Technische Universiteit Eindhoven
University of Technology
Department of Mechanical Engineering
Dynamics and Control Research Group

TNO innovation for life
Helmond, Integrated Vehicle Safety
Shared World Modelling and Localization

Multi-Agent Sensor-to-Global Fusion in a Dynamic and Uncertain Environment

Internship Report

Irfan Mashood Badshah
DC 2017.004

Supervisors:
Dr. Jos Elfring (TNO, Helmond)
Prof.dr. H. (Henk) Nijmeijer (TU/e)

Eindhoven, February 2017

Abstract

This work compares Sensor-to-Global Fusion methods with existing Sensor-To-Sensor Fusion methods for applications in shared world modelling. Sensor-To-Sensor Fusion (e.g. Ellipsoidal Intersection and Covariance Intersection) fuses only new measurements at every time sample from sensors. On the other hand, Sensor-to-Global Fusion methods (e.g. Information Matrix Fusion and Covariance Intersection with Memory), fuses past fusion data with the new measurements. The scenario discussed here simulates two automated vehicles (agents) estimating the states (relative position and velocity) of a third vehicle (object) by fusing the locally obtained measurements with measurements that are shared between the agents to obtain a better estimation of the object without losing consistency. Both Sensor-To-Sensor Fusion and Sensor-to-Global Fusion need to consider the common information between the prior estimates. Besides this, Sensor-to-Global Fusion would also need to consider the correlation between the estimation from the sensor filters and the estimation from the past in global track as both contain common past information. Most analysis of fusion algorithms focus on synchronized sensors. However, this work will also look at the results of fusion of sensors that are asynchronous/out-of-sequence and susceptible to failure (packet loss during communication). Average Root Mean Square Error, Covariance and Consistency of the fused estimates will be compared and the pros and cons of keeping track of past fusion data will be studied.

Preface

This documentation is the work carried out by me during my internship at TNO, Helmond. However, this work would not be complete if I do not thank my mentors and supervisors.

I would first like to thank Dr. Jos Elfring, my supervisor at TNO Helmond who entrusted me with carrying out this work. He has also patiently explained to me the concepts regarding sensor fusion several times which has laid a solid foundation needed to complete my internship. I would also like to thank him for the hours he spent reviewing my work and report and, the constructive feedback provided by him.

Also, I would like to thank Prof.Dr.H.Nijmeijer, my supervisor at TU/e for giving me the encouragement and confidence to explore fields new to me while taking up this internship.

Contents

List of Figures

List of Tables

Acronyms and Symbols

Acronyms

V2V	*Vehicle to Vehicle*
T2TF	*Track-to-Track Fusion*
S2SF	*Sensor-to-Sensor Fusion*
S2GF	*Sensor-to-Global Fusion*
Gaus	*Gaussian Fusion Method*
Kal	*Kalman Fusion Method*
CI	*Covariance Intersection*
EI	*Ellipsoidal Intersection*
IMF	*Information Matrix Fusion*
CIM	*Covariance Intersection with Memory*
RMSE	*Root Mean Square Error*
ARMSE	*Average Root Mean Square Error*

List of symbols

$\underline{\mathbf{x}}$ Random Distribution

$\underline{x}$ Vector

$\underline{\bar{x}}$ Ground Truth Vector

$\underline{\hat{x}}_a$ Prior Estimate Vector

$\underline{\hat{x}}_f$ Fused Estimate State Vector

$\underline{e}_a$ Error- Prior Estimate Vector

$\underline{e}_f$ Error- Fused Estimate Vector

$\mathbf{A}$ Matrix

$\mathbf{P}_a$ Prior Estimate Covariance Matrix

$\mathbf{P}_f$ Fusion Covariance Matrix

$\{\mathbf{Z}\}$ Information in Fisher Information Domain

Chapter 1

Introduction

The future of automobiles is heading towards automated and cooperative driving that are aimed at reducing accidents [19] and improving traffic flow [20]. Transferring the driving duties from the driver to the vehicle requires the vehicle to make its own decisions. However, good decision making is primarily supported by good knowledge of the environment. Though adding multiple sensors to such vehicles would increase their perception of the environment, they are limited by the field of view of these sensors. Another option would be to share the world model perceived by each vehicle. This shared information can be combined with the information perceived by each vehicle using its on-board sensors which would expand the knowledge of its environment. Also, uncertainties of a sensors can be reduced and the effects of sensor/communication failure can be attenuated.

1.1 Sensor Fusion

Autonomous driving vehicles rely heavily on sensors to perceive its surroundings. However, as sensors have inherent uncertainties in determining the states of the target such as position, velocity and acceleration, it would be wiser to observe an object using multiple sensors and combine the obtained information from these sensors to better estimate the states of the object. This process of combining measurements from multiple sensors for better state estimation of a tracked object is called sensor fusion. Apart from possibly increasing the accuracy of the estimate, fusing sensor data also increases robustness against sensors failure as multiple sensors are observing a single object.

Sensor fusion can be divided into two main problems [1], track association and state estimation problems. Each sensor outputs measurements such as distance and velocity of multiple objects in its Field of View. Track association deals with assigning the measurements from different sensors with a certain object in space. When the measurements from multiple sensors have been assigned to each object, these measurements are fused to obtain the most accurate estimate. This is referred to as state estimation.

This research will be limited to estimating the states (position and velocity in the longitudinal and lateral direction) of a single tracked object after track association using information such as the state estimates and error covariance estimates from each sensor.

1.2 Problem Statement

The goal of the project is to study Sensor Fusion Algorithms suitable for Vehicle-to-Vehicle (V2V) communication and analyse its performance in a given driving scenario. The performance of the fusion algorithms will be analysed while simulating real life scenarios such as when the sensors output measurements at different frequencies and in the presence of sensors failures (packet losses during communication). The fused estimates will be checked for consistency (error between fused estimate and ground truth is bounded) and should have lesser error and uncertainty than the prior estimates (measurement from sensor).

1.3 Plan of Approach

As the goal was to study and analyse sensor fusion algorithm suitable for V2V communication applications the following tasks were executed

- Study about sensor fusion, underlying concepts and assumptions required for the simulation setup.
- Choose Sensor Fusion algorithms suitable for V2V Communication.
- Develop simulation testing tool to analyse sensor fusion algorithms.
- Evaluate and study performance of the chosen sensor fusion algorithms under different conditions.

1.4 Report Structure

This internship report is structured as follows. Chapter 2 provides the background information and previous study regarding sensor fusion techniques. Chapter 3 describes the simulation setup used. It also describes the sensor fusion algorithms implemented and the evaluation methods required to quantify the performance of the algorithms. The Results are presented in Chapter 4 after which the Conclusion and Recommendations is given in Chapter 5.

Chapter 2

Background Information

2.1 Introduction

This chapter provides an overview of the sensor fusion techniques used in the industry. Fusing of information from sensors requires understanding of certain underlying principles such as the presence of common information in the measurements from the sensors and effects of synchronous and asynchronous sensors on the fused estimates. These will also be covered in this chapter. Moreover, the concept of consistency of estimate will be explained.

2.2 Track to Track Fusion

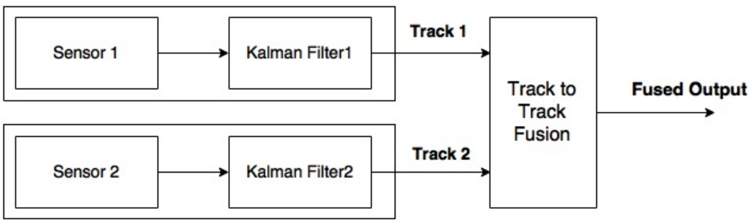

Figure 2.1: Track-to Track Fusion

In Track-to-Track Fusion (T2TF), the sensor feeds its measured information (e.g. distance, velocity) to a central fusion algorithm after local processing to reduce noise using an estimator (Figure. 2.1). The processed information from the sensor and filter is called a track. Assuming only a single object is present (no track association problems) the fusion algorithm fuses multiple sensor tracks (local tracks) into a unified global track [2]. Considering underlined boldfaced variables $\underline{\mathbf{x}}$ are used for random quantities (eg.Normal distribution, $\underline{\mathbf{x}} \sim N(\underline{\hat{x}}, \mathbf{A})$), where $\underline{\hat{x}}$ is the state vector containing the means of position and velocity in the longitudinal(x) and lateral(y) direction and covariance matrix is given as upper-case boldfaced $\mathbf{A}$, the output of the local tracks , Track1 and Track2 $[(\underline{\mathbf{x}}_a \sim N(\underline{\hat{x}}_a, \mathbf{P}_a)$ and $(\underline{\mathbf{x}}_b \sim N(\underline{\hat{x}}_b, \mathbf{P}_b))]$ will be fused to obtained $(\underline{\mathbf{x}}_f \sim N(\underline{\hat{x}}_f, \mathbf{P}_f)$.

As these tracks can be formed by any sensor given that the data from the sensors are the state estimate and error covariance, T2TF is modular and flexible. The tracks by each sensor are correlated to each other due to common information caused by same modelling assumptions, same prior information or double counting of data. Fusion methods such as a Kalman Filter assume that the information from the filters are not correlated. T2TF on the other hand accounts for this correlation leading to better estimate of the fused data.

Track to track fusion can be of the two types, Sensor-to-Sensor (S2S) and Sensor-to-Global (S2G) [4].

