CALCULATION OF COLLISION PROBABILITY FOR AUTONOMOUS VEHICLES USING TRAJECTORY PREDICTION

By

GAYATRI POWAR

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

written under the direction of

Prof. Zoran Gajic

and approved by

____

____

____

New Brunswick, New Jersey

October, 2016

ABSTRACT OF THE THESIS

Calculation of Collision Probability for Autonomous Vehicles using Trajectory Prediction

By GAYATRI POWAR

Thesis Director:

Prof. Zoran Gajic

The aim of this thesis is to create a decision making algorithm. The goal would be to check the feasibility of the current maneuver by finding the probability of collision between the subject and target vehicles. These outputs can be used by other Advanced Driver Assistance System (ADAS) features including path planner, lateral control, longitudinal control, etc. We make use of some sensors like camera/radar (simulated data) and fuse these together for better estimation of measurements. With earlier experience with cameras, they are really poor at giving longitudinal distances whereas radars give more accurate measurements longitudinally. Using these measurements about targets ahead in the environment, we predict the trajectories of the obstacles/targets as well as the subject vehicle (autonomous vehicle). The algorithm predicts if it is safe to continue with the current maneuver in the near future for several seconds ahead of time, and makes the decision if the maneuver is possible. The results are obtained using probabilistic approach whether the future trajectories are going to collide. The thesis primarily focuses on target tracking, efficient sensor data fusion and future collision estimation. With the lessons learnt using existing literature an efficient approach is employed.

The simulation is performed in PreScan simulator and MATLAB. Enhancements in the sensor data fusion using standby measurements and quasi-decentralized approach to combine measurements to yield improved results and achieve better scalability are proposed and implemented.

Acknowledgments

I would never have been able to finish my thesis without the guidance of my adviser and committee members, help from friends, and support from my family.

I would like to express my deepest gratitude to my adviser, Prof. Zoran Gajic, for his excellent guidance, patience and his valuable time. His mentorship was exceptional in providing a well rounded research experience consistent with my long-term career goals. I was given the opportunity to develop my own individuality and self-sufficiency by being allowed to work with enough independence.

I would also like to thank Magna Electronics for the unique internship opportunity to work on autonomous cars. This helped me gain a wider breadth experience to strengthen my academic background. Special thanks to Mr. Jeff Blackburn from TASS International for providing me with the PreScan software for simulation and development.

Finally I thank my parents, for their faith in me and allowing me to be as ambitious as I wanted. They have always been supportive and encouraging. And last but not the least, cheers to my friends Amit, Ankit and Suvigya who cheered me up and stood by me through the good times and bad.

Dedication

Dedicated to my family and friends

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Active safety systems are a promising solution to reduce a number of road accidents. Some Advanced Driver Assistance systems (ADAS) such as Adaptive Cruise Control, Collision Warning Systems and Emergency Braking Systems that exist in vehicles are able to warn drivers and also interfere on the state of the vehicle when a hazardous traffic situation is developed.

In this thesis, an algorithm is created, which decides the feasibility of a maneuver and provides information to the driver about the probability of collision in future. For this purpose, the algorithm has to first rely on accurate perception system that uses sensors, and data fusion techniques to track obstacles in the surroundings of the subject vehicle. Secondly, it is required to predict the trajectory of the detected vehicle in the surroundings of the subject vehicle and the trajectory of the subject vehicle. Using this information, the probability of collision has to be calculated.

Predicting the trajectory of a vehicle is not a deterministic task. Hence a probabilistic approach is used to account for the uncertainty of prediction. However, in many cases, trajectory prediction is made by assuming a certain vehicle model. It appears that linear motion models like constant velocity (CV) and constant acceleration (CA) have a major advantage of linearity of state transition equations which allows an optimal propagation of state probability distribution. Fast and reliable knowledge of moving objects and their trajectory prediction relative to subject vehicle is the main contribution of the thesis. Along with the algorithm implementation, the thesis also offers two enhancements to be applied to improve target tracking accuracy and optimal sensor data fusion.

1.2 Motivation

Motor vehicle travel is the primary means of transportation, providing an unparalleled degree of mobility. Yet for all its advantages, motor vehicle crashes in 2013 were the leading cause of death among individuals. Traffic Safety Facts in [11] indicate that in 2013, there were 32,719 people killed in the estimated 5,687,000 police-reported motor vehicle traffic crashes. An average of 90 people died each day in motor vehicle crashes in 2013, one fatality every 16 minutes. Reports suggests that majority of the crashes occur due to poor judgement and speeding.

Active safety systems can help drivers in preventing such collisions. They can provide the driver with information as to where other traffic participants and obstacles are located and how they are progressing. Different methods have been studied to tackle this problem. The two main strategies are telematics and sensing. In telematics, neighbourhood vehicles and infrastructure send information about their position, velocity or status (red light signal, turn status, etc.). This is better described as V2X(Vehicle to Vehicle, Vehicle to Infrastructure) communication. The advantage of this method is that the information is accurate. But a major disadvantage is that not all the objects on the road are equipped with the necessary hardware. Also there are other issues like communication security, data integrity that make them vulnerable to threats.

The other approach is to equip the subject vehicle with sensors like radar, camera, laser scanner, for perception of the environment. These systems do not have to rely on neighbouring vehicles. But the major challenge in this approach is that these systems have to deal with extracting reliable information, cater to bad weather conditions like rain, fog. In this thesis, we try to detect and track the neighboring vehicles using a monocular camera and a radar sensor and use this information to find the motion prediction.

1.3 Related Work

Various threat assessment techniques have been studied previously by many researchers. Generally, the problem is solved using the following steps: Estimation of current vehicle states, trajectory prediction and evaluation of risk of collision. Paper [2] calculates, based on location information, whether collision will occur or not based on the closed point of approach (CPA). Time and distance to CPA is calculated to assess risk of collision.

Recently, many researchers have focused on researching of new concepts of risk estimation by considering uncertainties in prediction. The papers [3], [6] are based on this approach. The paper [3] presents an approach to access the risk of collision of the subject vehicle by object trajectory prediction collision. The author presents a method to propagate the uncertainty of the pose of the vehicle along its predicted trajectory. This thesis also makes use of a similar approach.

Several methods of future trajectory prediction are studied over several years. Some research works propose to use a polynomial representation of trajectories. In [12], first, third and fifth degree polynomials are used to model the future trajectory. Some researchers make use of clothoids to model the vehicle trajectory [13]. These methods lack to totally determine the behavior of the vehicle. The papers [3], [6] assume a kinematic model to predict the trajectory of the vehicle. Different inputs can be considered like acceleration, speed, brake, acceleration pressure, etc. The advantage of having a vehicle model makes this method suitable for tracking problems.

Paper [10] deals with different vehicle models for vehicle tracking. The following models have been described: Constant Velocity (CV), Constant Acceleration (CA), Constant Turn Rate and Velocity (CTRV), Constant Turn Rate and Acceleration (CTRA) models. It is suggested that CTRA model performs better in tracking applications. But the increased performance also comes with added computation costs. In the thesis, the constant acceleration model is chosen because of its simplicity.

Paper [14] describes the necessity of multiple sensors in tracking applications. It describes the basic information about the use of cameras and radars and emphasis is on sensor data fusion. In [5], different techniques of sensor data fusion using multi sensors are introduced as follows:

1. Group Sensor Fusion Method: This method works similar to a single sensor Kalman filter. The observation model is modelled using models of all the available sensors.

2. Sequential Sensor Model: In this method, each sensor is modelled separately. Realization is made independently for each of them and then all estimations are combined in a sequential manner.

3. Inverse Covariance Form: This method explicitly derives equations for integrating multiple observations made at the same time into a common state estimate. Then, a set of recursive equations is derived for integrating individual sensor observations.

The book [1] gives a more in depth details of each of the above methods. Further paper [4] compares the above mentioned methods by comparing the computational complexity. Using MATLAB, computational loads of these methods are compared while the number of sensors are increased. The results show that the inverse covariance method has the best computational performance if the number of sensors is above twenty. For a smaller number of sensors, the group sensor fusion model is the most efficient and appropriate.

Paper [17] suggests that use of standby measurements in addition to actual measurements improves the estimation accuracy and minimizes the covariance error. The paper [17] is based upon a continuous time system. Based on the proposed findings a discrete time counterpart is implemented in the thesis. Research suggests that the optimal solution for estimation problems with different information sets requires an infinite dimensional Kalman filter which is not applicable from a practical point of view. Research in [20] demonstrates the solution approximately using constrained estimators specifying in advance the estimator dimensions. An alternative method is proposed in papers [21], [22] and [23] where the central idea is to rearrange the optimal centralized Kalman filter to be implemented via a decentralized scheme. This method is known as quasi - decentralization. The paper [19] presents a simplified method of quasi - decentralized estimation and control. Here, local non optimal Kalman filters are used for generation of global estimates by simple addition of local non-optimal estimates. In this thesis, we exploit this approach of quasi - decentralization to obtain a better estimate of the position of the subject and target vehicle. A theoretical and practical comparison of the Group Sensor Fusion (Centralized Sensor Data Fusion) and Quasi - Decentralized Data Fusion is presented in the thesis.

1.4 Thesis Objective

This thesis aims at estimation of risk related to collisions between subject vehicle and static and dynamic objects in the surroundings. This estimation of risk is achieved using three sequential actions: environment perception, trajectory prediction and collision estimation. This steps are illustrated in the Figure 1.1.

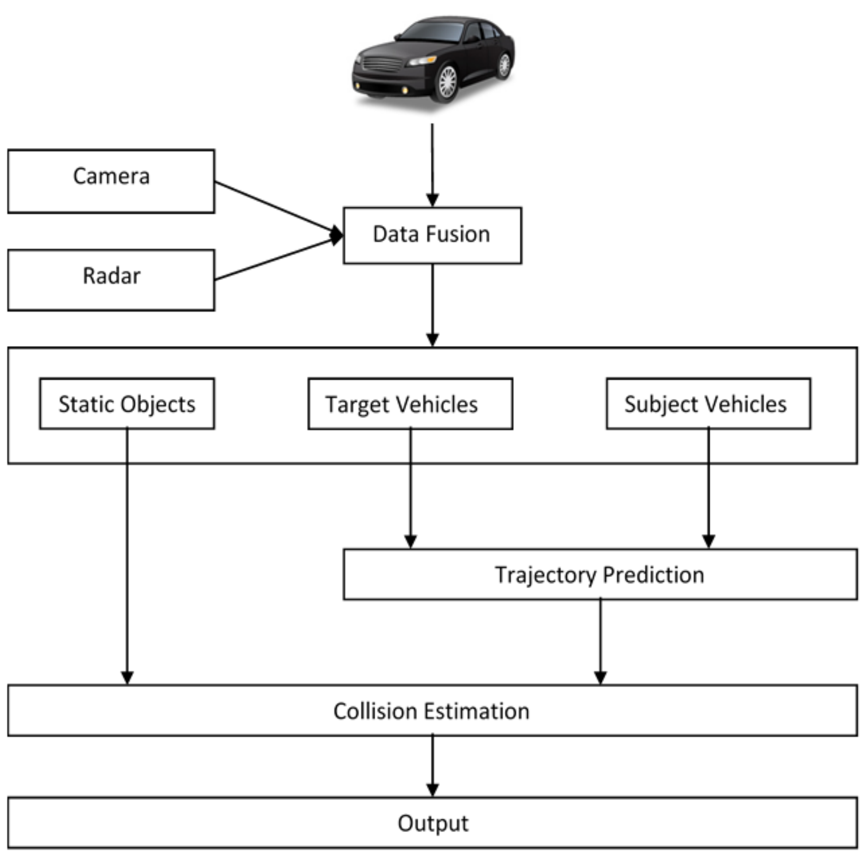


Figure 1.1: System Block Diagram

The subject vehicle is equipped with radar and front camera. The data fusion module receives data from sensors and estimates the states of the target vehicles. The knowledge of the vehicle is limited quantitatively by the range of its sensors and qualitatively by the precision and the richness of the delivered information. This module gives information of the current state of the environment. Using the information about the current state of the subject vehicle, target vehicle and surrounding static objects, a prediction of the future states is made using trajectory prediction module. Prediction of the future behavior of the driver is out of the scope of this thesis. In fact, any change in the drivers behavior influences the inputs of the system and consequently the future trajectory of the vehicle.

After prediction stage, the future trajectories of the subject vehicle and target vehicle are utilized to compute the possible collision between each trajectories. This estimation can be used by other ADAS systems to inform the driver about impending collision or other collision mitigation systems, etc. The next sections describe the crux of the algorithm used.

In addition to the algorithm implementation, the thesis offers two enhancements using existing research. Firstly, the usage of standby measurements to increase prediction accuracy and minimize error covariance is recommended and implemented. Furthermore, combining the estimates of local non optimal filter using quasi-decentralized approach to obtain global optimal Kalman filter for sensor data fusion is applied to improve target tracking accuracy. Also the comparison of sensor data fusion architectures is presented.

1.5 Organization of the Thesis

The overview and motivation of the thesis is described in the beginning of Chapter 1 followed by the various research and related work. Chapter 2 highlights the technical background behind the thesis. This includes the niceties about the theoretical concepts used in the implementation of the algorithm. In depth details regarding various concepts that were chosen and explored like Sensor Data Fusion, Trajectory Prediction, Propagation of Uncertainty and Collision Detection. The subsequent Chapter 3 speaks about the implementation of the algorithm developed in PreScan Simulator and MATLAB/Simulink. The entire concept is simulated over two scenarios and the results are demonstrated in Chapter 4. This chapter also provides results of several other important concepts that have been implemented. Finally, the Chapters 5 and 6 provides the thesis conclusion and future research respectively. The list of references used during the research are provided at the finish of the thesis document.

Chapter 2

Technical Background

2.1 Sensor and Sensor Data

There are variety of sensors providing information about the subject vehicle. The subject vehicle is considered to have following sensors:

- Vision Camera Sensor mounted on the windshield of the subject vehicle
- Radar sensor installed on the front grill

The advantages and disadvantages of both these sensors are summarized in Table 2.1

As seen in Table 2.1, both these sensors have orthogonal advantages and disadvantages which makes them well suited for sensor fusion. Hence, combining data from both the sensors, we get a solution that is more robust.

2.2 Vehicle Dynamic Models

It is difficult to realize continuous models because computer control algorithms work at discrete time steps and measurements and are always available at certain frame rates. In the case of a target

	Camera	Radar
Range	Poor	Accurate
Range Rate	Not Available	Directly available
Angle Resolution	Good	Poor
Velocity Resolution	Accurate	Poor
Weather Sensitivity	Sensitive to Weather	Insensitive to weather
Lateral Resolution	Accurate	Poor

Table 2.1: Camera and Radar Comparison

tracking application the input is also unknown so there is no control input *u(k)*. Hence we will be concerned with the estimation (following) problem only. The discrete time system is represented as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{2.1}$$

where,

x(k) is the state vector $x(k) \in R^n$

y(k) is the output vector $y(k) \in R^q$

A is the state or system matrix dim$[(A(.)] = n$ x n

C is the output matrix dim$[C(.)] = q$ x n

$w(k)$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance *Q(k)* $\quad w(k) \sim \mathcal{N}(0, Q(k))$

$v(k)$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $R(k) \quad v(k) \sim \mathcal{N}(0, R(k))$

According to [1], "Tracking is the estimation of the state of a moving object based on remote measurements. It maintains the objects state and identity despite detection errors and in presence of other objects." The target tracking cycle is represented in Figure 2.1

Different target motion models are used for target tracking like the Brownian model, constant velocity model, constant acceleration model, etc.

