MASTERS THESIS

MODELLING OF VEHICLE TRACKING AND PREDICTION OF TRAFFIC COLLISIONS

By Gashawbeza Abebe DEGEFIE
Carried at the Institute of Automotive Engineering
Member of Frank Stronach Institute
Head: Prof. Wolfgang Hirschberg
Supervisor: Dr. Arno Eichberger

Statutory declaration

Hereby, I confirm that the following thesis on the topic of

Modelling of Vehicle Tracking and Prediction of Traffic Collisions

is fully my work and all the sources and tools used for the work are mentioned in the reference.

Graz, June 2010

Signature

Acknowledgement

I thank my supervisor, Dr. Arno Eichberger for giving me the opportunity to carry out my Master's Thesis at the Automotive Engineering Institute of TU Graz and also for his incredible continual support and advice.

I would like to thank my family for their unfainting wonderful support throughout my stay, here in Graz, without whom it would have been impossible. I also thank AAI for providing me their support, during the time that I was carrying out this thesis.

Last but not least, I would also like to thank my friends here in Graz and else where, who in one way or another encouraged me along the way.

Abstract

Precrash systems are part of integrated safety systems, which are believed to reduce fatalities due to road accidents further. These systems include belt pre-tensioners, air bag systems and also special type of automatically reconfigurable seats. All these systems are intended to mitigate vehicle crash. If they have to be effective in doing so, they must be activated at the right time and when the accident is no more avoidable. For this a certain predictor is needed, which can look in the future and estimate the states of both the host and target vehicles. This can be accomplished by using Kalman filter with the accompanied use of different tracking models like constant velocity, constant acceleration, coordinated turn, common state space and IMM models. In addition to the state estimation, estimation of the collision probability at a certain specific time is also needed. Different models are explored in this work and some enhancements are applied to some of them, when it seemed that it is appropriate to do so. The first model is based on stochastic drivers' input and it does this by randomly taking different input values of both vehicles and generating trajectories with the use of a simplified kinematic model of the vehicles and then checking collision for each generated pair of trajectories and implementing polygonal intersection check-ups. The second model uses the size of the overlap of the projections of the two vehicles rectangular representations along an imaginary line perpendicular to the relative velocity and the time to collision to assess likeness of collision . The third model which is based on Monte Carlo simulation needs the mean and covariance of the states of the two vehicles as input. It generates different possible state values, with the assumption of Gaussian distribution. Next, it simulates all the generated states with a simplified single track vehicle model. Finally, it calculates the collision probability based on theses simulated states and polygonal intersection check-ups. The fourth one considers radar tracking with the use of either the Extended Kalman Filter or the Unscented Kalman Filter and the constan velocity motion model. This model deals with propagation of collision space and time to collision by generating different states from the mean and the covariance like the third model and using some non-linear transformation to get the propagated collision space (hit point) and TTC for all the generated states. The output of this model is the predicted mean hit point and TTC.

Contents

1 Introduction

1.1 Motivation

1.2 Vehicle Safety

Automobile safety is the science of vehicle design, construction, and equipment and road way systems to minimize the occurrence and consequences of automobile accidents. Vehicle safety is divided into two main categories, known as Active Safety and Passive Safety. Active safety refers to technology assisting in the prevention of a crash, whereas passive safety refers to components of the vehicle that help to protect occupants during a crash. It involves crashworthiness oriented design, development and integration of the restraint systems, engineering of crashworthiness based vehicle structure, seat and pedestrian and vulnerable road user protection systems. Post-crash Safety involves activities like immediate post-crash treatment, accident reconstruction, severity evaluation and emergency notification.

At the moment the passive safety technologies have reached a relatively matured state with less room remaining for innovation and further significant improvements are unlikely to be made in the near future [Che08]. Active Safety Systems and Integrated Safety Systems, which are a combination of active and passive systems and other automotive domains have a very high potential of further innovation and development in the coming years. Fig 1.1 shows the additional potential that can be obtained by using Integrated Safety compared to the potential of the individual approaches of Active and Passive Safety systems. The blue, red and green lines show the potential growth of Passive Safety, Integrated Safety and Active Safety respectively over the years past and to come.

The European Commission's transport policy has set the ambitious target of reducing the fatalities due to vehicle accidents by 50% in 2010. The traditional approach of isolated development of passive and active safety systems has not been enough to reach this target. Integrated Safety Systems have a very promising potential in reducing the fatality due to vehicle accidents [Che08]. In Fig 1.2, we can see the expected impact of Integrated Safety Systems in the future. Further development and application of Pre-crash-systems, Drive assistant systems and Integrated Safety Systems play a very decisive role in bringing down the road fatalities.