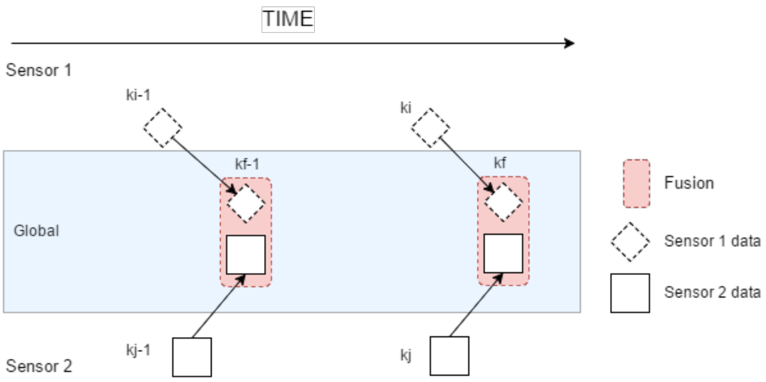

Figure 2.2: Sensor-to-Sensor Track Fusion

Sensor-to-Sensor track fusion (Figure(2.2)), also known as Track-to-Track Fusion without memory, fuses sensor level tracks to form global level tracks at predefined cycles. In Figure.2.2, sensor 1 and 2 output information at time $ki - 1$ and $kj - 1$ which are fused at time $kf - 1$. However, this information is not used at time kf. As there is no memory of the previous time samples, the track association and tracking errors are not propagated from one time step to the next. However, if the sensors are observing the same object, cross-correlation can exist between the sensor tracks due to common information.

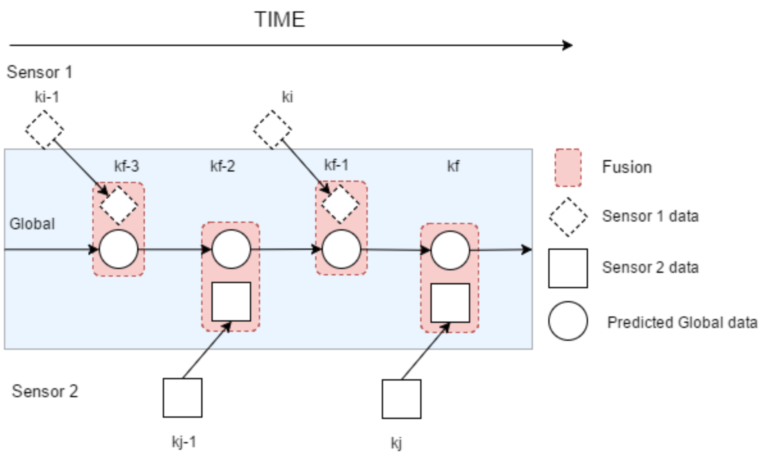

Figure 2.3: Sensor-to-Global Track Fusion

Sensor-to- Global track fusion (Figure(2.3)) , also called Track-to-Track fusion with memory maintains a global track over time by fusing sensor level tracks. In Figure.2.3, sensor 1 output information at time $ki - 1$ which is fused to the global track at time $kf - 3$. This information is used at time $kf - 2$ when fusing information from Sensor 2. However because of the memory, common information in the global and local tracks give rise to correlation between the two as both the global and local tracks carry the same past information. Moreover, past processing errors in the system tracks can affect future fusion performances [4][5].

2.3 Correlated Estimates

The information from the sensors are said to be correlated to each other if they contain common information. If the common information is not removed, it can be added more than once to the final fused estimate leading to erroneous results. Some of the causes of correlation are as follows:-

$$\underline{\mathbf{x}}_{k+1} = \mathbf{A}_k \underline{\mathbf{x}}_k + \mathbf{w}_k \tag{2.1}$$

Underlined boldfaced variables $\underline{\mathbf{x}}$ are used for random quantities, matrix is given as upper-case boldfaced variables $\mathbf{A}$ and vectors as underlined variables $\underline{x}$. Considering a Kalman filter model as given by (2.1), with the same modelling assumptions (e.g. same process noise $\mathbf{w}_k$ in the state transition equation during prediction, (2.1)) when using Kalman Filters in each track gives rise to common data in both the tracks.

Also, multiple parallel running Kalman Filters working on the same estimation problem give rise to correlations when they are initialized with the same prior estimates.

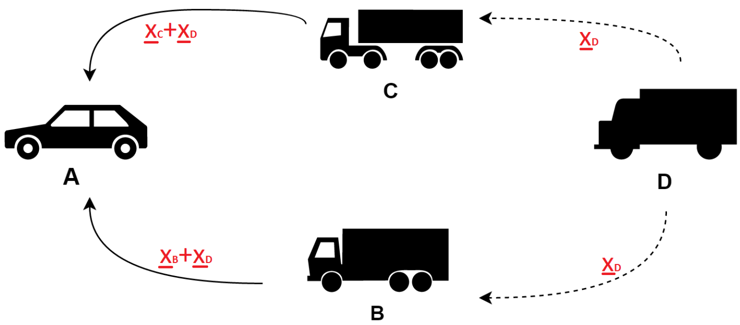

Figure 2.4: A common cooperative world model where objects share information.

Another source of correlation which is relevant to cooperative driving is double counting of sensor data. In the case as shown in Figure 2.4, let each vehicle observe a random distribution $\underline{\mathbf{x}}$. The object D communicates its observation $\underline{\mathbf{x}}_D$ to agents B and C. The agents B and C combine their observed states $\underline{\mathbf{x}}_B$ and $\underline{\mathbf{x}}_C$ respectively with that received from D. Now the information transferred to A from both B and C contain observation of D which is unknown to A. The common information D causes correlation between information B and C [7].

Implementing S2GF gives rise to correlation between global track and the local tracks due to same past information. This will be addressed in the coming sections.

Common fusing algorithms such as a Kalman Filter or fusing the two estimates into a common Gaussian Distribution assumes the prior estimates to be uncorrelated (or conditionally independent in case of using a Kalman filter) as the measurements are from different sensors. However, making this assumption is not right [7].

2.3.1 Synchronous and Asynchronous Sensors Measurement

Sensors used in real world applications output measurements at different rates giving rise to asynchrony. This is caused by the delay between the sensor actually observing the physical object and the data arriving at the fusion centre.The time required for acquisition, pre-processing and transferring the data to the fusion sensor adds up to cause the delay [8]. This could vary from sensor to sensor resulting in a lack of synchrony between them. Asynchrony can also be caused by the communication delay in wireless communication.

In case of asynchronous sensors,the measurements from each sensor can be aligned in time by predicting their states to the arrival time at the fusion sensor [9]. However, predicting estimates will increase the uncertainty[8].

2.3.2 Consistency of Estimate

The covariance matrix $\mathbf{P}$ containing information of the uncertainty of the states (eg. distance) in both the x and y direction can be graphically represented as an ellipse using the formula given below.

$$p : \{\underline{x}^T \mathbf{P}^{-1} \underline{x} = c\} \tag{2.2}$$

where c is a constant. The major and minor axis of the ellipse is equal to the standard divination σ long the the x and y axis or a multiple of the standard deviation depending on the value of c.

The ellipse indicates the region where the tracked objected is expected lie for a certain probability.

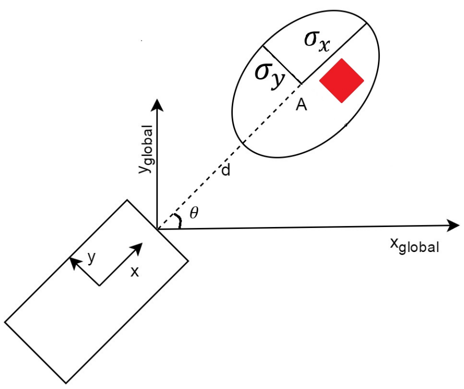

Figure 2.5: Consistent Estimate

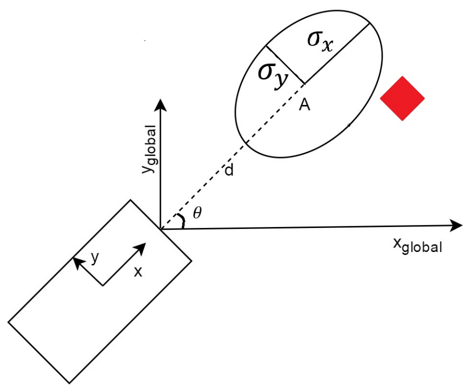

Figure 2.6: Inconsistent Estimate

In Figures 2.5 and 2.6, a sensor with local axis x and y tracks an object (red box). The sensor measurement is 'd' and the uncertainty is presented as a ellipse with major and minor axis of the first standard deviation σ_x and σ_y. In the figure 2.5, the red object lies within the ellipse/ the first standard deviation and is said to be consistent. However, in figure 2.6, the object lies outside the ellipse and the measurement is said to by inconsistent.

2.4 Summary

This chapter introduced the concepts that will be useful in the following chapter such as Sensor-to-sensor fusion and Sensor-to-Global fusion method which will be of help when choosing the required sensor fusion algorithms. Also the concepts of Synchronous and Asynchronous sensors, and correlated prior estimates will be used while creating the simulation setup. Lastly, the check for consistency will be one of the methods used while analysing the sensor fusion methods.

Chapter 3

Simulation Setup

3.1 Introduction

This chapter describes the experimental setup (simulation model) used to study the performance of the fusion algorithms. As it is important to study the error between the actual states of the vehicle and those estimated by the sensor (prior estimates) and the fusion algorithm (fused estimates), a simulation tool that generates the ground truth (actual state of the vehicle) was used. Actual sensor data logged on a vehicle was not used as the ground truth would be unknown.

Based on the literature study in the previous chapter, the design approach is presented in Section 3.2. A more detailed explanation of the individual components of the simulation setup is described in sections following Section.3.2. The driving scenario used in the simulation is described in Section3.3. Section3.4 describes the generation of the ground truth and the prior estimates from sensors. This is followed by Section 3.5 which details the model used in Kalman filter along with the values of the sensor and process noise used. Section 3.6 describes the sensor fusion algorithms used. The last section describes the methods used to evaluate the performance of fusion algorithms.

3.2 Design Approach

The simulation setup is based on the study of sensor fusion and its underlying principles presented in the previous chapter.