2.2.1 Constant Velocity Model

. In this model the target velocity is assumed to be constant. This model is useful in cases when the target motion is smooth. The system state vector and the system matrix are respectively given by

$$x = \begin{bmatrix} x & y & V_x & V_y \end{bmatrix}^\intercal \tag{2.2}$$

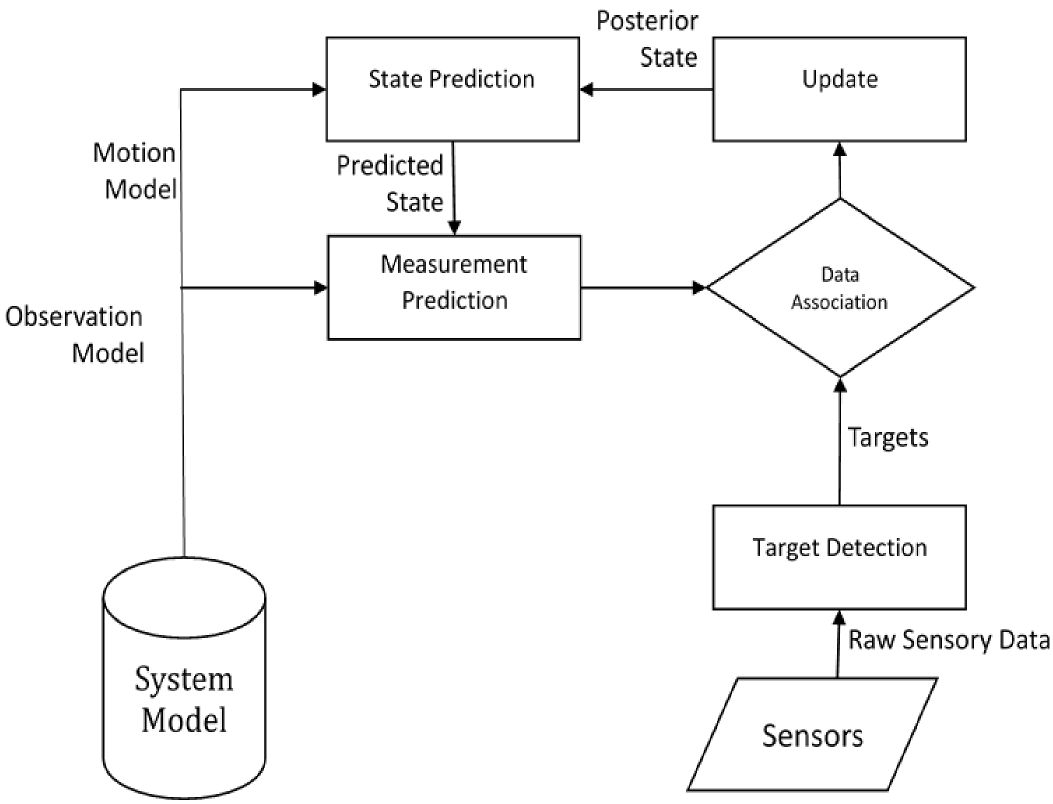


Figure 2.1: Target Tracking Cycle

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

where T is the sampling period.

2.2.2 Constant Acceleration Model - Ramachandras Model

This model consists of position, velocity and acceleration as the states of the system in 2D coordinate system assuming constant acceleration. This is useful when target motion is smooth in position and velocity. The state vector is given by

$$x = \begin{bmatrix} x & y & V_x & V_y & a_x & a_y \end{bmatrix}^{\intercal} \tag{2.4}$$

Each position coordinate of the moving vehicle is described as follows:

$$\begin{aligned} x_{k+1} &= x_k + V_{x_k}.T + a_{x_k}.\frac{T}{2} \\ V_{x_{k+1}} &= V_{x_k} + a_{x_k}.T \\ a_{x_{k+1}} &= a_{x_k} \end{aligned} \tag{2.5}$$

Similarly,

$$\begin{aligned} y_{k+1} &= y_k + V_{y_k}.T + a_{y_k}.\frac{T}{2} \\ V_{y_{k+1}} &= V_{y_k} + a_{y_k}.T \\ a_{y_{k+1}} &= a_{y_k} \end{aligned} \tag{2.6}$$

The corresponding system matrix is given by

$$A = \begin{bmatrix} 1 & 0 & T & 0 & T^2 2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2 2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

The quantities measured are positions and velocities in x and y directions. So the output matrix of the model defined in 2.7 is given by

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.8}$$

2.3 Kalman Filter

This section presents a basic description of the Kalman filter algorithm. This is based on the book [1]. The fundamental assumptions of the Kalman Filter are:

- Evolution of system state according to known linear plant dynamics driven by a known input and an additive process noise with zero mean with known covariance *Q(k)*.
- Measurement is a known linear function of the state with an additive measurement noise with zero mean with known covariance *R(k)*.

2.3.1 Algorithm Flow

At each step, the entire past is summarized by the state estimate $x\hat{(}k|k)$ and the associated covariance $P(k|k)$. The state estimation cycle consists of two parts:

1. State and measurement prediction (also called time update)
2. State update (also called measurement update)

In Figure 2.2 One cycle in the state estimation of a linear system is explained.

2.4 Coordinate Conversion in Target Tracking

2.4.1 The Conversion Problem

In case of sensors like radar or sonar the measurements that are available from target are in polar coordinates ie. range and azimuth or bearing. The range that is measured is the slant range. In most tracking problems the measurements are modelled as Cartesian coordinates. In order to perform the conversion the standard Kalman Filter is modified. The linear Kalman Filter with measurements converted to a Cartesian frame of reference is known as the Converted Measurement Kalman Filter (CMKF). In this filter the converted measurement covariance is recalculated at each sampling interval.

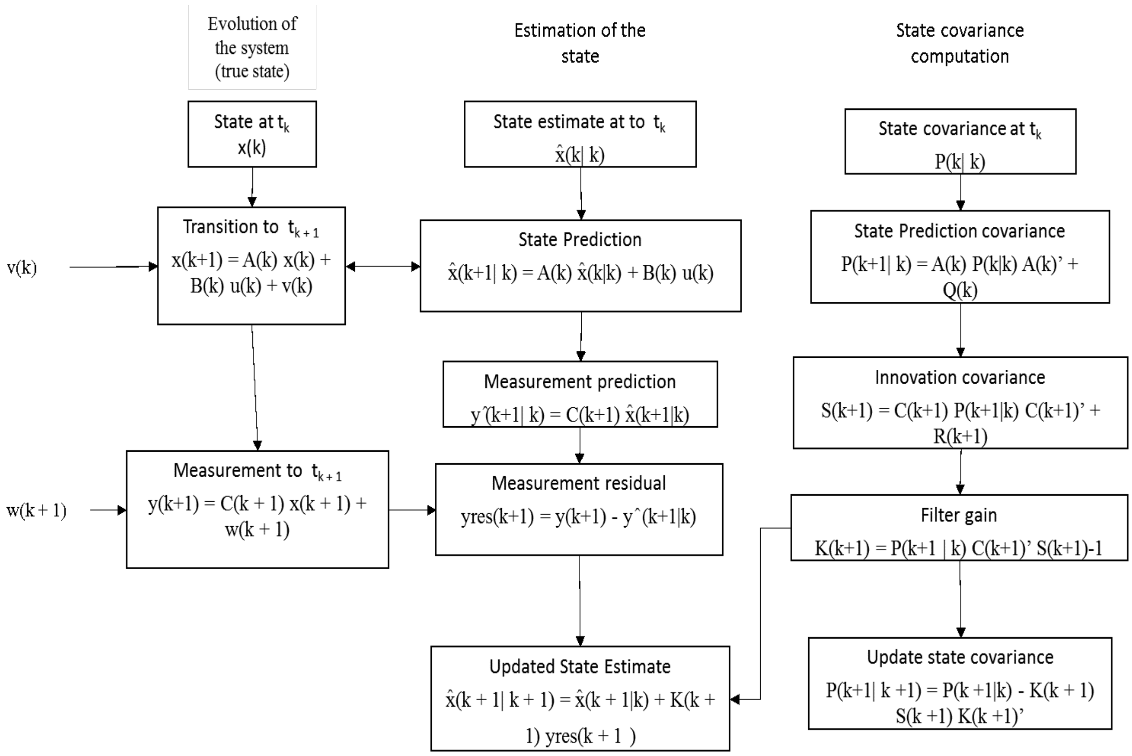


Figure 2.2: One cycle in the state estimation of a linear system

2.4.2 Errors in Conversion

Let r_m and θ_m be the measured range and measured azimuth respectively defined with respect to the true range r and true azimuth θ given as

$$\begin{aligned} r_m &= r + \tilde{r} \\ \theta_m &= \theta + \tilde{\theta} \end{aligned} \tag{2.9}$$

where the errors $\tilde{r}$, $\tilde{\theta}$ are assumed to be independent with zero mean and standard deviation σ_r and σ_θ respectively.

2.4.3 The Standard Conversion

The polar measurements are transformed to Cartesian by the standard coordinate conversion as follows

$$x_m = r_m \cos\theta_m$$
$$y_m = r_m \sin\theta_m \tag{2.10}$$

2.4.4 Converted Measurements for Kalman Filter

Denoting by (x,y) the true Cartesian position and taking the first order terms of the Taylor series expansion of equation 2.10, which is linearization yields the Cartesian coordinate errors

$$x_m - x \approx \tilde{r}\cos\theta_m - \tilde{\theta} r_m \sin\theta_m \triangleq \tilde{x_L}$$
$$y_m - y \approx \tilde{r}\sin\theta_m + \tilde{\theta} r_m \cos\theta_m \triangleq \tilde{y_L} \tag{2.11}$$

The mean of the errors is zero and given as

$$\mu \triangleq \begin{bmatrix} E\big[\tilde{x_L}\big] \\ E\big[\tilde{y_L}\big] \end{bmatrix} = 0 \tag{2.12}$$

The elements of the covariance matrix for the Kalman filter are given as

$$R_L^{11} \triangleq var\big(\tilde{x_L}\big) = r_m^2\sigma_\theta^2 \sin^2\theta_m + \sigma_r^2\cos^2\theta_m$$
$$R_L^{22} \triangleq var\big(\tilde{y_L}\big) = r_m^2\sigma_\theta^2 \cos^2\theta_m + \sigma_r^2\sin^2\theta_m \tag{2.13}$$
$$R_L^{12} \triangleq cov\big(\tilde{x_L}, \tilde{y_L}\big) = \big(\sigma_r^2 - r_m^2\sigma_\theta^2\big)\sin\theta_m\cos\theta_m$$

With the help of this conversion the consistency of measurements is obtained in the measured values. This modified Kalman filter has better estimation accuracy than the standard conversion when used in target tracking application

2.5 Sensor Data Fusion

Sensor data fusion is the process of combing information from a number of different sensors to provide a robust and complete description of an environment or the process of interest. Sensor data fusion is of special significance in any application where a large amounts of data from different sensors must be combined, fused and distilled to obtain information about appropriate quality and integrity on which decisions can be made. It is concerned with the fusion of information from sensors to arrive at an estimate of location and identify objects in an environment. In this thesis we assume that the data received from the sensors is synchronous i.e. at a given sampling time there is only one measurement coming from each sensor.

Data Fusion techniques can be classified into two main types depending upon the data fusion architecture as centralized and decentralized data fusion. This is mainly based on how the sensor data is processed and the state estimation is carried out. A theoretical and practical comparison of both the types is evaluated in this thesis.

- **Centralized Data Fusion** : In the centralized data fusion system, the raw sensor data is communicated to a central processor. This central processor fuses the data to form the estimate of the states. Every sensor has its own observation model but the process model is identical. All the observations are sent in unprocessed form to the central fusion center that combines the observations to provide an estimate of the state. The estimation is carried out in the same way as single sensor system. Based on the research in Paper [4] and Book [1], for smaller number of sensors the Group Sensor Fusion Model is efficient and appropriate. This method is chosen in our comparison of the data fusion techniques.

- **Decentralized Data Fusion** : In the decentralized data fusion system, each sensor has its own local sensor filter which processes the raw sensor data and calculates the local estimates that are sent to the central processor. In this case thus less information is communicated to the central processor. These types of fusion techniques have the advantage of modularity but the algorithms are quite complex.

- **Quasi-Decentralized Data Fusion** : Based on research discussed earlier, the result of combining data from local optimal filters may not be optimal. An infinite dimensional filter would be required in order to achieve an optimal estimate. As proposed in Papers [21], [22] and [23], the central idea is to rearrange the optimal centralized Kalman filter to be implemented via a decentralized scheme. This method is known as quasi - decentralization.

2.5.1 Centralized Sensor Data Fusion (The Group-Sensor Method)

The single most important problem in sensor data fusion is estimation. Essentially, an estimator is a function which takes as an argument a sequence of observations and whose action is to find an estimate for a parameter or state of interest. Almost all data fusion methods involve the estimation process: we obtain a number of observations from a group of sensors and using these observations we wish to find an estimate of the true state of the environment state variables that we are observing. The Kalman filter serves as the basis for most multi-sensor estimation problems. Sensor models are required to understand what information is provided, environment models are required to relate observations made to the parameters and states to be estimated, and some concept of information value is needed to judge the performance of the estimator. Defining and solving an estimation problem is almost always the key to a successful data fusion system.

The simplest way of dealing with a multi-sensor estimation problem is to combine all observations and observation models in a single composite group sensor and then to deal with the estimation problem using an identical Kalman filter algorithm to that employed in single-sensor systems. Defining the composite observation vector by

$$y(k) = \begin{bmatrix} y_1^{\mathsf{T}}(k) & y_2^{\mathsf{T}}(k) & y_3^{\mathsf{T}}(k) & ... & ... & y_n^{\mathsf{T}}(k) \end{bmatrix}^{\mathsf{T}} \tag{2.14}$$

and the composite observation model vector as

$$C(k) = \begin{bmatrix} C_1^{\mathsf{T}}(k) & C_2^{\mathsf{T}}(k) & C_3^{\mathsf{T}}(k) & ... & ... & C_n^{\mathsf{T}}(k) \end{bmatrix}^{\mathsf{T}} \tag{2.15}$$

with the noise vector

$$v(k) = \begin{bmatrix} v_1^{\mathsf{T}}(k) & v_2^{\mathsf{T}}(k) & v_3^{\mathsf{T}}(k) & ... & ... & v_n^{\mathsf{T}}(k) \end{bmatrix}^{\mathsf{T}} \tag{2.16}$$

It is assumed that the observation noise covariance is block diagonal with the blocks equal to the individual sensor observation noise covariance matrices. This set of observations defined is written as a single observation model as

$$y(k) = C(k)x(k) + v(k) \tag{2.17}$$

$$R(k) = E\{v(k)v^{\mathsf{T}}(k)\} = \text{blokdiag}\{R_1(k), R_2(k), ..., R_n(k)\} \tag{2.18}$$

The prediction phase of the multi-sensor Kalman filter algorithm makes no reference to the observations that are made and so is identical in every respect to the prediction phase of the single-sensor filter. However, the update phase of the cycle, which incorporates measurement information from sensors, will clearly be affected by an increase in the number of sensors. As per [4], the group sensor is the best suited model for a few number of sensors since the output is accurate and computationally less efficient.

2.5.2 Quasi - Decentralized Estimation

Consider a discrete time, linear time invariant (LTI) system without any exogenous input, described by state space model given in (2.1), which for the reason of convenience is repeated here

$$x(k+1) = Ax(k) + w(k)$$

where $x(k) \in R^n$ is the state vector and $y_i(k) \in R^l_i, i = 1, 2$ are the measured outputs, defined by:

$$y_1(k) = C_1 x(k) + v_1(k)$$

$$y_2(k) = C_2 x(k) + v_2(k)$$

$w(k) \in R^r$ and $v_i(k) \in R^r_i, i = 1, 2$ are zero mean stationary mutually uncorrelated white Gaussian noise with intensities $W > 0$ and $V_i > 0$ $i = 1, 2$. Using the results established in paper [19] we develop here a discrete time counterpart as

$$\hat{x}(k) = \hat{z_1}(k) + \hat{z_2}(k) \tag{2.19}$$

where$\hat{z_1}(k)$ and $\hat{z_2}(k)$ are obtained from non optimal local filter.

$$\hat{z_i}(k+1) = (A - P(C_1^\intercal V_1^{-1} C_1 + C_2^\intercal V_2^{-1})C_2))\hat{z_i}(k) + PC_i^\intercal V_i^{-1} y_i(k) \quad i = 1, 2 \tag{2.20}$$

$$\hat{z_1}(k+1) = (A - PC_1^\intercal V_1^{-1} C_1 - PC_2^\intercal V_2^{-1} C_2)\hat{z_1}(k) + PC_1^\intercal V_1^{-1} y_1(k) \tag{2.21}$$

$$\hat{z_2}(k+1) = (A - PC_1^\intercal V_1^{-1} C_1 - PC_2^\intercal V_2^{-1} C_2)\hat{z_2}(k) + PC_2^\intercal V_2^{-1} y_2(k) \tag{2.22}$$

$$\hat{z_1}(k+1) = (A - K_1 C_1 - K_2 C_2)\hat{z_1}(k) + K_1 y_1(k) \tag{2.23}$$

$$\hat{z_2}(k+1) = (A - K_1C_1 - K_2C_2)\hat{z_2}(k) + K_2y_2(k) \tag{2.24}$$

The equations (2.23) and (2.24) are equations of non-optimal local filter. The optimal global estimates are obtained by simple addition of non-optimal local estimates

$$\hat{x}(k) = \hat{z_1}(k) + \hat{z_2}(k) \tag{2.25}$$

$$\hat{x}(k+1) = (A - K_1C_1 - K_2C_2)\hat{x}(k) + K_1y_1(k) + K_1y_2(k) \tag{2.26}$$

$$\hat{x}(k+1) = \left(A - \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}\right)\hat{x}(k) + \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \tag{2.27}$$

$$\hat{x}(k+1) = (A - KC)\hat{x}(k) + K \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \tag{2.28}$$

$$where \quad K = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \quad and \quad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{2.29}$$

Thus globally optimal centralized Kalman filter(quasi decentralized) estimates are obtained and implemented in the thesis. The simulation and results are demonstrated in Chapter 4. This can be generalized to n sensors(n sources of data).

