperative Perception in an Automotive Context," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, pp. 1–20, 2022, doi: 10.1109/TITS.2022.3153815.

[55] J. A. Jang, K. Choi, and H. Cho, "A fixed sensor-based intersection collision warning system in vulnerable line-of-sight and/or traffic-violation-prone environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1880–1890, 2012, doi: 10.1109/TITS.2012.2207952.

[56] E. Arnold, M. Dianati, R. De Temple, and S. Fallah, "Cooperative Perception for 3D Object Detection in Driving Scenarios Using Infrastructure Sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1852–1864, 2022, doi: 10.1109/TITS.2020.3028424.

[57] J. Li, J. Gao, H. Zhang, and T. Z. Qiu, "RSE-Assisted Lane-Level Positioning Method for a Connected Vehicle Environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2644–2656, 2019, doi: 10.1109/TITS.2018.2870713.

[58] A. Fascista, G. Ciccarese, A. Coluccia, and G. Ricci, "Angle of Arrival-Based Cooperative Positioning for Smart Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2880–2892, 2018, doi: 10.1109/TITS.2017.2769488.

[59] W. Y. Shieh, C. C. J. Hsu, and T. H. Wang, "Vehicle Positioning and Trajectory Tracking by Infrared Signal-Direction Discrimination for Short-Range Vehicle-To-Infrastructure Communication Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 368–379, 2018, doi: 10.1109/TITS.2017.2697041.

[60] H. Song, W. Choi, and H. Kim, "Robust vision-based relative-localization approach using an RGB-depth camera and LiDAR sensor fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, 2016.

[61] Z. Liu *et al.*, "Robust Target Recognition and Tracking of Self-Driving Cars With Radar and Camera Information Fusion Under Severe Weather Conditions," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–14, 2021, doi: 10.1109/TITS.2021.3059674.

[62] K. Park, S. Kim, and K. Sohn, "High-Precision Depth Estimation Using Uncalibrated LiDAR and Stereo Fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 321–335, 2020, doi:

10.1109/TITS.2019.2891788.

[63] P. Yang, D. Duan, C. Chen, X. Cheng, and L. Yang, “Multi-Sensor Multi-Vehicle (MSMV) Localization and Mobility Tracking for Autonomous Driving,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14355–14364, 2020, doi: 10.1109/TVT.2020.3031900.

[64] M. Brambilla, M. Nicoli, G. Soatti, and F. Deflorio, “Augmenting Vehicle Localization by Cooperative Sensing of the Driving Environment: Insight on Data Association in Urban Traffic Scenarios,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1646–1663, 2020, doi: 10.1109/TITS.2019.2941435.

[65] B. Lv *et al.*, “LiDAR-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities,” *IEEE Access*, vol. 7, pp. 79895–79907, 2019.

[66] M. Gabb, H. Digel, T. Muller, and R. W. Henn, “Infrastructure-supported perception and track-level fusion using edge computing,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2019-June, no. Iv, pp. 1739–1745, 2019, doi: 10.1109/IVS.2019.8813886.

[67] D. Gulati, F. Zhang, D. Clarke, and A. Knoll, “Vehicle infrastructure cooperative localization using Factor Graphs,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2016-Augus, pp. 1085–1090, 2016, doi: 10.1109/IVS.2016.7535524.

[68] D. Gulati, F. Zhang, D. Clarke, and A. Knoll, “Graph-Based Cooperative localization using symmetric measurement equations,” *Sensors (Switzerland)*, vol. 17, no. 6, pp. 1–16, 2017, doi: 10.3390/s17061422.

[69] F. Zhang, M. Liu, D. Gulati, and A. Knoll, “Robust vehicle-infrastructure localization using factor graph and probability data association,” *J. Commun. Networks*, vol. 20, no. 4, pp. 406–413, 2018, doi: 10.1109/JCN.2018.000056.

[70] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” in *Sensor fusion IV: control paradigms and data structures*, 1992, vol. 1611, pp. 586–606.