As cooperative driving is aimed at tackling complex situations, a highway driving scenario with multiple vehicles accelerating in both the longitudinal direction and lateral direction is chosen for the simulation. To describe the motion of the vehicles with respect to each other, sinusoidal signals will represent the relative distance between the vehicles. A sinusoidal signal is chosen as it can be easily differentiated without reaching a point of no differentiability to obtain other states of the vehicles such as velocity and acceleration. A Kalman filter with a Constant Velocity Model was chosen as the local estimator to reduce noise. A constant velocity model would be sufficient as an object with large inertia cannot generate an acceleration large enough to change the position and velocity over a very small time sample. Hence the acceleration [which will not be measured] would be modelled as an uncertainty in position and velocity. The outputs from the Kalman filter would be the mean and the covariance matrix which would be used as the inputs for the fusion algorithms.

The sensor fusion algorithms are classified here as memoryless methods (S2SF) and methods with memory (S2GF). The work done by N.J.G. Koenraad [21] focusses on memoryless fusion methods and its performance in cooperative driving. However, memoryless methods are not robust against sensor failure and packet losses during communication. The two S2S methods presented here are Ellipsoidal Intersection (EI) and Covariance Intersection (CI). More robust fusion methods with memory such as Information Matrix Fusion (IMF) and Covariance Intersection with Memory (CIM) suitable for V2V communication would be presented in this chapter.These four methods have been chosen as they all estimate the common information in the prior estimates differently.

The performance of the fusion algorithms will be studied under an ideal situation and situations with sensor delays and communication failures.

3.3 Scenario Description

The scenario described here simulates a cooperative platoon formation of two agents on a highway, with a third object in close vicinity to the two agents (Figure. 3.1). The third object is observed by both the agents using their on-board sensors such as radar, camera etc. The world model (objects perceived by the agents using the on-board sensors) is shared amongst the two agents. One of the agents combines the measurement from the local sensor with those received from the other agent to better estimate the position of the object. The on-board sensors are assumed to have the same local estimator, a Kalman Filter giving rise to correlation between the prior estimates. The states of the object tracked would be the relative position and velocity.

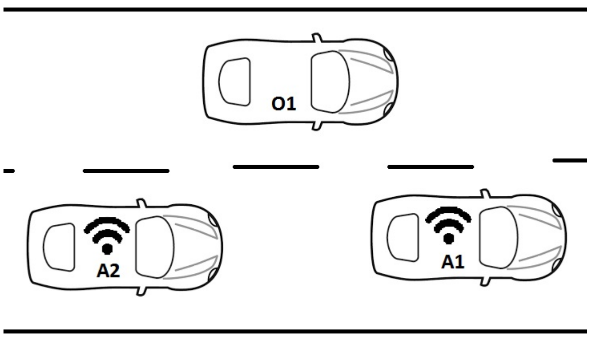

Figure 3.1: Scenario of highway platoon of agents (A1& A2) with obstacle (O1) on next lane

3.4 Generation of Ground Truth and Prior Estimates

The driving scenario depicted in Section.3.3 was modelled on `MatLab/Simulink`. The relative states (position and velocity) between Agent1 and the Object is generated as a sinusoidal signal [hence smooth changing velocity and position] (Figure.3.2). This is taken as the ground truth measurements (actual distance) between the agent and object. To this signal, noise is added to model it as a real world signal (Figure.3.3) and is passed through a Kalman Filter. The Kalman Filter has a Constant Velocity Model as given by (3.3). This processed signal can be taken as the first estimate $[N(\underline{\hat{x}}_a, \mathbf{P}_a)]$ of the distance between Object and Agent1 as observed by Agent1.

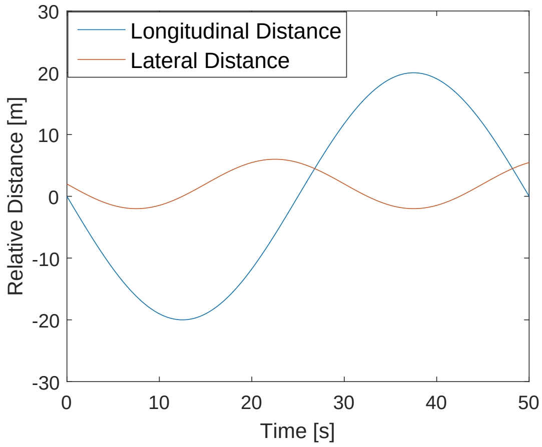

Figure 3.2: Ground Truth - Relative Distance between Agent1 and Object

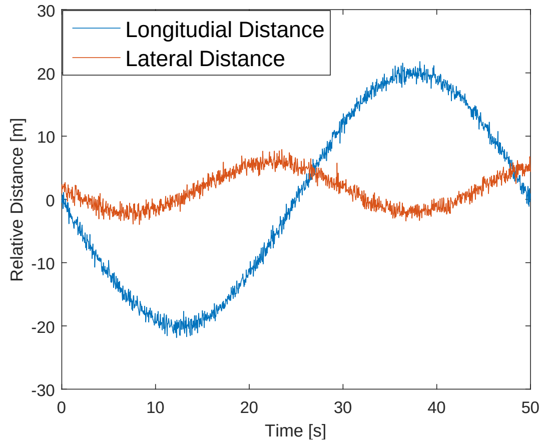

Figure 3.3: Noisy Estimate - Relative Distance between Agent1 and Object

Similarly, the relative states between Agent1 and Agent2 ($N(\underline{\hat{x}}_{21}, P_{21})$) and, Object and Agent2 ($N(\underline{\hat{x}}_{2o}, P_{2o})$) is generated. Using (3.1) the processed signals of the relative states of Object with respect to Agent2 and relative state of Agent1 with respect to Agent2 is used to calculate the relative states of Object with respect to Agent1 as observed by Agent2.

$$\begin{aligned} \mathbf{P}_b &= \mathbf{P}_{2o} + \mathbf{P}_{21} \\ \underline{\hat{x}}_b &= \underline{\hat{x}}_{2o} - \underline{\hat{x}}_{21} \end{aligned} \tag{3.1}$$

This is taken as the second estimate $N(\underline{\hat{x}}_b, \mathbf{P}_b)$. Both the estimates $N(\underline{\hat{x}}_a, \mathbf{P}_a)$ and $N(\underline{\hat{x}}_b, \mathbf{P}_b)$ are of Object with respect to Agent 1. The estimates $N(\underline{\hat{x}}_a, \mathbf{P}_a)$ and $N(\underline{\hat{x}}_b, \mathbf{P}_b)$ are then fused using one of the fusion algorithms to obtain position and velocity of Object with respect to Agent1 ($N(\underline{\hat{x}}_f, \mathbf{P}_f)$-fused estimate).

Considering $\underline{\bar{x}}$ as the ground truth and $\underline{\hat{x}}_a$ as the prior estimate, the average error (difference between ground truth and prior estimate, $e_a = |\underline{\bar{x}} - \underline{\hat{x}}_a|$) in position for a 100sec simulation is given in table 3.1. It should be noted that this error can be reduced by changing the sensor (reducing the noise). However, as the focus is on reducing the error by fusing the two prior estimates, decreasing the magnitude of the errors from the sensors by reducing sensor noise should not be given much attention.

$ARMSE.10^{-2}$	x	y	Σ
$\underline{e}_a$	2.65	2.49	5.14
$\underline{e}_b$	1.06	4.50	5.56

Table 3.1: Average Root Mean Square Error-Position (Prior Estimates)

To make the simulation more realistic, measurements from one of the agents are delayed to make the sensors asynchronous. Frequencies of sensors 1 and 2 if they are asynchronous would be 12.5Hz and 25Hz respectively. A prediction step then follows this delaying step to synchronize the sensors which in turn would increase the uncertainty of the measurement. However, the prediction step can be neglected to study the results of fusion without synchronization. To simulate sensor failure due to loss of communication, the estimated values from the model will not be sent to the fusion algorithm for a certain time duration. The failure will occur 10% of the time.

3.5 Constant Velocity Vehicle Model

The states of the target object observed by the sensors at time k are longitudinal and lateral distance and velocities. They can be represented in a vector form as

$$\underline{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T \tag{3.2}$$

The model used in the Kalman Filter (as seen in Figure 2.1) for predicting the states at time $k+1$ is a constant velocity model (3.3), assuming only the position and velocity of tracked object can be measured. As the acceleration is not measured it is unknown and uncertain. This uncertainty is modelled as zero mean Gaussian Distributed process noise in position and velocity.

$$\underline{x}_{k+1} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{F}} \underline{x}_k + \underbrace{\begin{bmatrix} \frac{\Delta T^2}{2} & 0 & 0 & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 & 0 \\ 0 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & \Delta T \end{bmatrix}}_{\text{G}} \underline{\mathbf{w}}_k \tag{3.3}$$

$$\underline{y}_k = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{H}} \underline{x}_k + \underline{\mathbf{v}}_k \tag{3.4}$$

The time step of the simulation ΔT is taken as 0.04s. $\underline{x}_k$ is the states of the tracked object $\underline{y}_k$ is the measurement output of the state $\underline{x}_k$ and, $\underline{\mathbf{v}}_k$ and $\underline{\mathbf{w}}_k$ are the zero mean Gaussian Distributed measurement of sensor noise and process noise (acceleration) respectively. The covariance of process noise $\underline{\mathbf{w}}(k)$ and sensor noise $\underline{\mathbf{v}}(k)$ for (3.3) is given as follows:-

$$\begin{aligned} cov(\underline{\mathbf{w}}_1(k)) &= W_1 = diag(5^2, 1^2, 5^2, 1^2) \\ cov(\underline{\mathbf{w}}_{2a}(k)) &= W_{2a} = diag(5^2, 1^2, 5^2, 1^2) \\ cov(\underline{\mathbf{w}}_{2b}(k)) &= W_{2b} = \frac{1}{5} diag(5^2, 1^2, 5^2, 1^2) \\ cov(\underline{\mathbf{v}}_1(k)) &= V_1 = diag(0.75^2, 0.75^2, 0.75^2, 0.75^2) \\ cov(\underline{\mathbf{v}}_{2a}(k)) &= V_{2a} = diag(0.25^2, 1.25^2, 0.25^2, 1.25^2) \\ cov(\underline{\mathbf{v}}_{2b}(k)) &= V_{2b} = diag(0.25^2, 1.25^2, 0.25^2, 1.25^2) \end{aligned} \tag{3.5}$$

In (3.5), the matrix of $cov(w(k))$ and the matrix of $cov(v(k))$ are of the form $diag(x, y, \dot{x}, \dot{y})$. W_1 and V_1 is noise associated with Agent1 observing Object, W_{2a} and V_{2a} is noise associated with Agent2 observing Object, W_{2b} and V_{2b} is noise associated with Agent2 observing Agent1. W_{2b} is smaller as both agents try to maintain constant distance and velocity between the two. Also, it can be assumed that the acceleration in the longitudinal direction (x) is greater than that in the lateral direction (y) if the vehicle is driving on a highway. The values for the sensor noises V_1, V_{2a} and V_{2b} are obtained from data sheets of common automotive sensors and have different magnitudes in different directions.