2.5.3 Comparison of Centralized and Quasi-Decentralized Sensor Fusion Techniques

- **Group Sensor Method (Centralized Sensor Data Fusion)**

  Pros :

  - **Simple Implementation** : The implementation is simple and similar to that of a single Kalman Filter.
  - **Single Central Filter** : Only one central Kalman filter is required for central processing of sensor data.

  Cons :

  - **Large Bandwidth Utilization** : Since large amount of sensor data is sent to the central processor, large bandwidth is utilized.
  - **Less Scalable** : The large bandwidth becomes a bottleneck for this architecture and hence the system becomes less scalable.
  - **Fault Intolerant** : The breakdown of the central processor can be catastrophic. This system is not fault tolerant.

- **Quasi- Decentralized Sensor Data Fusion**

  Pros :

  - **Less Bandwidth** : The load on the central processor is reduced since data of minimum dimension is sent to the central processor thus reducing the bandwidth utilization.
  - **Scalable** : This architecture is easily scalable to increase in number of sensors.
  - **Fault tolerant** : The objective of this design is that if any sensor node fails, the degradation of the system is gradual and non - catastrophic.
  - **Easy to detect fault** : Given that each node influences the fusion performance in a similar way, if any nodes fails the failures are easily detected.

  Cons :

  - **Large number of local filters** : Every sensor node has a local filter. Therefore increase in number of sensors requires large number of local filters.

2.6 Use of Standby Measurements

In critical applications most of the times standby sensor systems are available. These systems provide measurements when actual measurement systems fail. But these standby measurement systems are operating most of the times but remain idle. In paper [17] it is proved that the steady state estimation error of the linear optimal Kalman Filter can be considerably reduced if the standby measurements are used in addition to actual system measurements. Thus, instead of letting the standby measurements to be idle they may be used in the estimation as well to improve the estimation accuracy and minimize the error covariance. The paper [17] is based upon a continuous time system. In this thesis we implement and prove the results for a discrete time system.

Consider a linear dynamic discrete time system with state space variables $x(k) \in R^n$ and corresponding system measurements $y(k) \in R^r$ given as

$$x(k+1) = Ax(k) + Gw(k), \quad E\{w(k)w^T(k)\} = W\delta(k-m), \quad E\{x(k_0)\} = x_0, var\{x(k_0)\} = P_0$$

$$y(k) = Cx(k) + v(k), \quad E\{v(k)v^T(k)\} = V\delta(k-m)$$

where $w(k) \in R^m$ and $v(k) \in R^r$ are zero mean stationary Gaussian white noise processes. The optimal estimate of *x(k)* can be obtained by using the Kalman Filter as

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + K(y(k) - \hat{y}(k)) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{2.30}$$

where matrix K is chosen such that the estimation error $e(k) = x(k) - \hat{x}(k)$ is minimised in the mean square sense. We assume for simplicity that $w(k)$, $v(k)$ and $x(k_0)$ are mutually uncorrelated. The optimal Kalman filter gain at steady state is given by $K = PC^{\intercal}V^{-1}$ where P is a positive semi definite stabilizing solution of the algebraic Ricatti equation.

$$P = APA^{\intercal} + GWG^{\intercal} - (APC)(V + C^{\intercal}PC)^{-1}C^{\intercal}PA^{\intercal} \tag{2.31}$$

The equation 2.31 is the discrete time filter algebraic Ricatti equation. The minimum value of the estimation error variance is equal to $\sigma_{min} = \sigma^{opt} = tr(P)$. The optimal solution to the steady state estimation problem is obtained in an r - dimensional measurement space namely $y(k) \in R^r$. If additional standby measurements are also available in the operational mode, that is,

$$y_s(k) = C_s x(k) + v_s(k) \quad E\{v_s(k)v_s^{\intercal}(k)\} = V_s\delta(k-m)$$

The overall system measurements are now represented as

$$Y_s(k) = \begin{bmatrix} y(k) \\ y_s(k) \end{bmatrix} = \begin{bmatrix} C \\ C_s \end{bmatrix} x(k) + \begin{bmatrix} v(k) \\ v_s(k) \end{bmatrix} \tag{2.32}$$

In (2.32) $v(k)$ and $v_s(k)$ are uncorrelated since they come from separate channels. The optimal solution is now given as

$$K_s^{opt} = \tilde{P}\begin{bmatrix} C \\ C_s \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} V^{-1} & 0 \\ 0 & V_s^{-1} \end{bmatrix} \tag{2.33}$$

$$= \tilde{P}\big(C^{\mathsf{T}}V^{-1} + C_s^{\mathsf{T}}V_s^{-1}\big) \tag{2.34}$$

and satisfies the Ricatti equation

$$V + CPC^{\mathsf{T}} \geq \tilde{V} + C\tilde{P}C^{\mathsf{T}} \geq 0 \tag{2.35}$$

Substituting in discrete Ricatti equation

$$\tilde{P} = A\tilde{P}A^{\mathsf{T}} + GWG^{\mathsf{T}} - A\tilde{P}\tilde{C}\big(\tilde{V} + \tilde{C}\tilde{P}\tilde{C}^{\mathsf{T}})^{-1}\tilde{C}^{\mathsf{T}}\tilde{P}A^{\mathsf{T}} \tag{2.36}$$

The corresponding minimum value of estimation error is $\sigma^{\tilde{opt}} = tr(\tilde{P})$. Expanding equation (2.36) we get,

$$\tilde{P} = A\tilde{P}A^{\mathsf{T}} + GWG^{\mathsf{T}} - A\tilde{P}\begin{bmatrix} C \\ C_s \end{bmatrix}\left(\begin{bmatrix} V & 0 \\ 0 & V_s \end{bmatrix} + \begin{bmatrix} C \\ C_s \end{bmatrix}\tilde{P}\begin{bmatrix} C \\ C_s \end{bmatrix}^{\mathsf{T}}\right)^{-1}\begin{bmatrix} C \\ C_s \end{bmatrix}^{\mathsf{T}}\tilde{P}A^{\mathsf{T}} \tag{2.37}$$

$$\tilde{P} = A\tilde{P}A^{\mathsf{T}} + GWG^{\mathsf{T}} - AP\begin{bmatrix} C \\ C_s \end{bmatrix}\left(\begin{bmatrix} V & 0 \\ 0 & V_s \end{bmatrix} + \left[\begin{array}{c:c} C\tilde{P}C^{\mathsf{T}} & C\tilde{P}C_s^{\mathsf{T}} \\ \hdashline C_s\tilde{P}C^{\mathsf{T}} & C_s\tilde{P}C_s^{\mathsf{T}} \end{array}\right]\right)^{-1}\begin{bmatrix} C \\ C_s \end{bmatrix}^{\mathsf{T}}\tilde{P}A^{\mathsf{T}} \tag{2.38}$$

Based on research in Paper [18] if 2.35 holds then

$$\begin{aligned} &\therefore P \geq \tilde{P} \geq 0 \\ &\tilde{\sigma}^{opt} \leq \sigma^{opt} \end{aligned} \tag{2.39}$$

Thus standby measurements reduces the estimation error in Kalman filter. The above observations can be generalized to several standby measurements. The simulation results are demonstrated in Section 4.2

2.7 Trajectory Prediction

A spacial temporal representation of the displacement of the vehicle is known as a trajectory. It defines the dynamic representation of its path. The aim of the trajectory prediction is to predict at

t_0, the future displacement of the vehicle during the time slice $[t_0, t_0 + h]$ where h is the prediction period.

The quality of prediction depends upon the errors measured between the estimated value and the real value. There are various approaches for trajectory prediction.

- Geometric Approach

  In the geometric approach it is believed that since roads are designed with specific geometric models, the trajectory of the vehicle can be modelled with geometric features. Some related works like Paper [12] use polynomials representation of the trajectory like first, third or fifth order polynomial representation. Some other research works like Paper [13] make use of clothiods to model the trajectory. This approach does not make use of any model of the vehicle. This is the drawback for this type of approach.

- Dynamic Approach

  This approach has a major advantage that it tends to reproduce the dynamic behavior of the vehicle. The trajectory prediction is based on this model. The Kalman filter is used for future trajectory prediction.

In this thesis, the Kalman recursive filter is used for following purposes:

1. Sensor fusion for combining readings from a radar and a camera.
2. Trajectory prediction of vehicles.

2.7.1 The Kalman Predictor

The Kalman filter is a recursive estimator. There are two phases of the Kalman filter that alternate, the prediction state advances the state until the next observation and the update state, incorporates observations. The Kalman prediction involves skipping the updates and predicting multiple steps.

2.7.2 Multi-Step Kalman Predictor

Consider a discrete time, linear time invariant (LTI) system without any exogenous input, described by state space model given in equation 2.1, which for the reason of convenience is repeated here

$$x(k+1) = Ax(k) + w(k)$$

$$y(k) = Cx(k) + v(k)$$

Prediction involves estimating the state *x(n+r), r > 1* and n measurements are given.

The optimal estimate for the state *x(n+r)* based on n measurements is obtained as follows:

$$\begin{aligned}
\hat{x}(N+r|N) &= E[x(N+r)|y^N] \\
&= E[Ax(N+r-1) + v_1(N+r-1)|y^N] \\
&= E[Ax(N+r-1)|y^N] + E[v_1(N+r-1)|y^N] \\
&= AE[x(N+r-1)|y^N] + E[v_1(N+r-1)] \\
&= A\hat{x}(N+r-1|N)
\end{aligned} \tag{2.40}$$

Since, $v_1(N+r-1)$ is independent of y^N $\forall$r >1

$$E[v_1(N+r-1)|y^N] = E[v_1(N+r-1)] = 0 \tag{2.41}$$

By iterating the above equations, we get

$$\boxed{\hat{x}(N+r|N) = A^{r-1}\hat{x}(N+1)|N)} \tag{2.42}$$

The output $y(N+r)$ based on data y^N is given as :

$$\begin{aligned}
\hat{y}(N+r|N) &= E[y(N+r)|y^N] \\
&= E[Cx(N+r) + v_2(N+r)|y^N] \\
&= E[Cx(N+r)|y^N] + E[v_2(N+r)|y^N] \\
&= C\hat{x}(N+r|N) \\
&= CA^{r-1}\hat{x}(N+1)|N)
\end{aligned} \tag{2.43}$$

Since, $v_2(N+r)$ is independent of $y^N \forall r > 1$

$$E[v_2(N+r)|y^N] = E[v_2(N+r)] = 0 \tag{2.44}$$

By iterating the above equations, we get

$$\boxed{\hat{y}(N+r|N) = CA^{r-1}\hat{x}(N+1)|N)} \tag{2.45}$$

2.8 Propagation of Uncertainty

We obtain $X_N = [X1, X2, .., Xn]$ as predicted poses from the Kalman filter with the sampling time denoted by T. The error covariance matrix of X_{k+1} is given by

$$P_{k+1} = AP_kA^\intercal + Q \tag{2.46}$$

The initial state X_0 and P_0 are known from the Kalman filter tracking. In the first prediction P_0 is modelled as

$$P_0 = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}^2} & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}^2} \end{bmatrix} \tag{2.47}$$

To account for the uncertainty, the position of the target vehicle is modelled as a Gaussian variable with normal distribution.

2.8.1 Normal Distribution

In probability theory, Gaussian / normal distribution is commonly used. Physical quantities which are affected by measurements errors are assumed to have distributions that are normal. This is used for propagation of uncertainty in trajectory prediction. In our case, the normal distribution is extended to higher dimensions to get a multivariate normal distribution.

2.8.2 Bi-variate Normal Distribution

The probability density function of *[X, Y]* is

$$p(x,y) = \frac{1}{2\pi.\sigma_x.\sigma_y.\sqrt{1-\sigma^2}}.exp\left\{\frac{-1}{2(1-\sigma^2)}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2 - \frac{2(x-\mu_x)(x-\mu_y)}{\sigma_x.\sigma_y}\right]\right\} \tag{2.48}$$

σ is the correlation between x and y

σ_x , σ_y are the standard deviation of x and y

μ_x and μ_y are expected values of x and y position

This value is available at each prediction step in the Kalman filter. Since, correlation between x andy σ is 0

$$p(x,y) = \frac{1}{2\pi.\sigma_x.\sigma_y}.exp\left\{\frac{-1}{2}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]\right\} \tag{2.49}$$

The 95 % confidence in Gaussian bi-variate distribution is represented as an ellipse. The ellipse defines the region that contains 95% of all samples that can be derived from the Gaussian distribution. The angle of the ellipse is determined by the covariance of the data. The magnitude of the ellipse axes depends on the variance of the variables. The general equation of the ellipse with major axis 2a and minor axis 2b, is given as

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \tag{2.50}$$

In our case, for 95 % confidence length of the axes is defined in terms of standard deviation σ_x and σ_y , the error ellipse equation would be as follows:

$$\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2 = s \tag{2.51}$$

s is chosen depending upon the confidence level. For 95 % confidence level s is chosen as $s = 5.991$ using the Chi - Square distribution table shown in Figure 2.3

df	0.995	0.99	0.975	0.95	0.90	0.10	0.05	0.025	0.01	0.005
1	---	---	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188

Figure 2.3: Chi Square Distribution Table

2.8.2.1 Chi Square Distribution

The Chi square distribution presented in Figure 2.3 is defined in terms of the degree of freedom. For our case , we consider two degrees of freedom. The probability is available from the probability table for two degree of freedom. s from the Chi-Square distribution table is *5.991*. The standard deviation is derived from the matrix P. The increasing size of the standard deviation accounts for the increase in ellipse size.

2.9 Collision Detection

The probability of collision between a subject vehicle(s) and a target vehicle(t) having uncertain configurations is defined as:

$$P_{coll} = \iint P_s(x,y).P_t(x,y)dx.dy \tag{2.52}$$

The integral can be analytically computed as follows:

$$P_d(x;m,\Sigma) \approx \frac{exp\Big(\frac{-1}{2}(x-m)'\Sigma(x-m)\Big)}{\sqrt{(2\pi)^d.det(\Sigma)}} \tag{2.53}$$

with $m \in \mathbb{R}^d$, and Σ a symmetric, positive semi-definite square matrix of dimension d.

Let $P_s \sim P_d(m_1,\Sigma_1)$ and $P_t \sim P_d(m_2,\Sigma_2)$, where $P_d(m,\Sigma)$ denotes the d - dimensional Gaussian distribution with mean m and covariance Σ , with density $P_d(x;m,\Sigma)$.

$$\int P_s(x; m_1, \Sigma_1) P_t(x; m_2, \Sigma_2) dx = \int \frac{exp\Big(-\frac{1}{2}\Big[\alpha_1 + \alpha_2\Big]\Big)}{\big(2\pi\big)^2 \sqrt{det(\Sigma_1)}\sqrt{det(\Sigma_2)}} dx \tag{2.54}$$

with,

$$\begin{aligned} \alpha_1 &= (x - m_1)'\Sigma_1(x - m_1) \\ \alpha_2 &= (x - m_2)'\Sigma_2(x - m_2) \end{aligned} \tag{2.55}$$

Let

$$A = (x - m_1)'\Sigma_1(x - m_1) + (x - m_2)'\Sigma_2(x - m_2)$$

Solving we get,

$$A - B = m_1'{\Sigma_1}^{-1}m_1 + m_2'{\Sigma_2}^{-1}m_2 - ({\Sigma_1}^{-1}m_1 + {\Sigma_2}^{-1}m_2)'\big({\Sigma_1}^{-1} + {\Sigma_2}^{-1}\big)^{-1}({\Sigma_1}^{-1}m_1 + {\Sigma_2}^{-1}m_2) \tag{2.56}$$

$$B = [x - m]'\Sigma^{-1}[x - m]$$

Therefore,

$$\int \frac{exp\Big(-\frac{1}{2}A\Big)}{\big(2\pi\big)^2 \sqrt{det(\Sigma_1)}\sqrt{det(\Sigma_2)}} dx = \frac{exp\Big[-\frac{1}{2}\Big(A - B\Big)\Big]}{\big(2\pi\big)^2 \sqrt{det(\Sigma_1)}\sqrt{det(\Sigma_2)}} \times \int exp\Big(-\frac{1}{2}B\Big) dx \tag{2.57}$$

with,

$$\int exp\Big(-\frac{1}{2}B\Big) dx = \sqrt{(2\pi)^2}\sqrt{det(\Sigma} \tag{2.58}$$

$$\Sigma^{-1} = (\Sigma_1^{-1} + \Sigma_2^{-1})$$

Solving equation 2.57 the probability of collision is computed. This probability takes into account the uncertainties during future position prediction for both subject and target vehicles. The probability of collision computed can be used further for decision making by other ADAS systems. A decision could be made if the current maneuver is safe or not. This approach thus provides a larger perspective of the scene and solves our aim of collision risk assessment.