[71] A. W. Fitzgibbon, “Robust registration of 2D and 3D point sets,” *Image Vis. Comput.*, vol. 21, no. 13–14, pp. 1145–1153, 2003, doi: 10.1016/j.imavis.2003.09.004.

[72] B. Jian, I. C. Society, and B. C. Vemuri, “Robust Point Set Registration Using Gaussian

Mixture Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, 2011.

[73] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[74] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Underst.*, vol. 89, no. 2–3, pp. 114–141, 2003, doi: 10.1016/S1077-3142(03)00009-2.

[75] B. Luo and E. R. Hancock, "A unified framework for alignment and correspondence," *Comput. Vis. Image Underst.*, vol. 92, no. 1, pp. 26–55, 2003.

[76] G. McNeill and S. Vijayakumar, "A probabilistic approach to robust shape matching," in *2006 International Conference on Image Processing*, 2006, pp. 937–940.

[77] B. Luo and E. R. Hancock, "Symbolic graph matching using the EM algorithm and singular value decomposition," *Proc. - Int. Conf. Pattern Recognit.*, vol. 15, no. 2, pp. 141–144, 2000, doi: 10.1109/icpr.2000.906035.

[78] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, 2010.

[79] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No. PR00737)*, 2000, pp. 190–197.

[80] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, 2000.

[81] W. Wen-Tsun, "Basic principles of mechanical theorem proving in elementary geometries," *J. Autom. Reason.*, vol. 2, no. 3, pp. 221–252, 1986.

[82] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, 1999.

[83] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *CVPR*

2011, 2011, pp. 2969–2976.

[84] A. Masselli and A. Zell, "A new geometric approach for faster solving the perspective-three-point problem," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 2119–2124.

[85] T. Ke and S. I. Roumeliotis, "An Efficient Algebraic Solution to the Perspective-Three-Point Problem," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 55455, pp. 7225–7233, 2017.

[86] P. D. Fiore, "Efficient linear solution of exterior orientation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 140–148, 2001.

[87] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009, doi: 10.1007/s11263-008-0152-6.

[88] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 1, pp. 128–136, 2002.

[89] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Online continuous stereo extrinsic parameter estimation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1059–1066, 2012, doi: 10.1109/CVPR.2012.6247784.

[90] W. J. Bottega, *Engineering Vibrations*. 2006.

[91] J. Chen, R. J. Patton, and H.-Y. Zhang, "Design of unknown input observers and robust fault detection filters," *Int. J. Control*, vol. 63, no. 1, pp. 85–105, 1996.

[92] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital control of dynamic systems*, vol. 3. Addison-wesley Reading, MA, 1998.

[93] M. Wang, P. Yu, and Y. Li, "Performance analysis of GNSS/INS loosely coupled integration systems under GNSS signal blocking environment," in *E3S Web of Conferences*, 2020, vol. 206, p. 2013.

[94] A. Fascista, G. Ciccarese, A. Coluccia, and G. Ricci, "A localization algorithm based on V2I communications and AOA estimation," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 126–130, 2017, doi: 10.1109/LSP.2016.2639098.

[95] S. Wang, L. Chen, D. Gu, and H. Hu, “An optimization based Moving Horizon Estimation with application to localization of Autonomous Underwater Vehicles,” *Rob. Auton. Syst.*, vol. 62, no. 10, pp. 1581–1596, 2014, doi: 10.1016/j.robot.2014.05.004.

[96] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” *Proc. IEEE Conf. Decis. Control*, pp. 5492–5498, 2007, doi: 10.1109/CDC.2007.4434303.

[97] H. M. Leonard M Silverman, “CONTROLLABILITY AND OBSERVABILITY IN LINEAR SYSTEMS,” *Control*, vol. 5, no. 1, pp. 64–73, 1967.

[98] R. Olfati-Saber, “Kalman-Consensus filter: Optimality, stability, and performance,” *Proc. IEEE Conf. Decis. Control*, pp. 7036–7042, 2009, doi: 10.1109/CDC.2009.5399678.