3.6 Sensor Fusion Algorithms

The input for the fusion algorithms are the updated mean($\hat{\underline{x}}$) and covariance matrix ($\mathbf{P}$)of the prior estimates from the Kalman Filters. The prior estimates from the Kalman Filter ($\underline{\mathbf{x}}_a \sim N(\hat{\underline{x}}_a, \mathbf{P}_a)$& $\underline{\mathbf{x}}_b \sim N(\hat{\underline{x}}_b, \mathbf{P}_b)$) will be fused by the fusion algorithms to obtained ($\underline{\mathbf{x}}_f \sim N(\hat{\underline{x}}_f, \mathbf{P}_f)$. Two Sensor-to-Sensor fusion methods and two Sensor-to-Global are presented below.

3.6.1 Sensor to Sensor Fusion

Ellipsoidal Intersection (EI)

EI splits the prior estimates $\underline{\mathbf{x}}_a$ and $\underline{\mathbf{x}}_b$ into three pair-wise independent parts, one of which ($N(\gamma, \Gamma)$) is common to the two prior estimates. It then combines the three parts using the formula below.

$$\begin{aligned} \mathbf{P}_f^{EI} &= (\mathbf{P}_a^{-1} + \mathbf{P}_b^{-1} - \Gamma^{-1}) \\ \hat{\underline{x}}_f^{EI} &= \mathbf{P}_f(\mathbf{P}_a^{-1}\hat{\underline{x}}_a + \mathbf{P}_b^{-1}\hat{\underline{x}}_b - \Gamma^{-1}\gamma) \end{aligned} \tag{3.6}$$

Any correlation between the prior estimates will be given by($N(\gamma, \Gamma)$). Here γ and Γ are the mutual mean and covariance. Γ is obtained such that the smallest Γ satisfies $\mathbf{P}_a \preceq \Gamma$ and $\mathbf{P}_b \preceq \Gamma$. This would ensure that the fused covariance matrix is lesser than the prior estimates ($\mathbf{P}_f \preceq \mathbf{P}_a$ and $\mathbf{P}_f \preceq \mathbf{P}_b$), as given in [16] . As this method always guarantees higher accuracy for the fused estimates than the prior estimates, this method is considered though consistency is not guaranteed.

Covariance Intersection (CI)

Work done in [14] mathematically proves that CI guarantees consistency. Consider the prior estimates $N(\hat{\underline{x}}_a, \mathbf{P}_a)$ and $N(\hat{\underline{x}}_b, \mathbf{P}_b)$ and ellipse drawn from $\mathbf{P}_a$ and $\mathbf{P}_b$, the fused ellipse always lies inside the intersection of the prior estimates for all value of $\mathbf{P}_{ab}$ (correlation between prior states). As $\mathbf{P}_{ab}$ is unknown, CI finds an ellipse that tightly encloses the whole intersection region using convex combination of the prior mean and covariance as given below.

$$\begin{aligned} \mathbf{P}_f^{CI} &= (\omega\mathbf{P}_a^{-1} + (1-\omega)\mathbf{P}_b^{-1})^{-1} \\ \hat{\underline{x}}_f^{CI} &= \mathbf{P}_f(\omega\mathbf{P}_a^{-1}\hat{\underline{x}}_a + (1-\omega)\mathbf{P}_b^{-1}\hat{\underline{x}}_b) \end{aligned} \tag{3.7}$$

Where ω is a weighting coefficient computed by minimizing the objective function:

$$\omega = arg \min_{\omega\epsilon[0,1]} \{det(\omega\mathbf{P}_a^{-1} + (1-\omega)\mathbf{P}_a^{-1})\} \tag{3.8}$$

Either the least determinant or the least trace of the matrix could be minimized by solving the optimization problem. Even if the correlation is unknown, any value of ω between [0,1] would guaranty consistency unlike fusion using a Kalman Filter. However, the Covariance Intersection method always considers estimates to be correlated and neglects possible independent information. This results in the estimates being conservative (very high uncertainty) [15]. This could be prevented by using methods such as Split-Covariance method.

3.6.2 Sensor to Global Fusion

Information Matrix Fusion (IMF)

The Information Matrix Fusion is based on the Information filter where the Information Matrix (Inverse of Error Covariance) and Information Vector (state estimation) gives the amount of information carried by an observation. Unlike CI and EI, previous information is not discarded. The underlying principle of IMF is to decorrelate the information between updated measurements from sensors and those predicted from the previous time step to obtain the new information for each local track. This new information is then fused with the information in the global track.

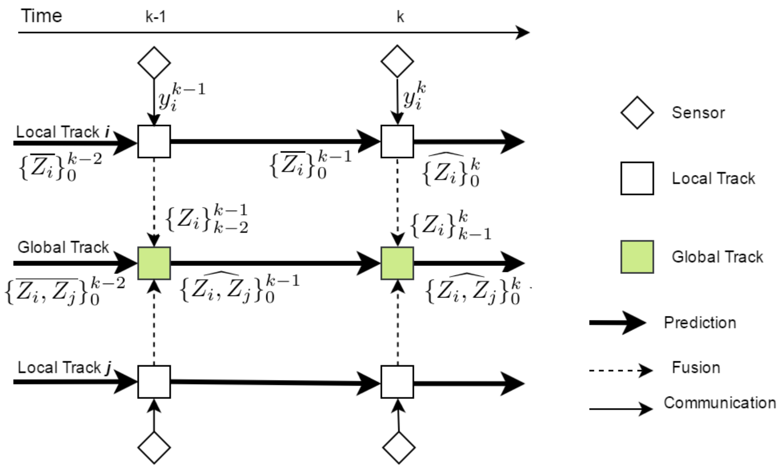

Figure 3.4: IMF information flow diagram

Taking the inverse of covariance matrix $\mathbf{P}^{-1}$ gives information of the data in Fisher Information Domain. In IMF (Figure. 3.4),$\{Z\}$ ($\{Z\} = \mathbf{P}^{-1}$) denotes the information or set of measurements in the local tracks after filtering. $\{\overline{Z}\}$ denotes predicted information and $\{\widehat{Z}\}$ denotes fused information. Here, at time $k-1$, sensors sends measurements y_i^{k-1} and y_j^{k-1} to the local tracks i and j (White boxes) respectively. Considering track i, these measurements updates the predictions $\{\overline{Z_i}\}_0^{k-2}$ as in Kalman filter. The output of a Kalman filter $\{Z_i\}_0^{k-1}$ at time $k-1$ contains all information of the local track from time 0 till $k-1$. However, information $\{Z_i\}_0^{k-2}$ is also present in the global track in $\{\overline{Z_i, Z_j}\}_0^{k-2}$ which has been predicted from the previous time sample and hence needs to be removed before fusing to prevent it from adding it twice (3.9).The common information from the output of the Kalman filter is removed by using the prediction $\{\overline{Z_i}\}_0^{k-2}$ from the Kalman filter to obtain new information $\{Z_i\}_{k-2}^{k-1}$ (decorrelation), which is sent to the global track (Green box) where the new information $\{Z_i\}_{k-2}^{k-1}$ is fused with information in the global track.

$$
\begin{aligned}
\{Z_f\}^k &= \{\overline{Z_i, Z_j}\}_0^{k-2} + \{Z_i\}_0^{k-1} \\
&= \{\overline{Z_j}\}_0^{k-2} + \{\overline{Z_i}\}_0^{k-2} + \{Z_i\}_0^{k-2} + \{Z_i\}_{k-2}^{k-1} \\
&= \{\overline{Z_j}\}_0^{k-2} + 2 * \{\overline{Z_i}\}_0^{k-2} + \{Z_i\}_{k-2}^{k-1}
\end{aligned} \tag{3.9}
$$

We can then take $\{Z\} = \mathbf{P}^{-1}$ for the first equation in (3.10) and $\{Z\} = \mathbf{P}^{-1}\hat{\underline{x}}$ for the second equation in (3.10). Considering (3.9), $\{Z\}_0^k - \{Z\}_0^{k-1} = \mathbf{P}^{-1}_{(l,k|k)} - \mathbf{P}^{-1}_{(l,k|k-1)}$ removes the common information $\{Z\}_0^{k-1}$ resulting in only $\{Z\}_{k-1}^k$ being sent for fusion. The fusion formula is given as below.

$$
\begin{aligned}
\mathbf{P}^{-1}_{(g,k|k)} &= \mathbf{P}^{-1}_{(g,k|k-1)} + (\mathbf{P}^{-1}_{(l,k|k)} - \mathbf{P}^{-1}_{(l,k|k-1)}) \\
\mathbf{P}^{-1}_{(g,k|k)}\hat{\underline{x}}_{(g,k|k)} &= \mathbf{P}^{-1}_{(g,k|k-1)}\hat{\underline{x}}_{(g,k|k-1)} + (\mathbf{P}^{-1}_{(l,k|k)}\hat{\underline{x}}_{(l,k|k)} - \mathbf{P}^{-1}_{(l,k|k-1)}\hat{\underline{x}}_{(l,k|k-1)})
\end{aligned} \tag{3.10}
$$

Where $\mathbf{P}^{-1}_{(g,k|k)}$ is the inverse of the covariance matrix of the global fused estimate at time k containing all information till time k. $\mathbf{P}^{-1}_{(g,k|k-1)}$ is the inverse covariance matrix of the prediction of the global track made for time k using information till time $k-1$. $\mathbf{P}^{-1}_{(l,k|k)}$ is the inverse of the updated covariance matrix of the local track l. $\mathbf{P}^{-1}_{(l,k|k-1)}$ is the inverse covariance matrix of the prediction of the local track l. $\underline{\hat{x}}_{(g,k|k)}$ is the fused global estimate. $\underline{\hat{x}}_{(g,k|k-1)}$ the predicted global estimate . $\underline{\hat{x}}_{(l,k|k)}$ the local estimate at time k using information till k. $\underline{\hat{x}}_{(l,k|k-1)}$ is the prediction at time k using information till $k-1$.