Chapter 3

Implementation

3.1 PreScan Simulation Software

PreScan software from TASS international is used in order to create simulation environment . This software is specially designed to provide a development environment of intelligent vehicles. It has been used for developing and debugging of control software. Software in loop simulation feature of PreScan is exploited in this thesis. The software has an integrated Simulink interface which is used to create control software. This chapter discusses the basics of PreScan software. The software consists of three parts : GUI, Simulink compilation sheet and the VisViewer. The PreScan Help Manual [16] provides a detailed description about the software.

3.1.1 Graphical User Interface (GUI)

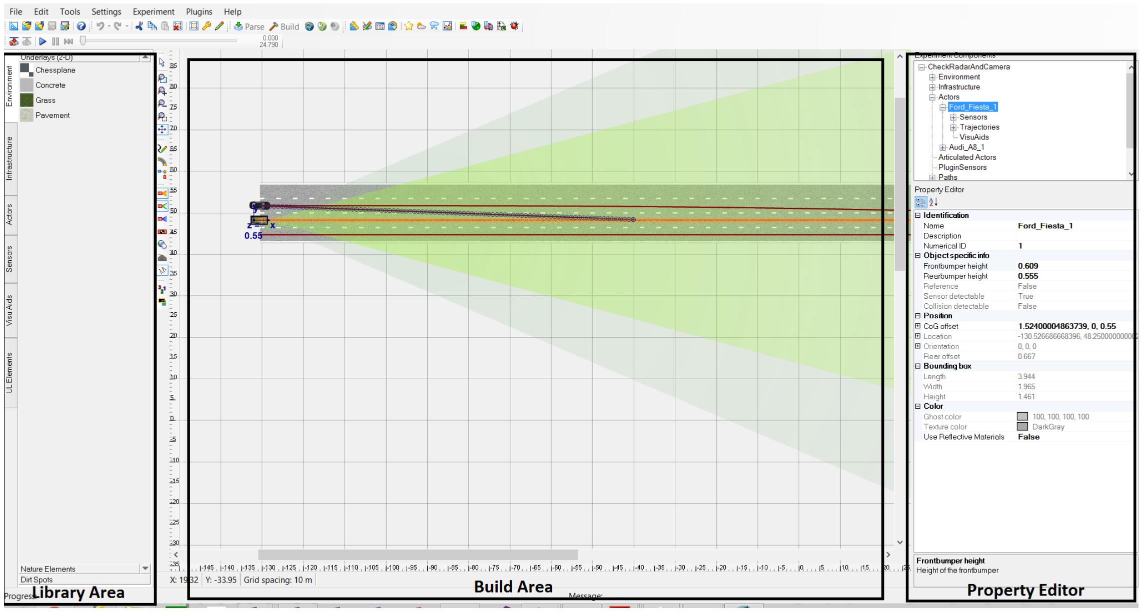
The Figure 3.1 shows the PreScan GUI. Here all the infrastructure is constructed, the sensors and actors are added. The scenarios to be tested are created. GUI is further divided into three sections Library Elements, the Build Area and the Property editor.

3.1.2 Infrastructure

The road environment is known as the infrastructure. Pieces of road sections are available like straight road, bent road, roundabouts, intersections etc.

3.1.3 Actors

Different cars, trucks, motorcycle are available to be used in simulation. An actor has to be configured before it is used. The configurations available are vehicle dynamics, driver model, trajectories and



Credit: PreScan Simulation Software

Figure 3.1: Graphical User Interface

others.

3.1.3.1 Vehicle Dynamics

Four options for vehicle dynamics are available for choice as shown in Figure 3.2:

1. None : No dynamics

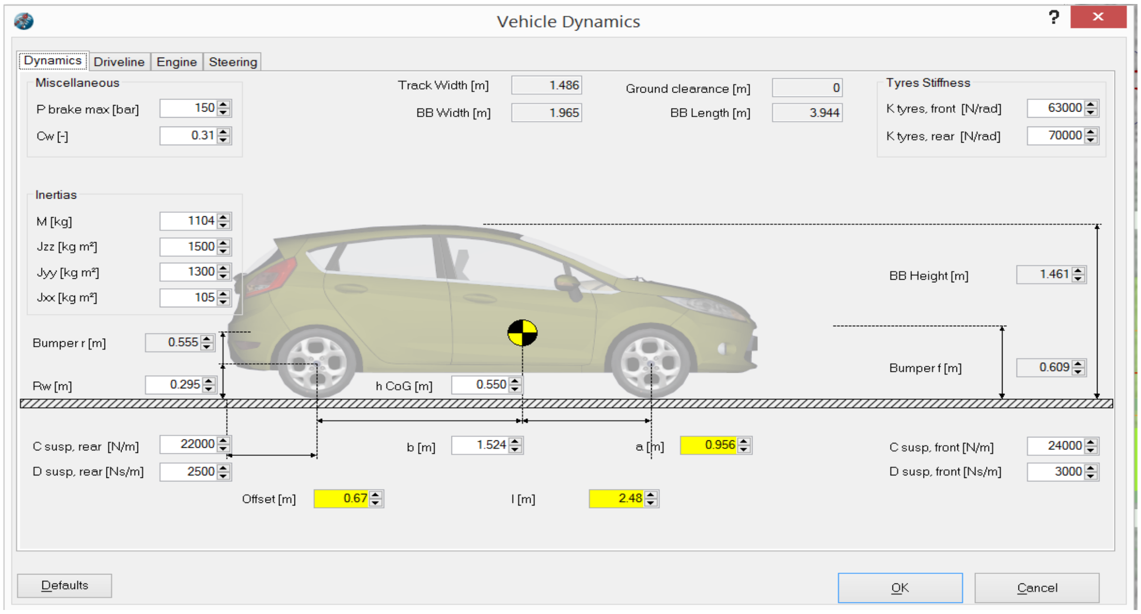
2. 2D Simple : A simple bicycle model with roll dynamics is available. In the customize tab the user can modify parameters for dynamics, driveline and engine. For a full list of descriptions for all parameters please see the PreScan manual.

3. 3D Simple : This model is similar to 2D model with added suspension.

4. User Specified : User can create his own Simulink model.

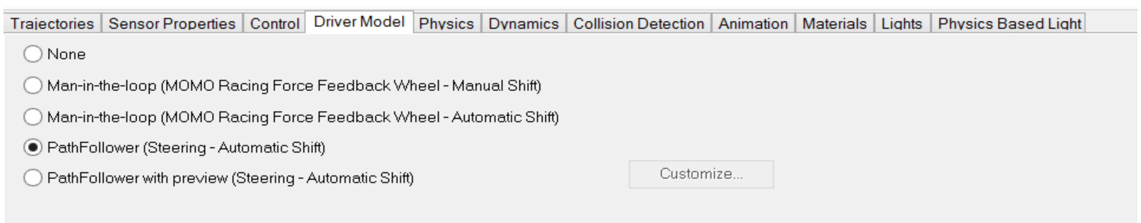
3.1.3.2 Driver Model

The driver Model decides how the vehicle is going to be controlled. Different setting available for the user as shown in Figure 3.3 are



Credit: PreScan Simulation Software

Figure 3.2: VehicleDynamics



Credit: PreScan Simulation Software

Figure 3.3: Driver Model

1. Man In Loop - This is used when hardware has to be interfaced.

2. Software in Loop (Path Follower) - Here PreScan calculates the steer angle required to minimize the lateral error been the vehicle and the user trajectory.

3.1.3.3 Trajectories

Trajectory is defined as a path and speed profile to be followed. The path can be defined using inherited path or manually. The Figure 3.4 shows how an inherited path can be created. Once a path is created the speed profile has to be created for the specific actor.

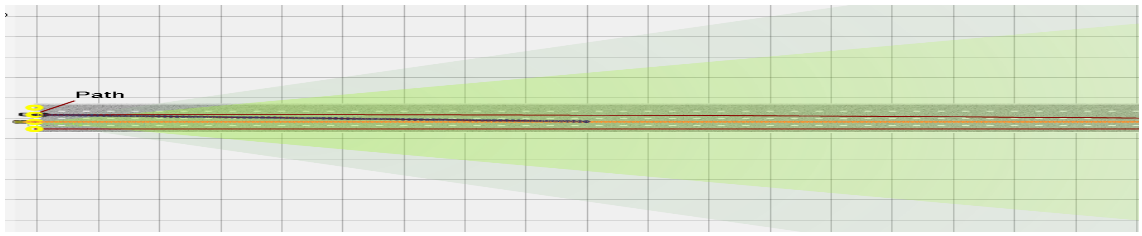


Figure 3.4: Trajectory

Signal Name	Description
Object ID	Numerical ID of the detected object
RangeX[m]	X component of the Range, in sensor coordinates
RangeY [m]	Y component of the Range, in sensor coordinates
RangeZ [m]	Z component of the Range, in sensor coordinates
DopplerVelocityX/Y/Z [m/s]	Velocity of target point, relative to the sensor, along the line of-sight between sensor and target point, decomposed into X, Y, Z of the sensor's coordinate system.
Theta [deg]	Azimuth angle in the sensor's coordinate system at which the target is detected.

Table 3.1: Camera Sensor Outputs

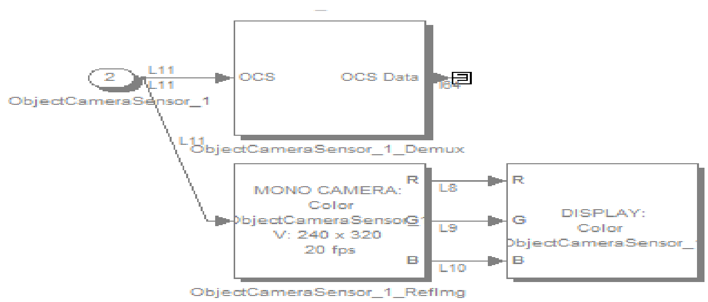
3.1.3.4 Sensors

Sensors can be added to a vehicle. Three categories of sensors are available: Idealized, detailed and ground truth. We used camera and radar sensor in our simulation. The outputs of camera and radar are listed in Table 3.1 and 3.2

Refer Figure 3.5 and 3.6 for simulink models of camera and radar sensors:

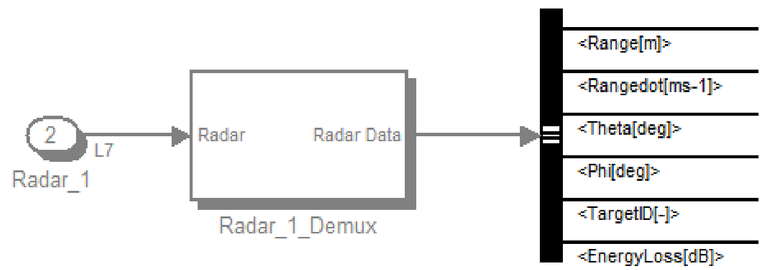
Signal Name	Description
Object ID	Numerical ID of the detected object
Range[m]	Range of the object in sensor coordinates
DopplerVelocityX/Y/Z [m/s]	Velocity of target point, relative to the sensor, along the line of-sight between sensor and target point, decomposed into X, Y, Z of the sensor's coordinate system.
Theta [deg]	Azimuth angle in the sensor's coordinate system at which the target is detected.

Table 3.2: Radar Sensor Outputs



Credit: PreScan Simulation Software

Figure 3.5: Camera Simulink Model



Credit: PreScan Simulation Software

Figure 3.6: Radar Simulink Model

3.1.4 Simulink Compilation Sheet

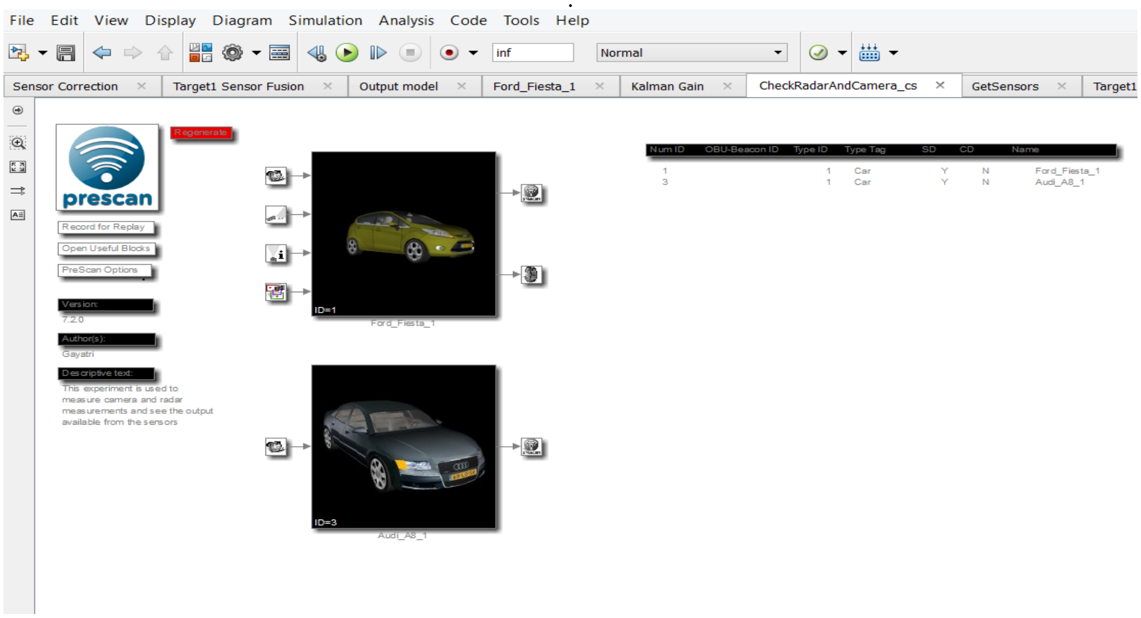
Once the experiment is built, a Simulink compilation sheet is created. All the coding of control algorithms goes here. A subject vehicle with other target vehicles and sensors is shown in Figure 3.7

3.1.5 VisViewer

When an experiment is built in GUI and all the coding is done using the compilation sheet , the simulation is viewed in the VisViewer.

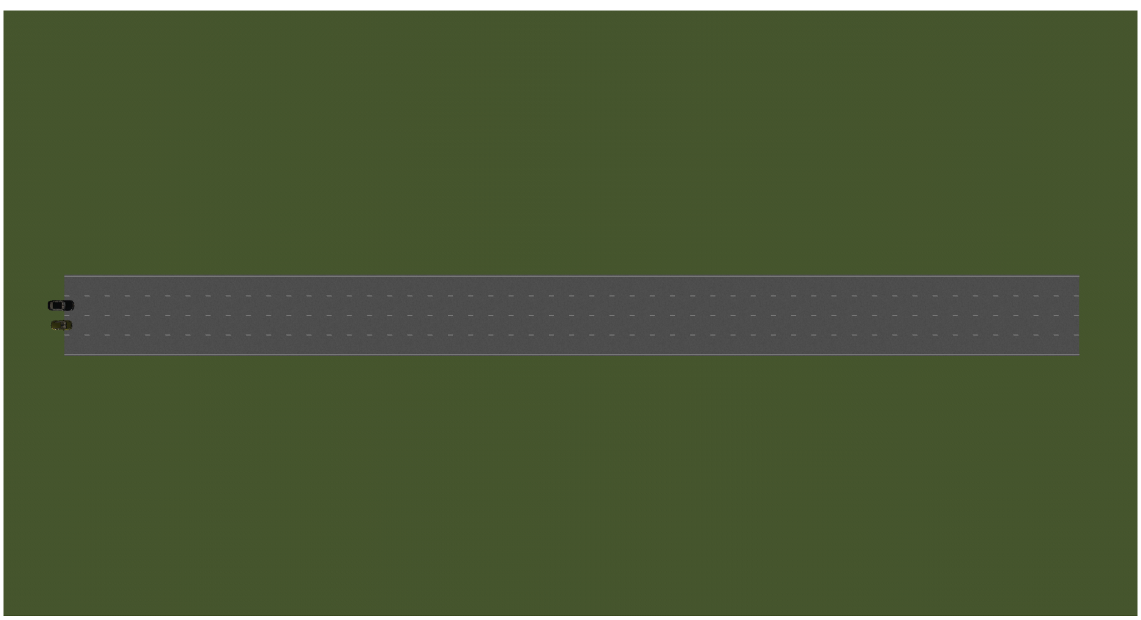
PreScans VisViewer also comes with the option to record simulations. This is set up through the Video capture option in the toolbar. Here the user can choose which view to capture as well as the quality and format of the video.

The Figure 3.8 shows how simulation is seen in the VisViewer.