Main Advantages

- Filtering any number of sensors at each time step by summing information matrix or vectors
- Initial guess for information matrix and vector need not be made. They can be taken as 0.
- Robust against sensor failure and packet loss/drop out

Disadvantages

- Less accurate when local sensors are asynchronous.
- Decorrelation requires previously fused sensor-level track to be saved
- Communication or storage of large amount of information.

Covariance Intersection with Memory (CIM)

Another suggested Sensor-to-Global Fusion algorithm is Covariance Intersection with Memory. It is similar to the CI method mentioned in section 3.6.1. The equation is given below. Unlike CI, it maintains a global track by fusing new data from the sensor. A drawback of CIM is that the information from the local track is not decorrelated before fusing. This would then cause the error to increase.

$$\begin{aligned} \mathbf{P}_f^{CIM} &= (\omega_1 \mathbf{P}_g^{-1} + \omega_2 \mathbf{P}_a^{-1} + \omega_3 \mathbf{P}_b^{-1})^{-1} \\ \underline{\hat{x}}_f^{CIM} &= \mathbf{P}_f^{CIM}(\omega_1 \mathbf{P}_g^{-1} \underline{\hat{x}}_g + \omega_2 \mathbf{P}_a^{-1} \underline{\hat{x}}_a + \omega_3 \mathbf{P}_b^{-1} \underline{\hat{x}}_b) \\ Where \quad & \omega_1 + \omega_2 + \omega_3 = 1 \end{aligned} \tag{3.11}$$

The values of $\omega_1, \omega_2, \omega_3$ are obtained by solving a convex optimization problem to get a covariance matrix $\mathbf{P}_f^{CIM}$ with the least determinant. This would ensure that ellipse formed by $\mathbf{P}_f^{CIM}$ would encircle the interaction between the prior estimates and global estimates completely.

3.7 Limitations of Simulation Setup

It should be kept in mind that lower error in the prior estimate from the sensor can be obtained by using sensors effected by lesser noise. However, as the aim is to obtain estimates with lesser error by fusing two noisy signals from sensors, modelling the exact characteristics of a sensors (eg. distance dependent error) was not required. Moreover, due to the same reason no effort was put into actual verification of the generated signal with those from actual sensors.

Also, the only source of common information in the prior estimates is due to the Kalman Filter. This is considered sufficient as adding another source of common information as presented in Section2.3 will increase the amount of common information but will not change the outcome of the results. It is only required to have common information in the prior estimates irrespective of its magnitude.

3.8 Evaluation Methods

The ground truth, prior estimates from the sensor and the fused estimates from the four fusion algorithms were compared using the methods below.

3.8.1 Average Root Mean Square Error

Each simulation runs for 100s and the absolute errors (difference between ground truth and prior estimate and ground truth and fused estimate) of the position and velocity in both the longitudinal and lateral directions will be calculated. This will be plotted with the 95% confidence region which is equal to 1.96 times the standard deviations obtained from the covariance matrix of each state. The 95% region is the region where the probability of the error being less than 1.96 times the standard deviation is 95% if the estimate is consistent. If the error lies outside this region for a large part of the simulation, it could hint that the algorithm is inconsistent. However, as conclusions cannot be drawn from a single simulation, results produced by several runs of the same test can be combined to produce estimation errors without statistical inconsistencies. This is called the Monte Carlo Simulation. 10000 Monte Carlo Simulation will be run to estimate the Root Mean Square Error (RMSE) for the positions and velocities across all simulations at each time samples and plotted with the 95% confidence region. The Average Root Mean Square Error (ARMSE) gives the sum of the error of each state(position and velocity) for all time samples across all simulation. A lower ARMSE value would imply the algorithm is good. The equations for RMSE and ARMSE are given below.

$$RMSE(k) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\underline{\hat{x}}_n(k) - \underline{\bar{x}}_n(k))^2} \tag{3.12}$$

$$\underline{ARMSE(k)} = \frac{1}{K}\sum_{k=0}^{K} RMSE(k) \tag{3.13}$$

Where N is the number of Monte Carlo simulations, K is the total time, $\underline{\hat{x}}$ can be either the states of the prior estimates or fused estimates and $\underline{\bar{x}}$ is the ground truth.

3.8.2 Confidence/Covariance of Estimate

The fused estimate is a Gaussian distributed data ($\underline{\mathbf{x}}_f \sim N(\underline{\hat{x}}_f, \mathbf{P}_f)$). The covariance $\mathbf{P}_f$ indicates how uncertain (standard deviation) the fused estimate is. The covariance of the fused data should be smaller than or equal to the prior data. This would indicate the fused result is more confident (lesser uncertainty) than prior information. The smaller the covariance, the better the fusion algorithm.

3.8.3 Consistency of Estimates

One of the important criteria is that the fused estimates given as $(\underline{\hat{x}}_f, \mathbf{P}_f)$,should remain consistent if the prior estimates are known to be consistent. The consistency of an estimate means the covariance matrix of the error $\underline{\tilde{x}} = |(\underline{\hat{x}}_f - \underline{\bar{x}})|$ over a large number of simulations (10000 Monte Carlo simulations) should be smaller/bounded by the covariance matrix obtained by fusion (P_f) at any time sample.

$$\mathbf{P}_f - \tilde{\mathbf{P}} \succeq 0 \tag{3.14}$$

Where $\tilde{\mathbf{P}} = E[\underline{\tilde{x}}\underline{\tilde{x}}^T]$ is the covariance of the error matrix.

(3.14) would ensure the object actually lies within the region estimated by the fused ellipsoid drawn from the estimated covariance matrix $\mathbf{P}_f$. The ellipse can be draw as the locus of the points given by (3.15) where $c = 5.991$ to obtain the major and minor axis equal to 1.96 times the standard deviation.

$$p : \{\underline{x}^T \mathbf{P}_f^{-1} \underline{x} = c\} \tag{3.15}$$

3.8.4 Robustness

Sensors and wireless communication in connected vehicles are always susceptible to failure (due to packet losses in communication) and hence its influence on the fusion algorithm should be studied.Though using multiple sensors should make the system robust against failure of a sensor, relying on a single sensor would then cause the error to increase as no fusion would take place, and fusion of prior estimates is expected to give lesser error than the prior estimate. The full extent of sensor failures can be studied only when all sensors fail at the same time. However, this is very unlikely to happen. As stated earlier, in the simulation one of the sensors inputs would fail 10% of the time. The response of fusion algorithms to failure would be studied in terms of ARMSE.

3.8.5 Computational Load

As the fusion algorithm would be applied to multiple tracked objects, they need to be quick and simple to compute. The computation load will be measured in terms of time taken to compute the fused algorithm using each of these methods. However these computation times are obtained from `MatLab` and hence would not be optimal. They would be much lesser in an embedded implementation.

3.8.6 Data Transfer Size

When applying these methods in fusing data received over wireless communication, it should be kept in mind that the bandwidth of the communication is limited. Hence the information required by fusion algorithms should be limited.

3.9 Summary

To study the performance of the fusion algorithms in the next chapter, a simulation tool was setup in `MatLab/Simulink`. This setup can produce the ground truth of the states of the vehicles, the prior estimates (containing common information) as the output of the sensors and the fused estimates obtained from the sensor fusion algorithms. The sensor fusion algorithms chosen estimate the common information differently, which would help in choosing the most suitable method for V2V communication. Lastly, the evaluation methods presented would compare the prior estimates from the sensors with those from the sensor fusion algorithm.

Chapter 4

Simulation Results

Using the simulation setup presented in the previous chapter, the performance of the fusion algorithms in fusing data from synchronous and asynchronous sensors have been presented here. Also, the ability of the fusing algorithms to handle sensor failures (packet loss due to network congestion) will be studied. The aim is to compare two S2S methods -Covariance Intersection (CI) and Ellipsoidal Intersection(EI) with two S2G methods namely Information Matrix Fusion (IMF) and Covariance Intersection with Memory (CIM). Previous work done be N.J.G Koenraad compares EI and CI with Kalman Fusion and Gaussian in depth. Results from his work will be stated here too.

4.1 Ideal Situation

The sensors in this analysis are all synchronized and do not fail (packet loss due to network congestion). However, they have different uncertainties along longitudinal and lateral direction.