Credit: PreScan Simulation Software

Figure 3.7: Compilation Sheet



Credit: PreScan Simulation Software

Figure 3.8: VisViewer

3.2 Implementation of the Concept

3.2.1 System Overview

The basic blocks of the autonomous vehicle guidance function developed is shown in Figure 3.9. The Collision Probability Detection module holds the implementation of the algorithm proposed in the thesis. This module receives measurement data from camera and radar Simulink blocks and vehicle data from vehicle sensors. The outputs of the Collision Probability Detection namely uncertainty Ellipses and Collision Probability could be used by other subsystems in the automated vehicle for example lateral/longitudinal control, motion planning, etc.

The remaining blocks including Path Follower, Vehicle Dynamics, Visualization are generated by PreScan based on the simulated environment data received from the software.

3.2.2 Collision Probability Detection Subsystem

This forms the core algorithm module for robust prediction. It consists of the sensor correction module, current environment state, Trajectory Prediction Module and Collision Detection Module which will be explained in depth in subsequent subsections. The Simulink block diagram of this subsystem is shown in Figure 3.10. The inputs and outputs of this block are described in Table 3.3

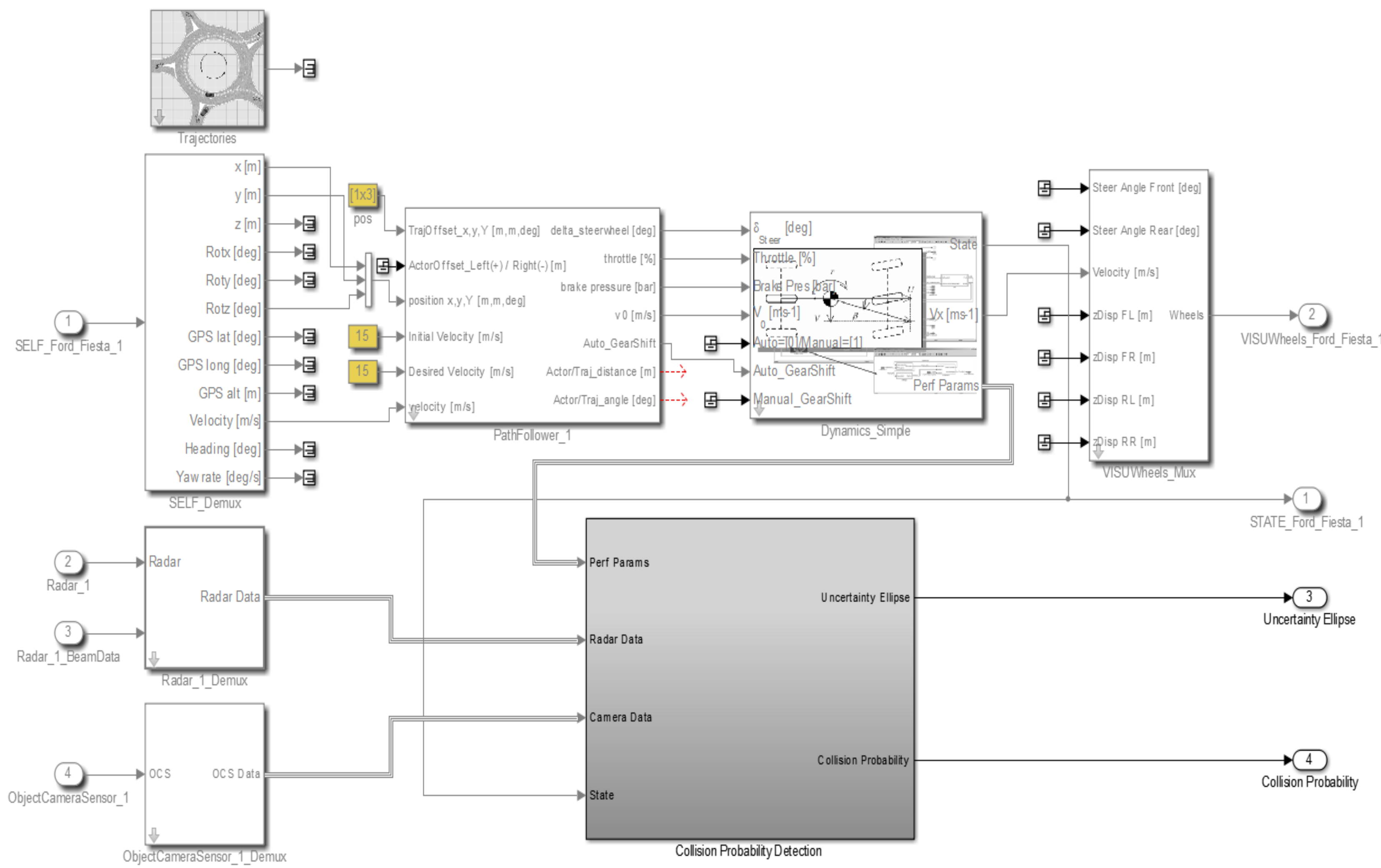


Figure 3.9: Collision Detection System

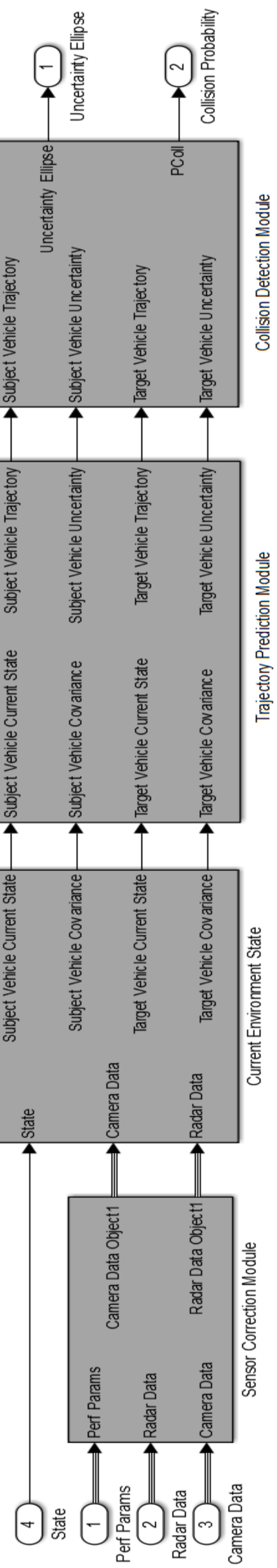


Figure 3.10: Collision Probability Detection Module

I/O	Signal Name	Signal Description
Inputs	State	Measurement Data From Vehicle Sensors
	Perf Param	Velocity Data of subject Vehicle
	Radar Data	Measurement data from radar sensor(Range, Velocity X,Velocity Y, Azimuth Angle)
	Camera Data	Measurement data from camera sensor(RangeX, RangeY, Velocity X,Velocity Y, Azimuth Angle)
Outputs	Uncertainty Ellipse	Ellipses accounting the uncertainty in trajectory prediction
	Collision Probability	Probability of Collsion $0 \leq P \leq 1$

Table 3.3: Collision Probability Detection Subsystem

3.2.2.1 Sensor Correction Module

This module is responsible for signal conditioning. It receives the raw data from sensors and converts into appropriate data useful for further calculations. This includes finding the absolute velocities of targets, object validation, longitudinal and lateral range calculation in case of radar sensor, etc. The input and outputs of this block are described in Table 3.4.

3.2.2.2 Current Environment State

This subsystem block is responsible for the estimation of current environment. This includes the current state of the target vehicle and subject vehicle. It is further divided into two subsections vehicle current state and target vehicle sensor fusion modules. They are explained in further details in subsequent text. The simulink block of this module is shown in Figure 3.11. The inputs and outputs of this block are shown in Table 3.5.

The Kalman Filter forms the heart of this subsystem. The Kalman filter is used two functions as a state estimator and for sensor fusion.

Vehicle Current State

The Kalman Filter receives measurement data from vehicle sensors like longitudinal and lateral velocity and acceleration and estimates the states of the subject vehicle. The simulink model of the Kalman filter is shown in Figure 3.12.

I/O	Signal Name	Signal Description
Inputs	Perf Params	Velocity Data of subject Vehicle
	Radar Data	Measurement data from radar sensor(Range, Velocity X,Velocity Y, Azimuth Angle)
	Camera Data	Measurement data from camera sensor(RangeX, RangeY, Velocity X,Velocity Y, Azimuth Angle)
Outputs	Camera Data Object1	Conditioned Camera Data
	Radar Data Object1	Conditioned Radar Data

Table 3.4: Sensor Correction Module

I/O	Signal Name	Signal Description
Inputs	Camera Data	Conditioned Camera Data
	Radar Data	Conditioned Radar Data
Outputs	Subject Vehicle Current State	Current State Estimate of Subject Vehicle
	Subject Vehicle Covariance	Current Covariance Matrix of Subject Vehicle
	Target Vehicle Current State	Current State Estimate of Target Vehicle
	Target Vehicle Covariance	Current Covariance Matrix of Target Vehicle

Table 3.5: Current Environment State Module

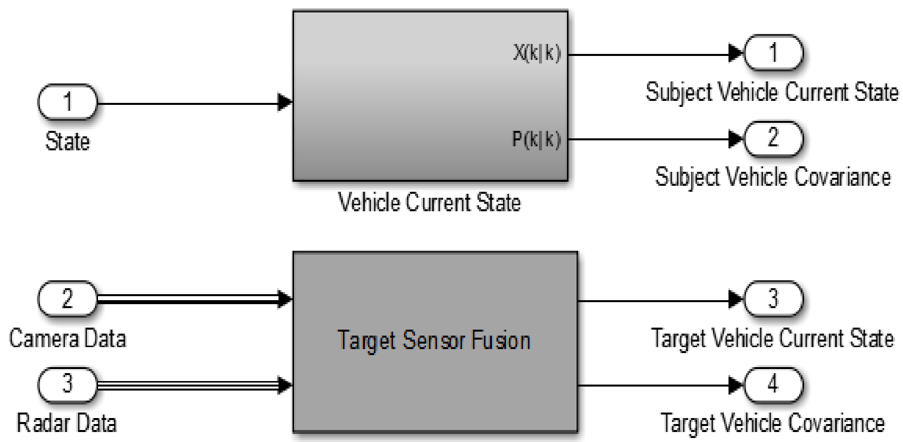


Figure 3.11: Current Environment State

Target Sensor Fusion

A practical evaluation of both centralized and quasi-decentralized sensor data fusion is conducted. The Group Sensor Fusion as discussed in previous sections is developed as shown in Figure 3.13. The data from camera and radar are fused together to accurately estimate the state of the target vehicles. As proposed earlier the standby measurements is used to improve the performance of estimation by reducing the covariance error. Also the quasi decentralized sensor data fusion is implemented. Here, local non optimal Kalman filters process the camera and radar data at each node and the estimates from both the filters are combined to get an optimal estimate. The implementation of the quasi-decentralized process is shown in Figure 3.14.

3.2.2.3 Trajectory Prediction Module

The trajectory prediction module performs the trajectory prediction of the subject and target vehicle for a future time interval, for example 4 sec ahead of time. The Kalman Predictor is used for trajectory prediction which uses the current state of the environment to create future trajectories of the target and subject vehicle. This module also calculates the covariance error of the estimate at each time step of prediction. It is to be noted that since measurements are not available for updating

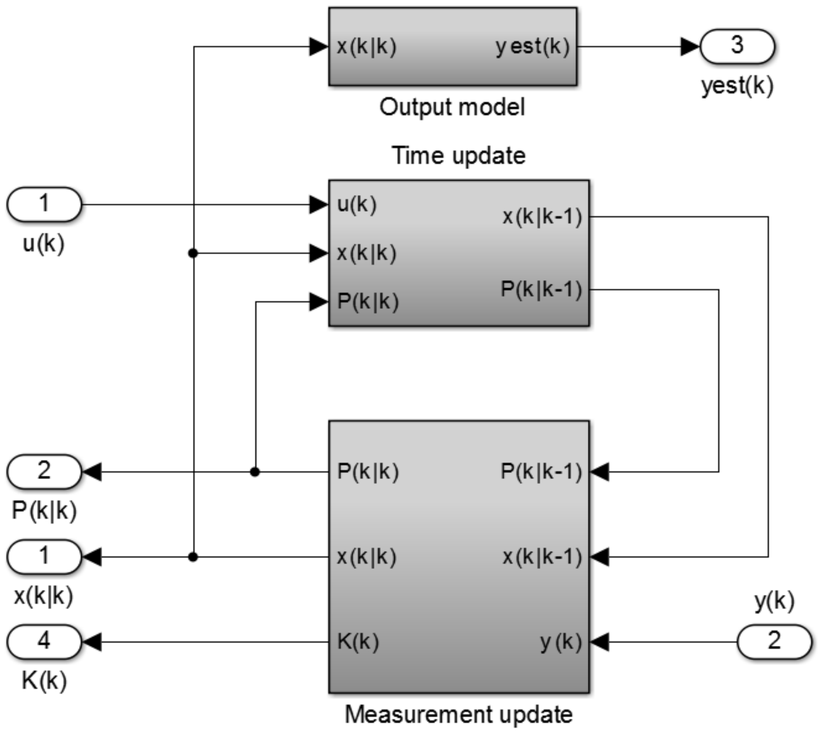


Figure 3.12: Kalman Filter

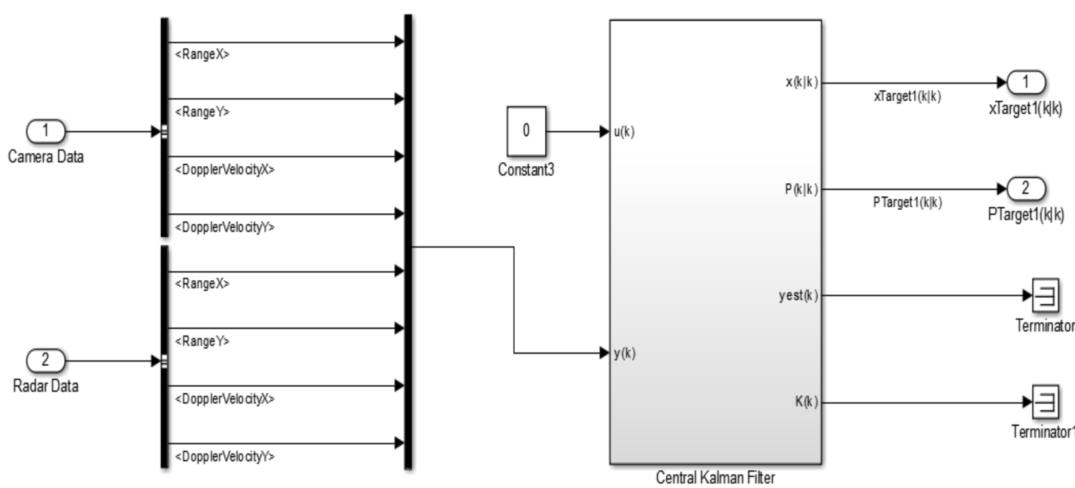


Figure 3.13: Group Sensor Data Fusion

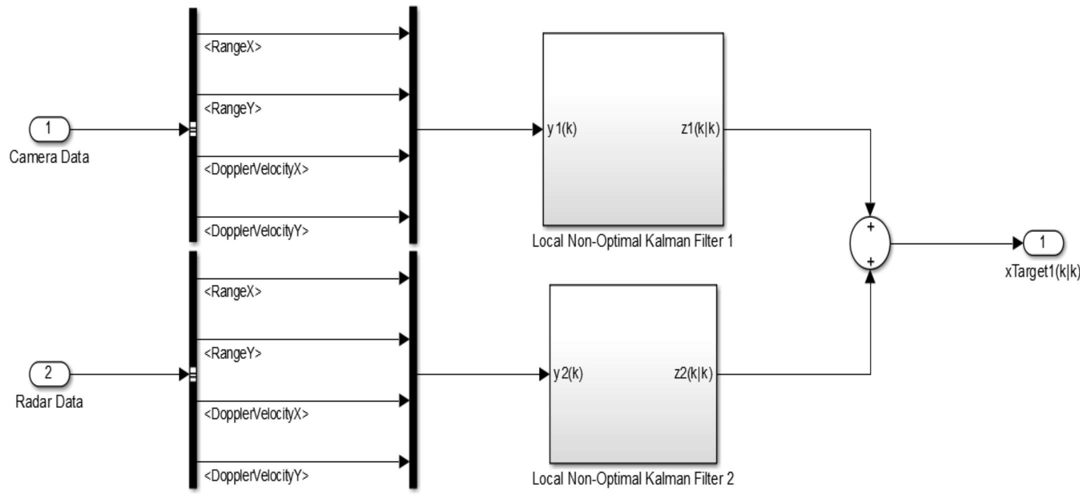


Figure 3.14: Quasi-Decentralized Data Fusion

that the covariance error keeps increasing at each future time step. The inputs and outputs of the trajectory prediction module are described in Table 3.6.

3.2.2.4 Collision Detection Module

This is the decision making subsystem. It uses the predicted trajectory and the uncertainty ellipses of the subject and target vehicle to predict the probability of collision. The algorithm for the calculation is discussed in earlier sections. The inputs and outputs of the collision detection module are described in Table 3.7.