4.1.1 Average Root Mean Square Error

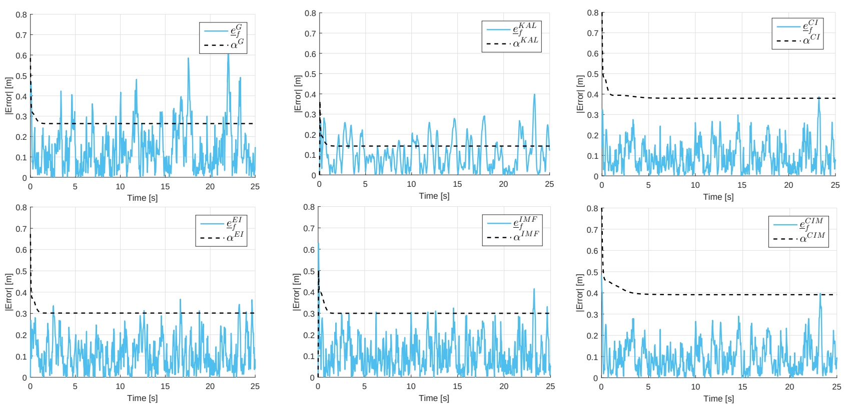

Figure 4.1: RMSE-Logitudinal position of Agent1 w.r.t Object1

In Figure.4.1 the error between the ground truth in longitudinal distance and fused estimate in longitudinal distance ($\underline{e}_f = |\underline{\bar{x}} - \underline{\hat{x}}_f|$) between the Agent1 and the Object obtained by fusing the prior estimates using the fusion algorithms for one cycle of Monte Carlo simulation has been plotted. Also, the 95% confidence region is plotted (dotted line). Though each cycle is for 100 sec., only 25 sec. of the simulation has been shown to ease reading.

It can be seen clearly that the errors in estimating the longitudinal distance by fusion using the methods CI, EI, IMF and CIM all lie within the 95% confidence region. This gives a hint that the methods are consistent unlike the Gaussian Distribution method and Kalman Filter method where the errors for a large part lie outside 95% confidence region.

$ARMSE.10^{-2}$	x	y	Σ
$\underline{e}_a$	2.65	2.49	5.14
$\underline{e}_b$	1.06	4.50	5.56
$\underline{e}_f^G$	2.70	3.65	6.35
$\underline{e}_f^{KAL}$	1.17	6.07	7.24
$\underline{e}_f^{CI}$	0.80	2.53	3.33
$\underline{e}_f^{EI}$	0.40	2.40	2.80
$\underline{e}_f^{IMF}$	0.69	1.98	2.67
$\underline{e}_f^{CIM}$	0.80	2.60	3.40

Table 4.1: Average Root Mean Square Error- Position

$ARMSE.10^{-2}$	$\dot{x}$	$\dot{y}$	Σ
$\underline{e}_a$	4.87	4.04	8.91
$\underline{e}_b$	3.12	5.29	8.41
$\underline{e}_f^G$	2.82	4.58	7.40
$\underline{e}_f^{KAL}$	3.48	6.43	9.91
$\underline{e}_f^{CI}$	2.76	4.24	7.00
$\underline{e}_f^{EI}$	2.05	4.02	6.07
$\underline{e}_f^{IMF}$	2.85	4.02	6.87
$\underline{e}_f^{CIM}$	2.84	4.34	7.18

Table 4.2: Average Root Mean Square Error- Velocity

To study the error further, the results obtained from 10000 Monte Carlo simulations are given in tables 4.1 and 4.2. CI, EI, IMF and CIM all have errors ($\underline{e}_f^{CI}, \underline{e}_f^{EI}, \underline{e}_f^{IMF}, \underline{e}_f^{CIM}$) less than that of the prior estimates ($\underline{e}_a = |\underline{x} - \underline{\hat{x}}_a|$ and $\underline{e}_b = |\underline{x} - \underline{\hat{x}}_b|$). This clearly shows that the fused estimates are better than the prior estimates from the sensor.

In terms of error in position (tables 4.1), IMF has the least error. This is because IMF makes a prediction and updates this prediction using the new measurements from the sensors unlike S2S methods which fuse only new measurements from the sensors. Also, IMF removes the common information in the prior estimates from the local track and that from the global track due to prediction. IMF however does poorly while fusing the velocity compared to EI as the effect of the process noise (acceleration) on velocity($\dot{x} \propto w_k \Delta T$) is considerably higher than on position ($x \propto w_k \frac{\Delta T^2}{2}, \Delta T = 0.04s$). This is because IMF does not consider the common process noise in the prior estimates resulting in greater error in case of estimating velocity.

From the tables(4.1 and 4.2) it can be noted that CIM has error similar to that of CI but higher even though it makes a prediction of the states like IMF. This is due to fact that CIM does not decorrelate the new measurement from the predicted measurement in the Global track like what IMF does, leading to the common information adding multiple times. CI does not have to deal with this as it does not keep a track of the previous data.

4.1.2 Covariance of Estimates

The uncertainty of the fused estimates can be studied from the covariance given in tables 4.3 and 4.4. The covariance of the fused estimates of CI (cov_f^{CI}), EI (cov_f^{EI}), IMF (cov_f^{IMF}) and CIM (cov_f^{CIM}) are smaller than that by prior estimate (cov_a and cov_b). Hence the performance of the above methods are better than the prior as smaller the covariance, more certain the estimates would be.

Both the CI methods are conservative (larger covariance) as both assume that the information from the sensors are completely correlated to each other and have no independent part. On the other hand, the KF method assumes the information sources, namely the prediction from previous step and the two estimates $\underline{\mathbf{x}}_a$ and $\underline{\mathbf{x}}_b$ are conditionally independent of each other resulting in very small covariance. However, IMF does not consider common information between the sensor measurements if process noise is low but accounts for the correlation between predicted values and the sensor estimates and hence it results in a covariance value which lies in between that of KF and CI. It can be noted out of the four methods proposed, IMF has the least covariance, resulting in more certain estimates.

$.10^{-2}$	x	y	Σ
cov_a	7.68	3.65	11.33
cov_b	1.80	10.27	12.07
cov_f^G	1.35	2.68	4.03
cov_f^{KAL}	0.30	0.21	0.51
cov_f^{CI}	3.28	4.72	8.00
cov_f^{EI}	2.40	3.65	6.05
cov_f^{IMF}	2.30	3.26	5.56
cov_f^{CIM}	4.04	4.92	8.96

Table 4.3: Covariance- Position

$.10^{-2}$	$\dot{x}$	$\dot{y}$	Σ
cov_a	53.21	5.48	58.69
cov_b	49.32	11.84	61.16
cov_f^G	23.38	3.74	27.12
cov_f^{KAL}	7.72	0.65	8.37
cov_f^{CI}	46.77	6.79	53.56
cov_f^{EI}	41.09	5.48	46.57
cov_f^{IMF}	34.37	5.63	40.00
cov_f^{CIM}	44.83	6.38	51.21

Table 4.4: Covariance- Velocity

4.1.3 Check for Consistency

Comparing Figures 4.2 till 4.7, we can study the consistency of each of the method (in estimating position). As defined in Section 3.8.3, the covariance of the error between the estimated value $\underline{\hat{x}}$ and true value $\underline{\bar{x}}$ should be bounded by the covariance matrix $\mathbf{P}$ of $\underline{\hat{x}}$. Figures 4.2 ,4.4 and 4.6 is the plot of the ellipse formed by the covariance matrix of the fused estimates and figures 4.3, 4.5 and 4.7 plots the ellipse formed by the True covariance which is the covariance of the error between the ground truth and prior or fused estimate ($cov(|\underline{\bar{x}} - \underline{\hat{x}}|)$) that can be obtained from the spread of 10000 samples. If the latter ellipse is smaller than the former, it confirms the algorithm is consistent and that the object tracked would actually lie within the estimated covariance ellipse. Both these sets of plots have mean of the distribution at 0 and prior estimates as $\mathcal{E}(\underline{0}, \tilde{\mathbf{P}}_a)$ and $\mathcal{E}(\underline{0}, \tilde{\mathbf{P}}_b)$.

Knowing this, it can be observed from the figures below that the two prior estimates $\mathbf{x}_a$ and $\mathbf{x}_b$ are consistent. Clearly KF and Gaussian Distribution methods (Figures 4.2 and 4.3) do not meet the consistency criteria as their true covariance is larger than the estimated covariance. CI and CIM are consistent as both have their estimated covariance larger than the true covariances (Figures 4.4 and 4.5). But in case of IMF and EI the ellipses appear to be of the same size (Figures 4.6 and 4.7). To check for consistency, (3.14) ($\mathbf{P} - \tilde{\mathbf{P}} \succeq 0$) is used to check if the resulting matrix is positive definite indicating it is consistent. This confirms EI and IMF are consistent too.