I/O	Signal Name	Signal Description
Inputs	Subject Vehicle Current State	Current State Estimate of Subject Vehicle
	Subject Vehicle Covariance	Current Covariance Matrix of Subject Vehicle
	Target Vehicle Current State	Current State Estimate of Target Vehicle
	Target Vehicle Covariance	Current Covariance Matrix of Target Vehicle
Outputs	Subject Vehicle Trajectory	Future Trajectory of Subject Vehicle
	Subject Vehicle Uncertainty	Future Uncertainty of Subject Vehicle
	Target Vehicle Trajectory	Future Trajectory of Target Vehicle
	Target Vehicle Uncertainty	Future Uncertainty of Target Vehicle

Table 3.6: Current Environment State Module

I/O	Signal Name	Signal Description
Inputs	Subject Vehicle Trajectory	Future Trajectory of Subject Vehicle
	Subject Vehicle Uncertainty	Future Uncertainty of Subject Vehicle
	Target Vehicle Trajectory	Future Trajectory of Target Vehicle
	Target Vehicle Uncertainty	Future Uncertainty of Target Vehicle
Outputs	Uncertainty Ellipses	Ellipses accounting the uncertainty in trajectory prediction
	Collision Probability	Probability of Collision $0 \leq P \leq 1$

Table 3.7: Current Environment State Module

Chapter 4

Simulation and Results

4.1 Probability of Collision

The approach to find the probability of collision using the method of trajectory prediction has been tested in simulation. Two different use cases were studied with the aim of evaluating the collision probability of the subject vehicle with a static and dynamic obstacle. The subject vehicle is equipped with a camera and a radar sensor. The simulation is carried out on a straight road of length of 200 m with a lane width of 3.5 m. Three figures are presented to demonstrate the simulation results. The first figure denoted by (a) represents a bird eye view of the scenario at the mentioned time instant. The other two graphs depict the predicted trajectory for 4 seconds and the probability of collision over the entire future time considered. Both scenarios occur on a straight road with 4 lanes, each 3.5 m wide. The PreScan simulator provides simulated outputs for the subject and target vehicle at the rate of 20 Hz. At each sample time, a 4 seconds trajectory prediction is made for each object. The camera, radar and the predicted trajectories logic samples at 20 Hz.

4.1.1 Scenario Evaluation

Two different scenarios are considered as discussed below:

4.1.1.1 Scenario 1: Collision with static obstacle

The subject vehicle is travelling at a speed of 25 mps. Figure 4.1(a) represents the position of the subject vehicle over time. Figure 4.1(b) represents the probability of collision and the future trajectory is depicted in Figure 4.1(c). The static object is located at a distance of 175 m. At 3 seconds, no collision is predicted and the probability of collision is zero. At time instant of 5 seconds, a very low probability of collision is observed. As the subject vehicle gets closer to the obstacle, the probability of collision increases as the trajectory of subject vehicle intersects the position of the obstacle. The probability increases to around 0.1 at time 7 seconds and finally at 8 second and 8.5 seconds, there is a high risk of collision with the static obstacle.

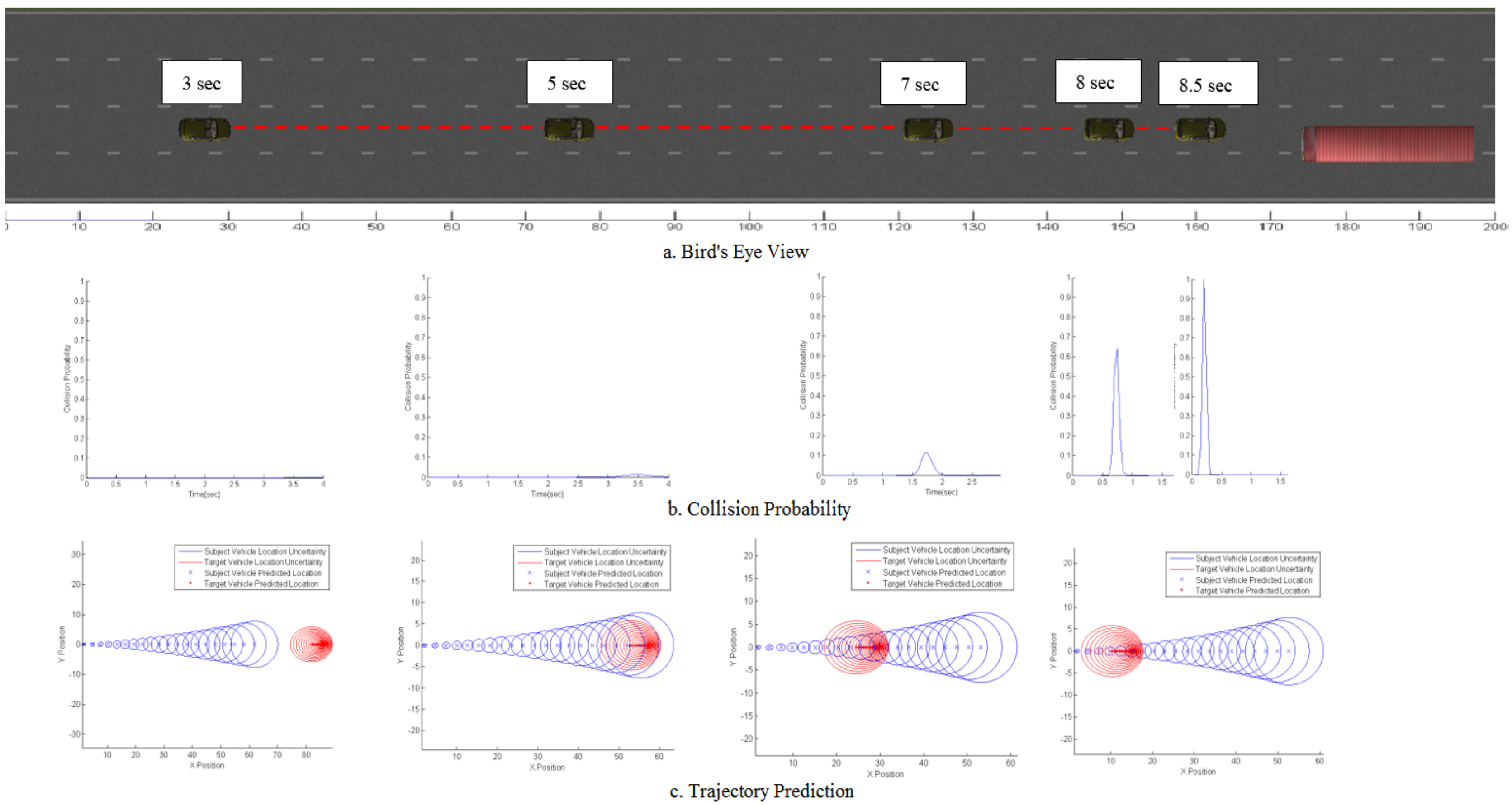


Figure 4.1: Collision with static object at different time intervals

4.1.1.2 Scenario 2: Collision with dynamic obstacles

In this scenario, the target vehicle is travelling on lane 2 with velocity of 7 mps; whereas subject vehicle is travelling with velocity of 12 mps. The observations at different times are explained below.

At time = 3 second

As it can be observed from Figure 4.2(a), the two vehicles are far away from each other. Their predicted trajectories do not overlap at any future time step as shown in Figure 4.2(c) and hence the probability of collision is zero as shown in Figure 4.2(b).

At time = 6 seconds

At this time, the target vehicle is making a lane change and enters into the lane as shown in Figure 4.3(a). There is a negligible probability of collision as show in Figure 4.3(b) and 4.3(c).

At time = 7 seconds

As seen in Figure 4.4(a), the target vehicle has entered the new lane. It can observed that the trajectories of the subject and target vehicle show some overlap 2.5 seconds ahead as seen in Figure 4.4(c) and hence showing a probability of collision of around 0.1 in Figure 4.4.

At time = 8 seconds

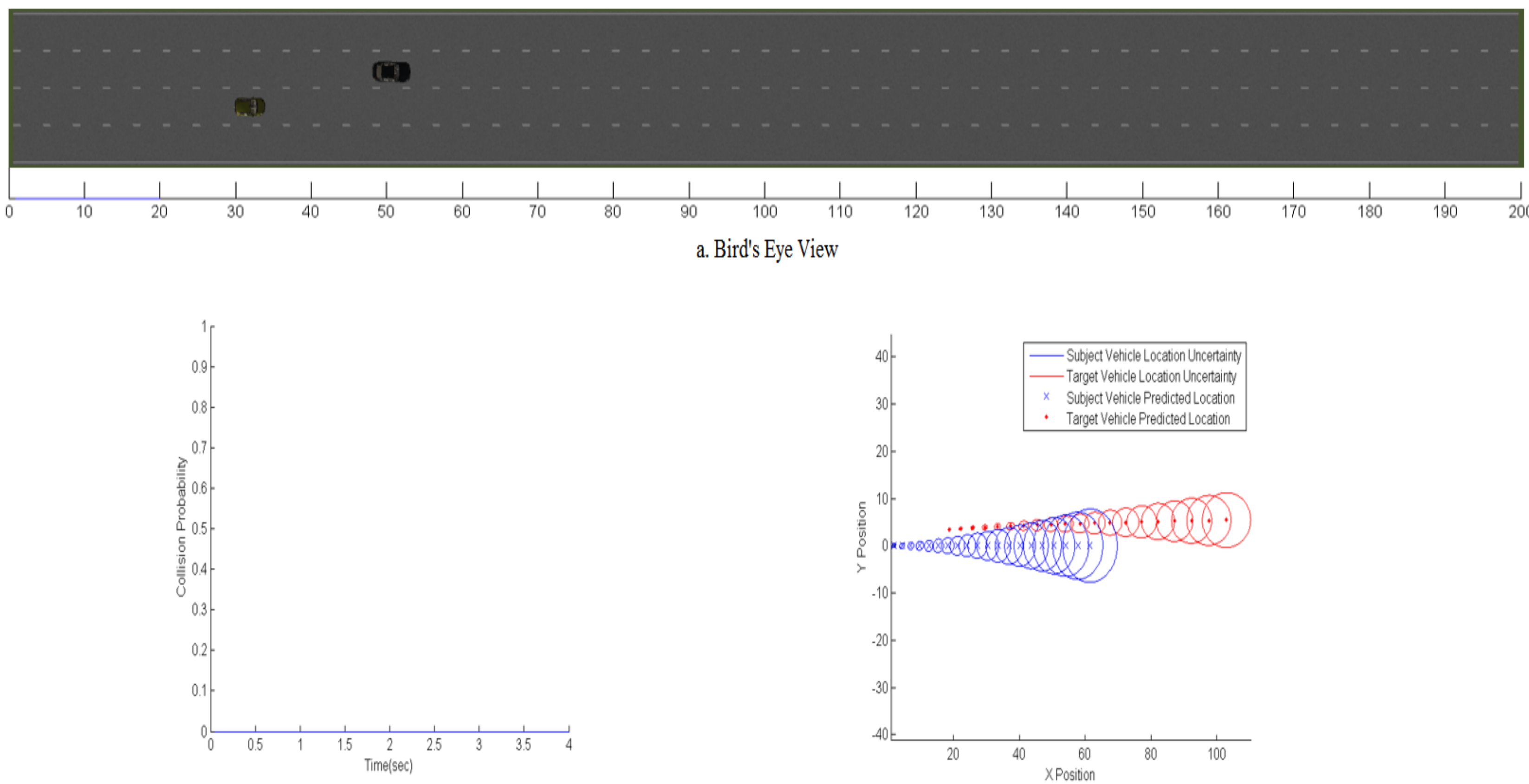
The final velocity of the subject vehicle is 20 mps. As the target vehicle gets closer to the subject vehicle, its speed increases as seen from Figure 4.5(a). Figure 4.5 depicts that the trajectories of both the vehicles show overlapping 1 to 2 seconds in future resulting in the increase of collision probability to around 0.25 as shown in Figure 4.5.

At time = 8.5 seconds

At this time, the subject vehicle is at 20 mps and over-speeding. The future trajectories of both vehicles indicate that the risk of collision has increased since there are high percentage of overlapping ellipses as seen from Figure 4.6(c). This increases the probability of collision to 0.4 as shown in Figure 4.6(b).

At time = 9 seconds

With further over-speeding, the subject vehicle gets much more closer to the target vehicle as seen from the bird eye view in Figure 4.7(a). The future trajectories of both vehicles overlaps and very high possibility of collision after 0.5 seconds is predicted as indicated in Figure 4.7(b).

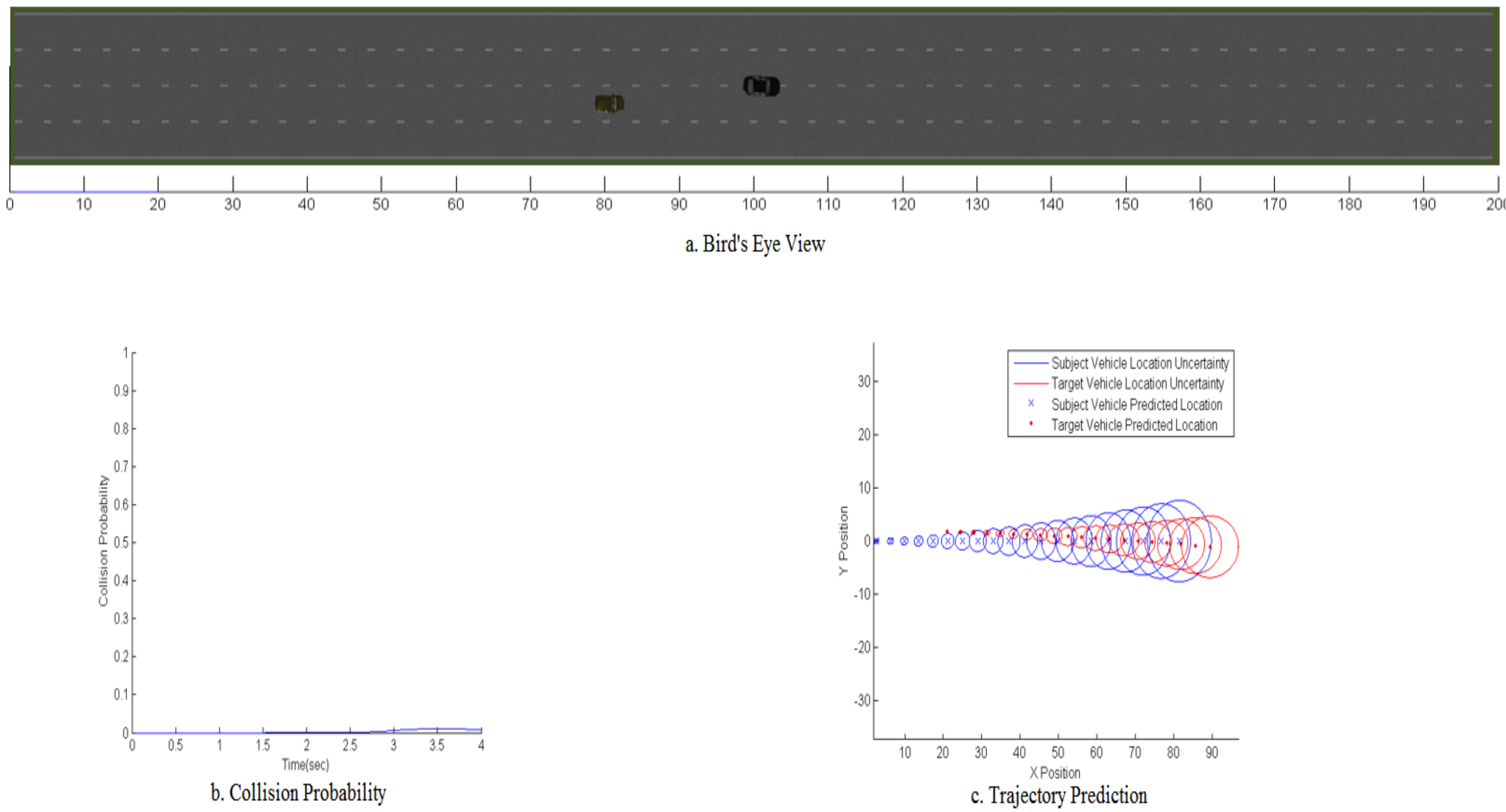


a. Bird's Eye View

b. Collision Probability

c. Trajectory Prediction

Figure 4.2: Collision with dynamic object at 3 seconds

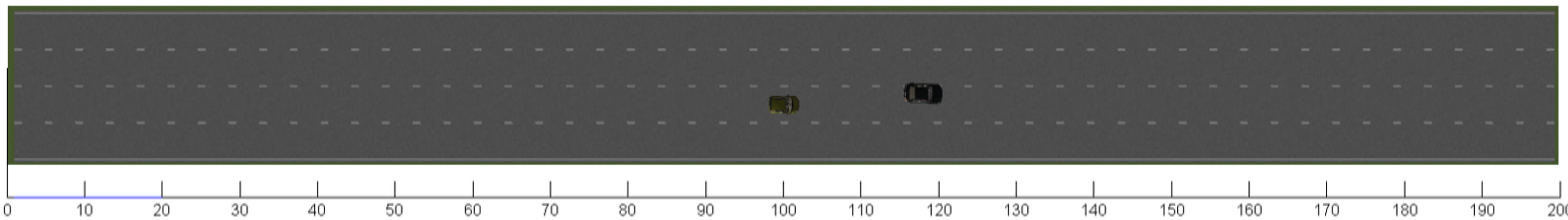


a. Bird's Eye View

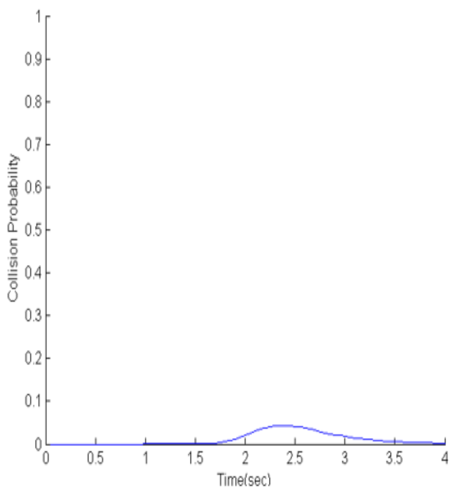
b. Collision Probability

c. Trajectory Prediction

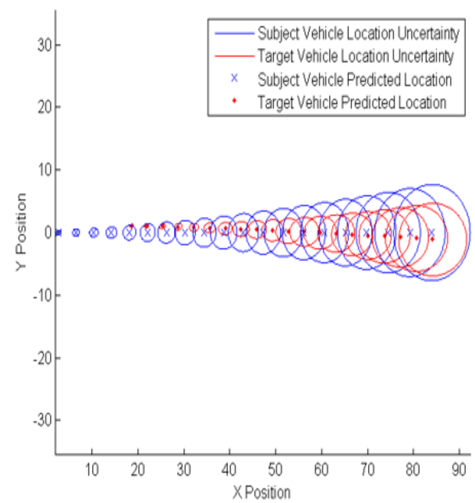
Figure 4.3: Collision with dynamic object at 6 seconds



a. Bird's Eye View



b. Collision Probability



c. Trajectory Prediction

Figure 4.4: Collision with dynamic object at 7 seconds

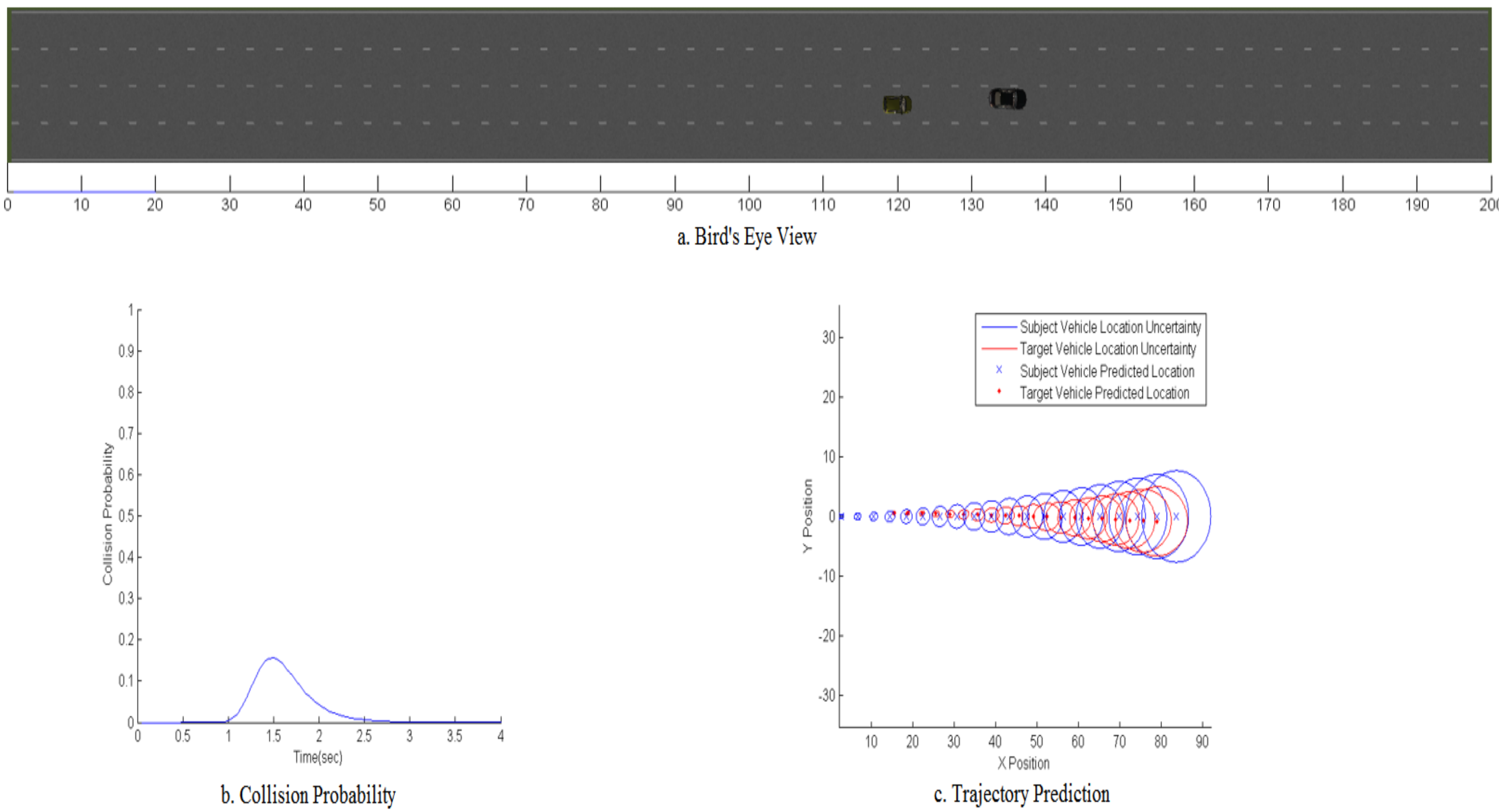
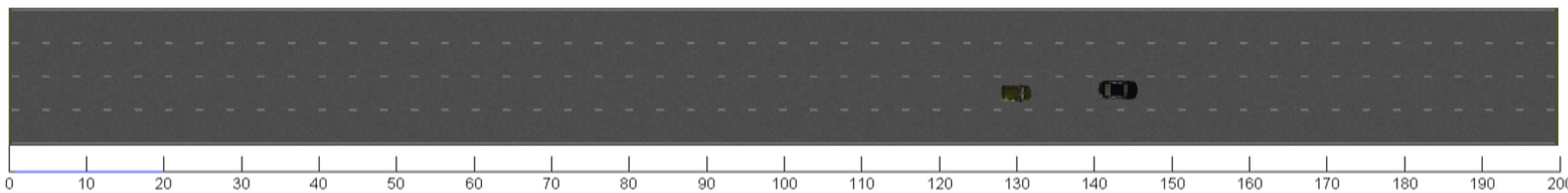
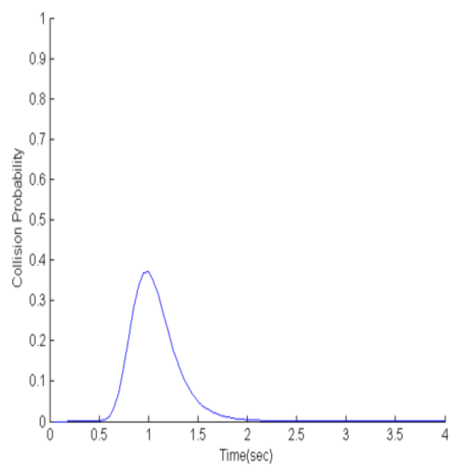


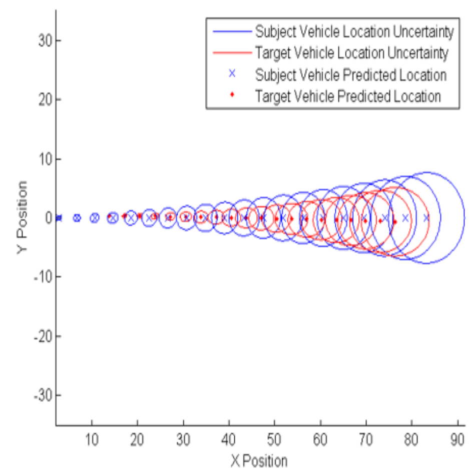
Figure 4.5: Collision with dynamic object at 8 seconds



a. Bird's Eye View



b. Collision Probability



c. Trajectory Prediction

Figure 4.6: Collision with dynamic object at 8.5 seconds

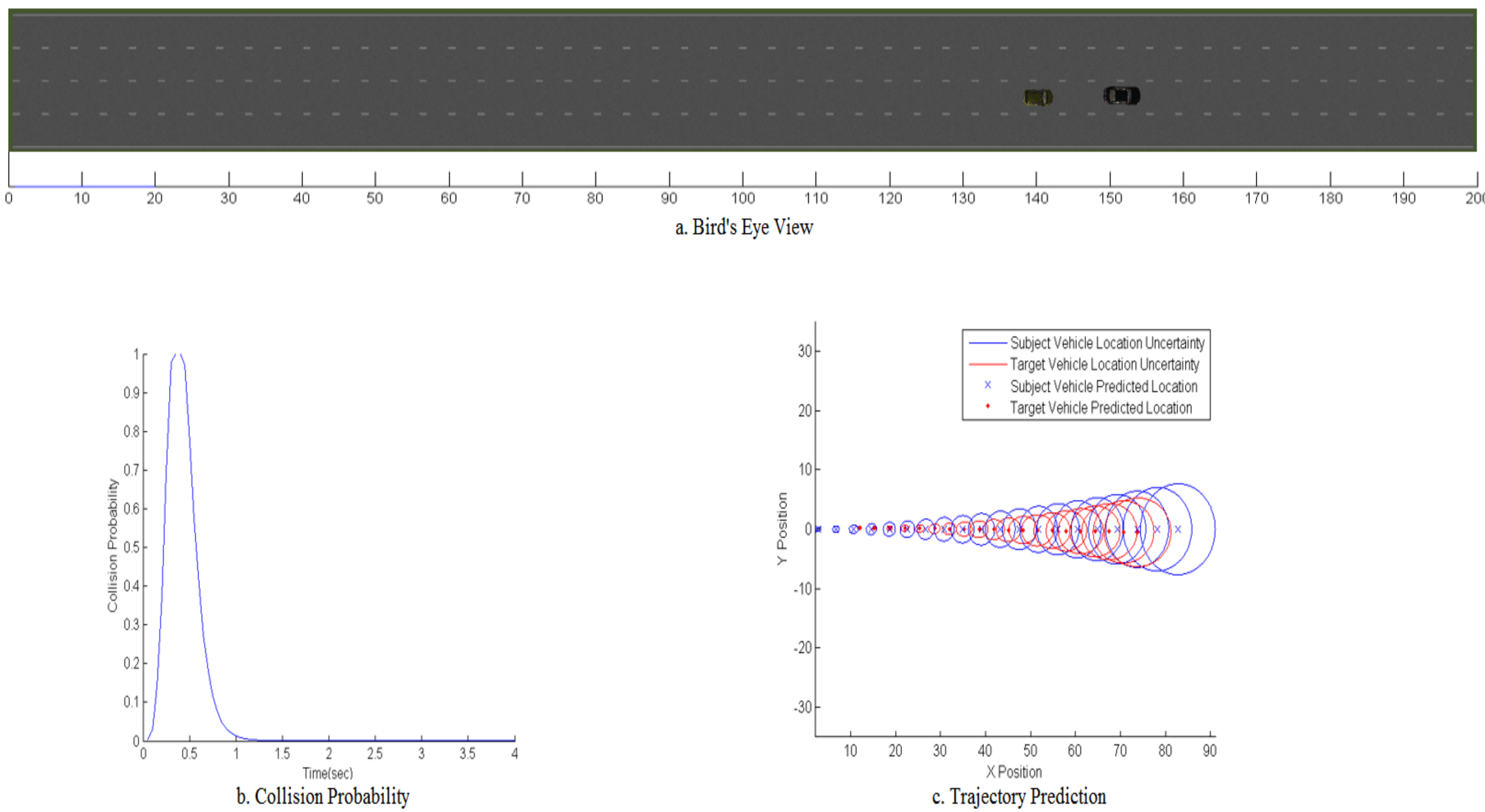


Figure 4.7: Collision with dynamic object at 9 seconds

4.2 Evaluation of the Use of Standby Measurements

The system considered is a constant acceleration model for the target vehicle. The measurements of the target vehicle are captured using two cameras, namely main and standby, mounted on the windshield. The effect of using standby measurements for discrete time system is studied in this simulation. For the linear dynamic time system, initialization of parameters for Kalman filter and covariance matrices are shown in Figure 4.8. The augmented system denotes the consideration of standby measurements. On solving for the discrete algebraic Ricati equation using "DARE" function, the error covariance (P) for the system without standby measurements is greater than error covariance with standby measurements Paug.

```
%Kalman Initialization for Evaluating Use of Standby Measurements
T = 0.05; % Sampling Time


% System Matrix
ASys = [1 0 T 0 T^2/2 0;
        0 1 0 T 0 T^2/2;
        0 0 1 0 T 0;
        0 0 0 1 0 T;
        0 0 0 0 1 0;
        0 0 0 0 0 1];


BSys = [0 0 0 0 0 0]';


% Output Matrix
CSys = [1 0 0 0 0 0;
        0 1 0 0 0 0;
        0 0 1 0 0 0;
        0 0 0 1 0 0];
```

```
% Output Matrix Augmented
CSys_Aug = [1 0 0 0 0 0;
            0 1 0 0 0 0;
            0 0 1 0 0 0;
            0 0 0 1 0 0;
            1 0 0 0 0 0;
            0 1 0 0 0 0;
            0 0 1 0 0 0;
            0 0 0 1 0 0];
DSys = [];


% Covariance Matrices
QSys = [0.005 0 0 0 0 0;
        0 0.001 0 0 0 0;
        0 0 0.009 0 0 0;
        0 0 0 0.005 0 0;
        0 0 0 0 0.001 0;
        0 0 0 0 0 0.009];


RSys = 0.05*eye(4);
RSys_Aug = 0.05*eye(8);




xoSys= [0 ;0; 0 ;0 ;0 ; 0];
PoSys = eye(6);
uoSys = 0;
```

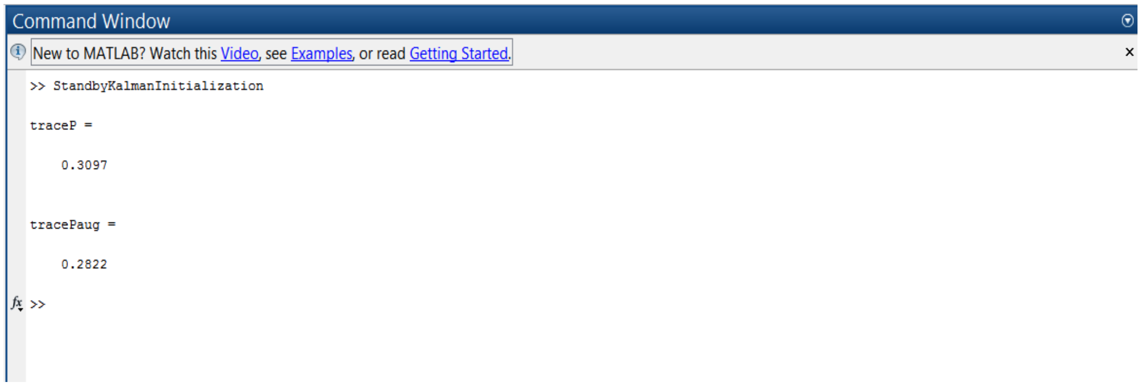


Figure 4.8: Evaluation of the Use of Standby Measurements at Steady State

```
% Solution of Discrete Time Algebraic Ricatti Equation
P = dare(ASys',CSys',QSys,RSys);
Paug = dare(ASys',CSys_Aug',QSys, RSys_Aug);


traceP = trace(P)
tracePaug = trace(Paug)
```

This is also studied in simulation over a time of 12 seconds. Figure 4.9 shows a graph of steady state error covariance with and without standby measurements. The results confirm the theoretical observations as stated in an earlier section.