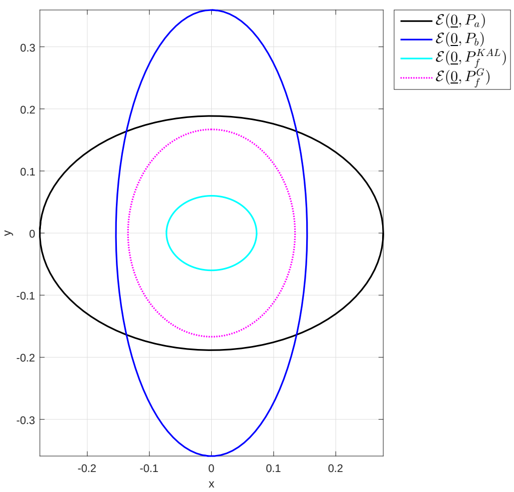

Figure 4.2: Estimated Covariance-KAL&Gaus

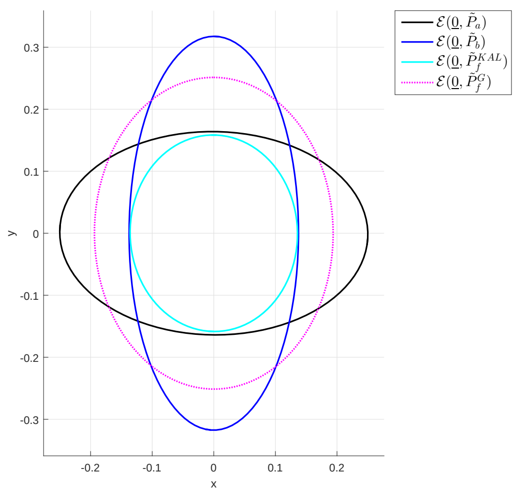

Figure 4.3: True Covariance-KAL&Gaus

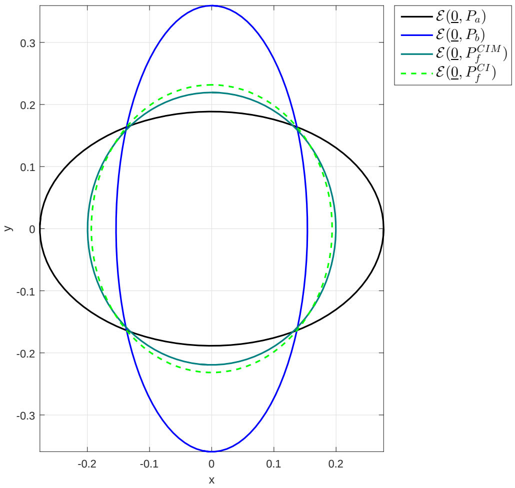

Figure 4.4: Estimated Covariance- CI&CIM

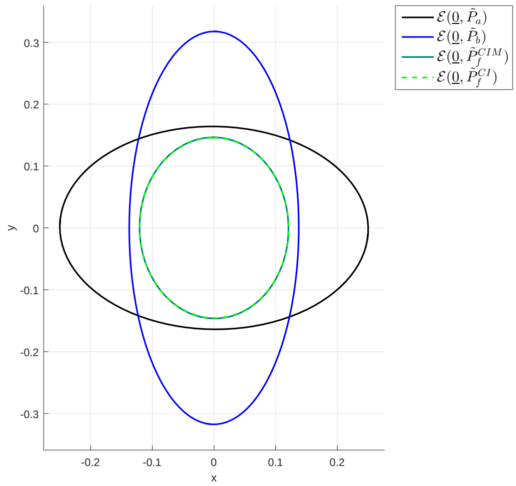

Figure 4.5: True Error Covariance- CI&CIM

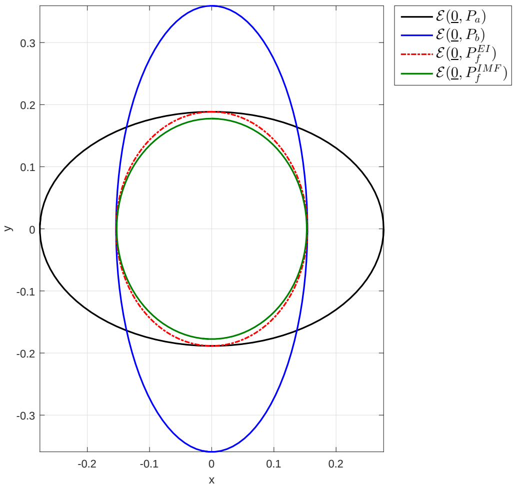

Figure 4.6: Estimated Covariance- EI&IMF

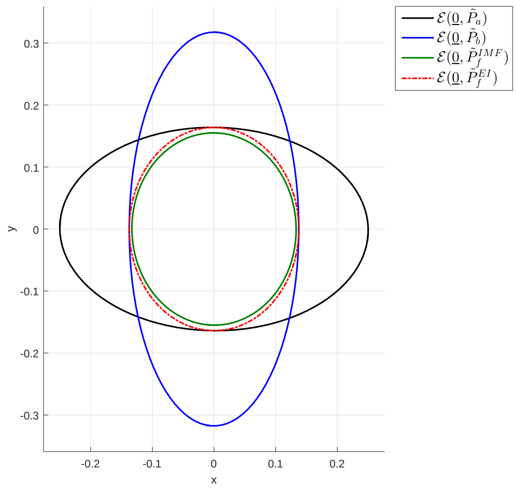

Figure 4.7: True Error Covariance- EI&IMF

4.2 Asynchronous Sensors/Out of Sequence

In section 4.1 it was assumed that the sensors were operating at the same frequency and measurements arrived in sequence. However, in real world applications, the sensors work at different frequencies as the local information is obtained and updated at different time samples [10]. This would result in a non-ideal situation for IMF as IMF works best when the communication is at full rate (estimates are communicated and fused after each observation and update time). Hence it is crucial to study effects of asynchronous sensors on the fused result.

In this simulation, one of the sensors was running at a lower frequency. The fusion algorithms would receive delayed information from one of the sensors. From tables 4.5 and 4.6 it can be observed that the IMF error has increased considerably with respect to the other methods (CI,EI,CIM). This confirms that IMF preforms best only at full rate communication.

$ARMSE.10^{-2}$	x	y	Σ
$\underline{e}_a$	20.02	12.49	32.51
$\underline{e}_b$	13.35	30.41	43.76
$\underline{e}_f^G$	28.16	26.96	55.12
$\underline{e}_f^{KAL}$	12.36	11.93	24.29
$\underline{e}_f^{CI}$	11.77	11.65	23.42
$\underline{e}_f^{EI}$	13.40	12.47	25.87
$\underline{e}_f^{IMF}$	15.42	13.27	28.69
$\underline{e}_f^{CIM}$	12.85	11.65	24.50

Table 4.5: Average Root Mean Square Error- Position

$ARMSE.10^{-2}$	$\dot{x}$	$\dot{y}$	Σ
$\underline{e}_a$	29.51	11.01	40.52
$\underline{e}_b$	25.83	16.25	42.08
$\underline{e}_f^G$	19.35	11.11	30.46
$\underline{e}_f^{KAL}$	13.78	10.36	24.14
$\underline{e}_f^{CI}$	18.53	10.31	28.84
$\underline{e}_f^{EI}$	25.91	11.40	37.31
$\underline{e}_f^{IMF}$	31.49	11.96	43.42
$\underline{e}_f^{CIM}$	19.06	10.38	29.44

Table 4.6: Average Root Mean Square Error- Velocity

4.3 Synchronizing using Prediction

Though synchronous sensors are hard to achieve, data to the fusion centre can be made to arrive at the same time by using a prediction model as given in Appendix 7.2. However, this would result in higher uncertainties which could result in an non-ideal situation for IMF leading to higher errors. Hence it is important to study the how predictions can effect the fusion results especially on IMF.

In this case the sensors are synchronized by predicting the measurements to the time of arrival at the fusion centre, making the communication full rate. Table 4.7 gives the error of the different fusion algorithms. Error in estimating position using IMF method is within the range of other fusion algorithms. This shows that it is essential that asynchronous or out of sequence sensor be synchronized before fusing in the case of IMF. However, it can be observed that the errors for all fusion algorithms have increased with respect to those obtained by fusing synchronous sensors.

$ARMSE.10^{-2}$	x	y	Σ
$\underline{e}_a$	2.55	3.19	5.74
$\underline{e}_b$	0.39	3.88	4.28
$\underline{e}_f^G$	2.15	3.68	5.84
$\underline{e}_f^{KAL}$	0.80	4.22	5.03
$\underline{e}_f^{CI}$	0.45	3.45	3.90
$\underline{e}_f^{EI}$	0.39	3.19	3.58
$\underline{e}_f^{IMF}$	0.60	2.65	3.25
$\underline{e}_f^{CIM}$	0.64	3.55	4.19

Table 4.7: Average Root Mean Square Error- Position

$.10^{-2}$	x	y	Σ
cov_a	12.65	4.65	17.30
cov_b	2.38	13.46	15.84
cov_f^G	1.99	3.45	5.45
cov_f^{KAL}	0.57	0.44	1.02
cov_f^{CI}	3.33	8.03	11.36
cov_f^{EI}	2.37	4.65	7.02
cov_f^{IMF}	2.65	3.23	5.57
cov_f^{CIM}	4.32	6.18	10.58

Table 4.8: Covariance-Position

4.4 Robustness

The previous three simulations did not include failure of sensors due to packet loss in communication. As stated in Section 3.8.4, it is important to study effects of sensor and communication failure.

IMF and CIM are Sensor to Global T2TF methods which retain past measurements as global tracks. If there is a loss of data, these global tracks are predicted to next time sample. EI and CI however fuse only current data and do not keep track of past measurements. In case it does not receive measurement, no fused estimates can be obtained. This could lead to large errors. Table 4.9 validates this claim. Both CIM and IMF which maintain global tracks preform better than EI and CI as their errors are smaller.

$ARMSE.10^{-2}$	x	y	$\dot{x}$	$\dot{y}$	Σ
$\underline{e}_a$	65.76	40.98	11.95	7.948	126.65
$\underline{e}_b$	1.168	4.49	3.50	7.38	16.56
$\underline{e}_f^G$	3.31	4.32	3.40	6.84	17.88
$\underline{e}_f^{KAL}$	1.43	4.91	3.68	7.67	17.70
$\underline{e}_f^{CI}$	1.17	3.32	3.31	6.54	14.36
$\underline{e}_f^{EI}$	1.17	3.29	3.55	6.39	14.39
$\underline{e}_f^{IMF}$	1.23	2.67	3.62	6.05	13.59
$\underline{e}_f^{CIM}$	1.16	3.07	3.23	6.37	13.84

Table 4.9: Average Root Mean Square Error- Position (with sensor failure)

4.5 Computational Time

Table4.10 compares the computation time required by each method. IMF seems to have the the least time compared to CI, EI and CIM as IMF only requires to invert the matrix and carry out simple mathematical operations such as addition and subtraction to find the fused estimates. EI uses eigen value decomposition and CI solves an optimization problem to find the fused estimates causing them to use more time. CI with memory solves two optimization problems to determine ω_1, ω_2 and ω_3 to find the fused estimate.

$Method$	$time$
$Gaus$	$1.61.10^{-4}s$
Kal	$2.31.10^{-4}s$
CI	$8.76.10^{-4}s$
EI	$7.66.10^{-4}s$
IMF	$2.58.10^{-4}s$
CIM	$11.14.10^{-4}s$

Table 4.10: Time Required for Computation

$Method$	$DataSize(kiloBytes/sec)$
$Gaus$	4
Kal	4
CI	4
EI	4
IMF	8
CIM	4

Table 4.11: Size of Data Transmitted

4.6 Amount of Data Transferred

The four states given in (3.2) will be considered.

CI, EI and CIM all require only the updated state estimate(4X1 vector) and the covariance matrix (4X4 Matrix) of the tracked object from the Kalman Filter of each sensor track to carry out fusion. However, IMF requires even the predicted state estimates and covariance matrix from the Kalman filter for decorrelation before fusing. Hence another set of 4X1 vector and 4X4 matrix would need to be communicated.

The calculation of the predicted states and covariance matrix can be carried out in the fusion centre too. However, the covariance of the process noise (4x4 matrix) would then need to be communicated to predict the covariance matrix as given in Appendix (7.1). Moreover, if the process noise is high, IMF requires the information such as Kalman Gains to decorrelate spatially before fusing. This is a major drawback of the IMF.

Assuming the data type is Double, each matrix is of size 128 Bytes and each vector is of size 32 Bytes. If the system is runnung at 25Hz, then per second the amount of data needed to be communicated is given in table 4.11

Chapter 5

Conclusions and Recommendations

5.1 Conclusion

The main goal of the project was to study the performance of sensor fusion methods that keep track of past fusion data (e.g. IMF and CIM) and compare them with methods that do not maintain past data (e.g. EI and CI). From the simulation, it can be concluded that IMF produces the best results when the process noise is very low. This is because it can predict what the states at the next time sample is and fuse it with the new measurement unlike S2S methods which rely only on new measurements from the sensors. However IMF performs only 4% better than the next best fusion method, EI.

IMF performs poorly when the process noise is high as in the case of estimation of velocity. We can also conclude from comparing CIM with the other methods that, if previous data is retained in the global track for fusing, the new estimates should be decorrelated using its prediction before fusing to prevent increase in error.

IMF also performs poorly when the sensors are not synchronized. However adding a simple prediction model to synchronize the tracks improves its performance even if the uncertainty increase due the prediction. In terms of consistency, both IMF and CIM are consistent as proven in the simulations.

One of the major advantages IMF and CIM that is proven in the simulation is that they are more robust to communication failures (such as packet loss due to network congestion) than EI and CI as the former two maintain past fusion data which can be used to predict the possible fused estimate when one of the sensors or communication from it fails.

Compared to the other sensor fusion methods such as EI, CI and CIM, IMF needs more information (two times to by exact) to carry out decorrelation of common information before fusion. This is not favourable when the mode of communication is wireless.

Considering only a small increase in performance compared to EI, IMF or any other fusion method that maintains past measurements (S2GF) may not be the best choice when communication bandwidth is limited as twice the amount of information would need to be transmitted to carry out decorrelation.

5.2 Recommendation

This project compares four sensor fusion methods in detail. Of these four, one of the methods, Information Fusion Method (IMF) has other variations than the one presented here. The presented version requires both the predicted and updated state estimates and error covariances from the Kalman filter to be transmitted. This is a major drawback of this method. [1] presents a method called Decentralized Minimum Information Method that can estimate the predicted state and covariance hence reducing the amount of information required to be transferred. This method can be considered for further researched for sensor fusion application in V2V communication.

In scenarios such as overtaking or lane change, the vehicle will most likely not be travelling at a constant velocity and hence a constant acceleration model is preferred over a constant velocity model as used in this project. As seen in [1], sensor fusion results may vary when using constant velocity model and constant acceleration model. Also, if a non-linear model is used, the model can be linearised form by Taylor approximation [1].

Chapter 6

Bibliography

[1]. Eriksson, Viktor. "Evaluation of Decentralized Information Matrix Fusion for Advanced Driver-Assistance Systems in Heavy-Duty Vehicles." Student Master Level Thesis, 2016.

[2]. Tian, Xin, Ting Yuan, and Yaakov Bar-Shalom. "Track-to-Track Fusion in Linear and Nonlinear Systems." Advances in Estimation, Navigation, and Spacecraft Control. Springer Berlin Heidelberg, 2015. 21-41.

[3]. Matzka, Stephan, and Richard Altendorfer. "A comparison of track-to-track fusion algorithms for automotive sensor fusion." Multisensor Fusion and Integration for Intelligent Systems. Springer Berlin Heidelberg, 2009. 69-81.

[4]. Chong, Chee-Yee, et al. "Architectures and algorithms for track association and fusion." IEEE Aerospace and Electronic Systems Magazine 15.1 (2000): 5-13.

[5]. Aeberhard, Michael, et al. "Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception." IEEE Transactions on Intelligent Transportation Systems 13.4 (2012): 1717-1726.

[6]. Noack, Benjamin, et al. "Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion." Multisensor Data Fusion: From Algorithms and Architectural Design to Applications. Ed. Hassen Fourati. CRC Press, 2015. 169-192.

[7]. Bar-Shalom, Yaakov. "On the track-to-track correlation problem." IEEE Transactions on Automatic Control 26.2 (1981): 571-572.

[8]. Liu, Feng, Jan Sparbert, and Christoph Stiller. "IMMPDA vehicle tracking system using asynchronous sensor fusion of radar and vision." Proc. IEEE Intelligent Vehicles Symposium. 2008.

[9].Lin, Xiangdong, Yaakov Bar-Shalom, and Thiagalingam Kirubarajan. "Multisensor multitarget bias estimation for general asynchronous sensors." IEEE Transactions on Aerospace and Electronic Systems 41.3 (2005): 899-921.

[10]. Tian, Xin, and Yaakov Bar-Shalom. "On algorithms for asynchronous track-to-track fusion." Information Fusion (FUSION), 2010 13th Conference on. IEEE, 2010.

[11]. Chong, Chee-Yee. "Hierarchical estimation." Proc. MIT/ONR Workshop on C3. 1979.

[12]. Chang, Kuo Chu, Tian Zhi, and Rajat K. Saha. "Performance evaluation of track fusion with information matrix filter." IEEE Transactions on Aerospace and Electronic Systems 38.2 (2002): 455-466.

[13]. Chang, Kuo-Chu, Rajat K. Saha, and Yaakov Bar-Shalom. "On optimal track-to-track fusion." IEEE Transactions on Aerospace and Electronic Systems 33.4 (1997): 1271-1276.

[14]. Julier, Simon J., and Jeffrey K. Uhlmann. "A non-divergent estimation algorithm in the presence of unknown correlations." In Proceedings of the American Control Conference. 1997.

[15]. Li, Hao, Fawzi Nashashibi, and Ming Yang. "Split covariance intersection filter: Theory and its application to vehicle localization." IEEE Transactions on Intelligent Transportation Systems 14.4 (2013): 1860-1871.

[16]. Sijs, Joris, and Mircea Lazar. "State fusion with unknown correlation: Ellipsoidal intersection." Automatica 48.8 (2012): 1874-1878.

[17].Li, Hao, et al. "Track-to-track fusion using split covariance intersection filter-information matrix filter (scif-imf) for vehicle surrounding environment perception." 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013.

[18]Liu, Feng, Jan Sparbert, and Christoph Stiller. "IMMPDA vehicle tracking system using asynchronous sensor fusion of radar and vision." Proc. IEEE Intelligent Vehicles Symposium. 2008.

[19]Biswas, Subir, Raymond Tatchikou, and Francois Dion. "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety." IEEE communications magazine 44.1 (2006): 74-82.]

[20]Fernandes, Pedro, and Urbano Nunes. "Platooning with IVC-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow." IEEE Transactions on Intelligent Transportation Systems 13.1 (2012): 91-106.

[21]J.G.Koenraad, Jos Elfring and J.G van de Molengraft, "Multi-Agent decentralized Track-to-Track Fusion". Master Student Thesis. Not Published Yet.

Chapter 7

Appendix

7.1 Gaussian Distribution

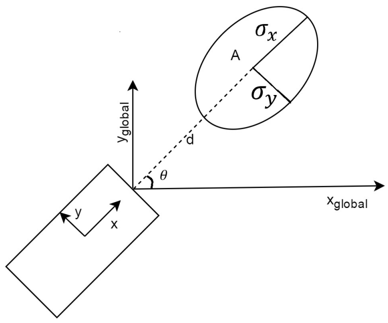

Figure 7.1: Gaussian Distribution of a sensor measurement as observed by a vehicle.Point A is d away from the vehicle and σ_{max} and σ_{min} are the uncertainties in the x and y axis of the vehicle.

Information from sensors (such as position and velocity of object) will be modelled as a Gaussian distribution ($\underline{\mathbf{x}} \sim N((\underline{\hat{x}}), \mathbf{P})$) which contains a mean $\underline{\hat{x}}$ (most likely estimate) and covariance matrix $\mathbf{P}$ (uncertainty of the estimate). The Gaussian distribution is a continues probability distribution function that gives the distribution (variance) of the data about its mean. Larger the variance, more the uncertainty. This is perfect to represent noisy sensor measurement such as distance as the true measurement is unknown due to noise. In Figure 7.1, point A is d away from the vehicle and σ_x and σ_y are the uncertainties in the x and y axis respectively of the vehicle frame of reference.

7.2 Prediction Model

$$\begin{aligned} \underline{x}_{(k|k-1)} &= \mathbf{A}\underline{x}_{(k-1|k-1)} \\ \mathbf{P}_{(k|k-1)} &= \mathbf{A}\mathbf{P}_{(k-1|k-1)}\mathbf{A}^T + \mathbf{Q}_{(k)} \end{aligned} \tag{7.1}$$

The sensor measurements are synchronized in time by prediction using the equations given in (7.1). Here $x_{(k|k-1)}$ and $P_{(k|k-1)}$ represent the state and covariance matrix predicted to time k using information until time $k-1$, $\mathbf{A}$ is the state transformation matrix in the constant velocity model as given in Equation, $\underline{x}_{(k-1|k-1)}$ and $\mathbf{P}_{(k-1|k-1)}$ represent the previously fused state vector and covariance matrix at time $k-1$ containing information until time $k-1$ and $\mathbf{Q}_{(k)}$ is the process noise covariance matrix at time k .