4.3 Evaluation of the Sensor Data Fusion Techniques

This section presents the practical comparison of the centralized and decentralized sensor data fusion techniques. The data coming from camera and radar was captured for a target vehicle. The data consisted of relative position in X and Y direction. This data was provided to both Group Sensor Fusion and Quasi-Decentralized Data Fusion. The graphs for Relative X and Relative Y Position estimate computed by both methods is shown in Figures 4.10 and 4.11. From the graphs we see that both the techniques produce almost similar results and both are optimal filter implementations.

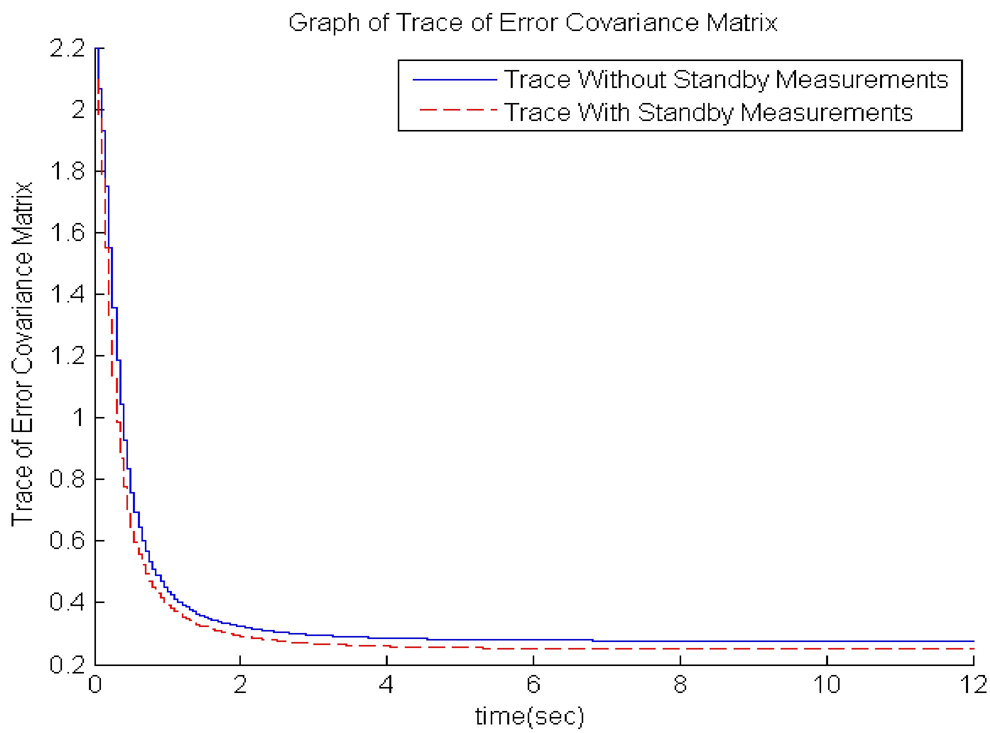


Figure 4.9: Evaluation of the Use of Standby Measurement using sensor data

Thus depending upon the application, one can decide which technique is appropriate.

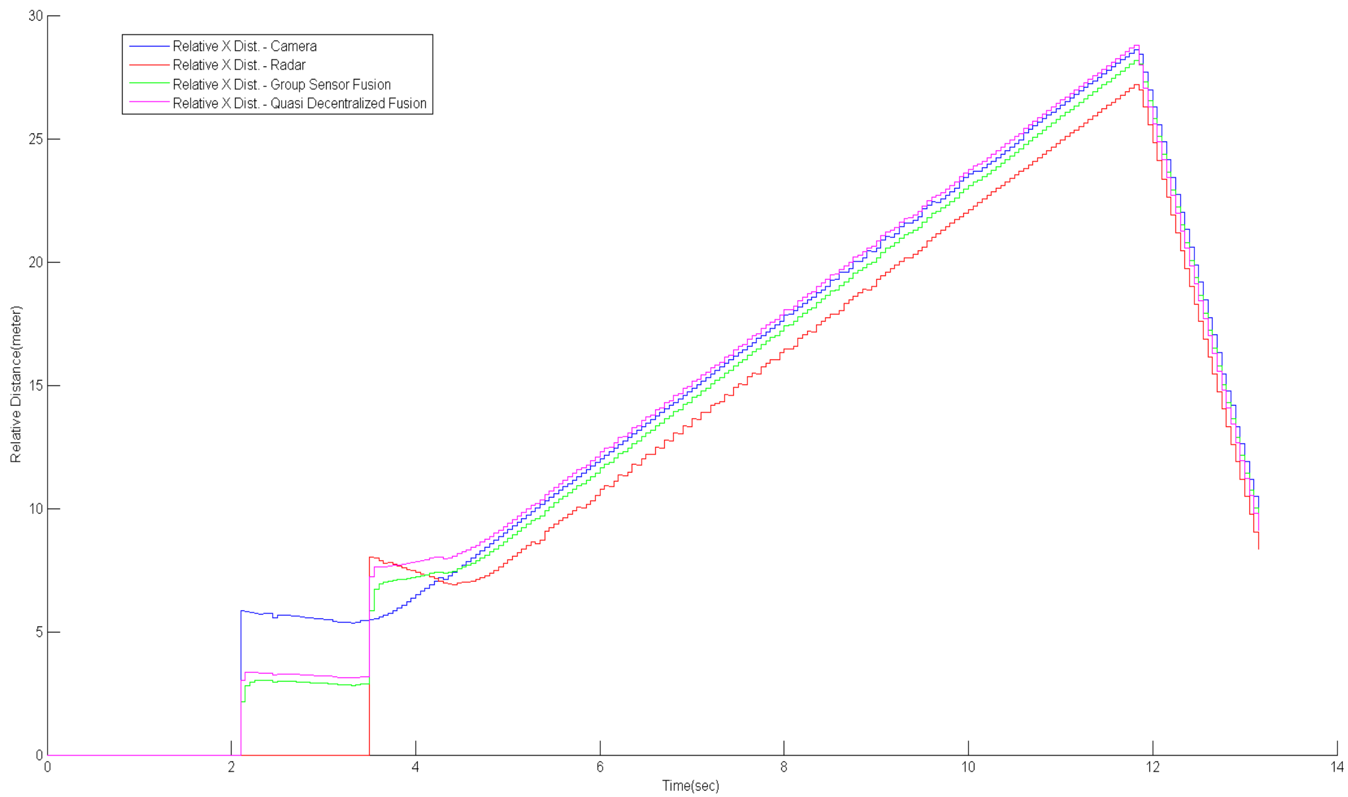


Figure 4.10: Comparison of Sensor Data Fusion - Relative X Distance

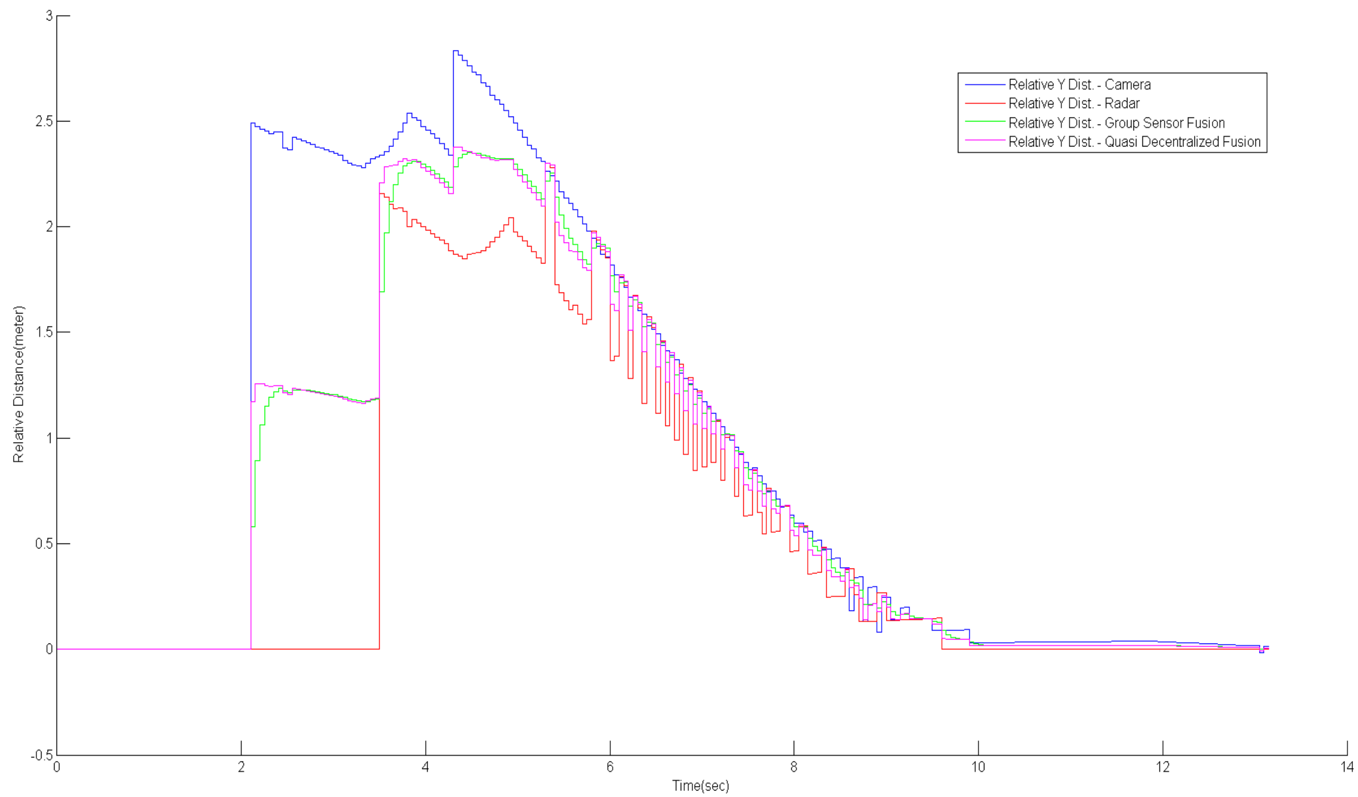


Figure 4.11: Comparison of Sensor Data Fusion - Relative Y Distance

Chapter 5

Conclusion

Driving safety is one of the key topics of research in the automotive industry, which is making an exponential progress towards autonomy. In this thesis document, an approach for tracking of moving and stationary obstacles and evaluating the probability of collision is proposed, discussed and implemented.

The thesis aims at providing information or Pre-Warning of a future collision. The proposed method is capable to track the target vehicles using a camera and a radar sensor and predict the future trajectories of both subject and target vehicles. The uncertainty in prediction is also considered and accounted for using statistical approach and the probability of collision is evaluated over the entire prediction time frame. This approach provides a larger view of evolution of the dynamic scene and is better for estimating the collision probability.

The three tasks of environment perception, trajectory prediction and collision estimation have been implemented using best suggested approaches through extensive related research. Kalman filter forms the heart of the algorithm. Kalman filter is used in noise filtering, sensor data fusion and trajectory prediction. Enhancements to the algorithm by making use of existing research are proposed and verified using simulation. The use of standby measurements, whenever available, to reduce the covariance error in the estimation process is demonstrated with the help of the discrete time linear time-invariant system. It is also suggested that a global optimal solution for sensor data fusion is obtained by a quasi-decentralized estimation using non-optimal local Kalman filters.

The overall approach is tested in two different scenarios and successful simulation results are presented. This algorithm can be used by other ADAS features in robust decision making based on the probability of collision that is output.

Chapter 6

Future Work

The thesis recommends a methodology to estimate the probability of collision of a future risk of collision. In this case the subject and the target vehicles were modelled in a linear model for simplicity and ease of implementation. This could be enhanced further by making use of complex vehicle models like Constant Turn Rate Velocity or Constant Turn Rate Acceleration, etc.

Currently the subject vehicle is mounted with a camera and radar sensor. It would be fascinating to mount other sensors like LIDAR, use Vehicle to Vehicle Communication for getting more information about the environment and better risk estimation. Also currently the approach is not tested for target vehicles taking turns or traffic at intersection which could be a future addition in the research. It would also be interesting to estimate driver intent along with trajectory prediction using techniques like driver behavior modelling.

Also, the scope of the thesis is limited to the environment as perceived by the sensors mounted on the front grill and windshield. This could be widened using rear and side facing sensors to get a pseudo 360 degree understanding of the environment and thus improve the prediction accuracy.

Since there is a lot of research going on in the field of automated driving and ADAS, the future scope of the application considered is boundless and as future advancements happen the research can be strengthened further.

References

[1] Y. Bar-Shalom and X. Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, Storrs, CT, 1995.

[2] M. Huang, Y. Lin, C. Yang, H. Chou. *A Collision Pre-Warning Algorithm based on V2V Communication.* Ubiquitous Information Technologies Applications, 2009. ICUT '09. Proceedings of the 4th International Conference, Fukuoka, 1 - 6, 2009.

[3] A. Hounou, P. Bonnifait, V. Cherfaoui. *Risk Assessment for Collision Avoidance Systems.* 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 386-391, 2014.

[4] S. Hoseini, R. Ashraf. *Computational Complexity Comparison Of Multi-Sensor Single Target Data Fusion Methods By MATLAB.* International Journal of Chaos, Control, Modelling and Simulation (IJCCMS) Vol. 2, No.2, June 2013.

[5] H. Durrant-Whyte. *Multi Sensor Data Fusion.* Australian Centre for Field Robotics, The University of Sydney NSW 2006 Australia, Version 1.2, 2001.

[6] S. Ammoun and F. Nashashibi. *Real Time Trajectory Prediction for Collision Risk Estimation between Vehicles.* Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference, Cluj-Napoca, 417 - 422, Aug. 2009.

[7] A. Houenou, P. Bonnifait, V. Cherfaoui, W. Yao . *Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition.* 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 4363 - 4369, November 3-7, 2013.

[8] Rajamani. *Vehicle Dynamics and Control*. Springer Science + Business Media, Inc., 233 Spring, Street, New York, NY 10013, USA, 2006.

[9] F. Sandblom and M. Brannstrom. *Probabilistic Threat Assessment and Driver Modeling in Collision Avoidance Systemsl*. 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 914 - 919, 2011.

[10] R. Schubert, E. Richter, and G. Wanielik. *Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking*. Information Fusion, 11th International Conference, Cologne, 1 - 6, 2008.

[11] National Highway Traffic Safety Administration. *Traffic Safety Facts*. 2013.

[12] I. Papadimitriou, M. Tomizuka. *Fast Lane Changing Computations using Polynomials*. Proceeding of the 2003 American Control Conference, Vol. 1, 4-6 June 2003.

[13] D. Khosla. *Accurate Estimation of Forward Path Geometry using Two Clothoid Road Models*. Intelligent Vehicle Symposium, 2002. IEEE, Vol. 1, 17-21 June 2002.

[14] The MathWorks, Inc *Sensor Fusion with Monocular Vision and Radar,* Mathworks Korea, 2015.

[15] A. Lambert, D. Gruyer, G. S. Pierre. *A Fast Monte Carlo Algorithm for Collision Probability Estimation*. Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference, Hanoi, 406 - 411, Dec. 2008.

[16] Tass BV, TNO. *PreScan Help* https://tass-safe.com/en/downloads, 2012.

[17] Z. Gajic. *The Use of the Standby Measurements Reduces the Estimation Error of the Linear Kalman Filter* Conference on Information Science and Systems, Princeton University, P.776, 1992.

[18] A. Ran and R. Vregdenhil. *Existence and Comparison Theorems for Algebraic Ricatti Equation for Continous and Discrete Time Systems*. Linear Algebra and its Applications, Vol. 99, 63-83, 1988.

[19] Z.Gajic. *On the Quasi-Decentralized Estimation and Control of Linear Stochastic Systems*. Systems Control Letters Vol. 8, 441 - 444, 1987.

[20] D. Looze, P. Houpt, N. Sandell, and M. Athans. *On Decentralized Estimation and Control with Application to Freeway Ramp Metering*. IEEE Trans. Automat. Control Vol. 23, 268 - 275, 1975.

[21] J. Speyer. *Computation and Transmission Requirements for a Decentralized Linear - Quadratic-Gaussian Control Problem*. IEEE Trans. Automat. Control Vol. 24, 266 - 269, 1979.

[22] T.Chang. *Computation and Transmission Requirements for a Decentralized Linear - Quadratic Gaussian Control*. IEEE Trans. Automat. Control Vol. 25, 609 - 700, 1980.

[23] A. Willsky, M. Bello, D. Castanon, B. Levy and G. Varghese. *Combining and Updating of Local Estimates and Regional Map Along Sets of One - Dimensional Tracks*. IEEE Trans. Automat. Control Vol. 27, 799 - 813, 1982.

[24] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME Journal of Basic Engineering, (Series D), 35-45, 1960.

[25] G. Welch, G. Bishop. *An Introduction to the Kalman Filter*. Siggraph 2001 Course 8,University of North Carolina at Chapel Hill, ACM Inc, 2001.