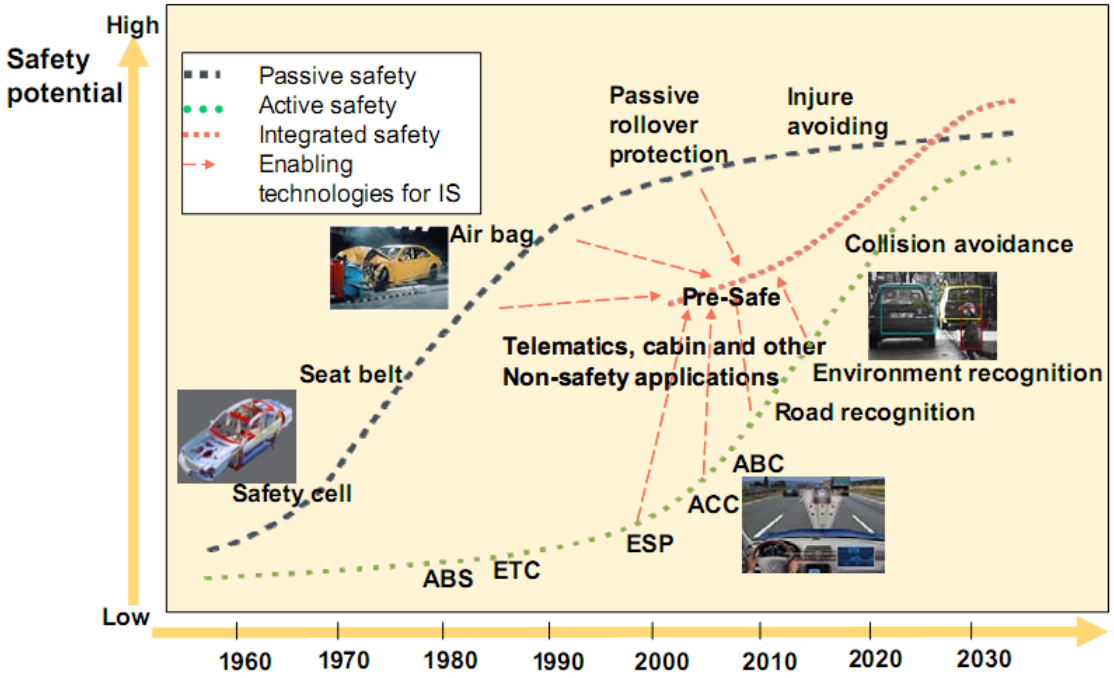


Figure 1.1: Development of active, passive and integrated safety electronic systems [GL02]

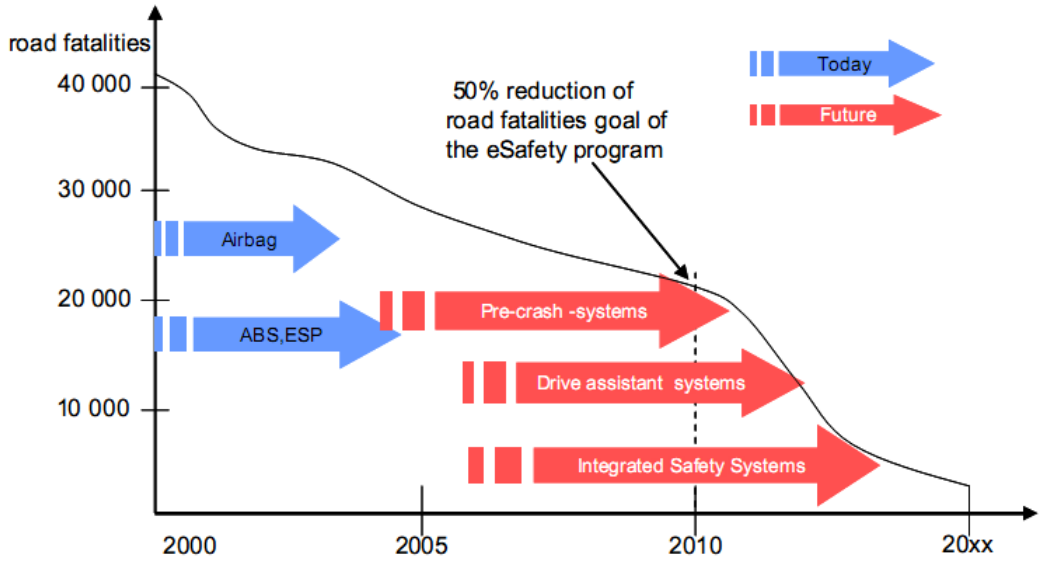


Figure 1.2: Targets set by EU of 50% reduction in the road fatalities before 2010 [EUC01]

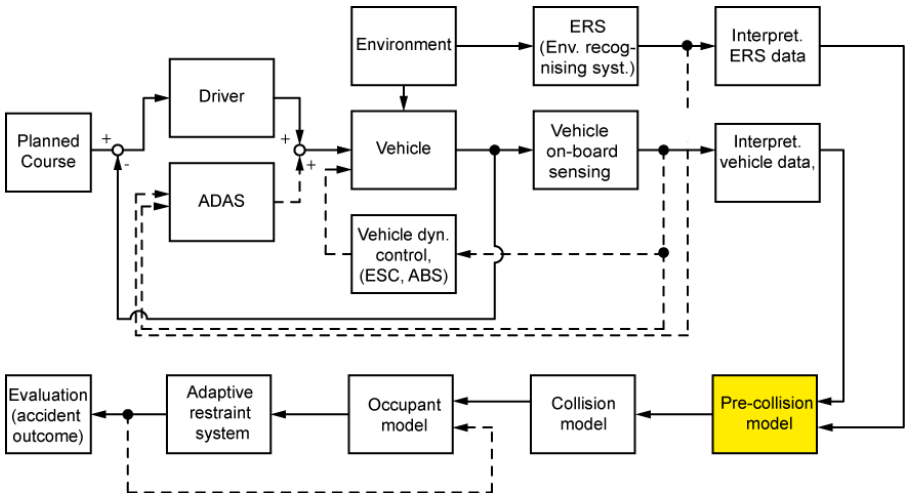


Figure 1.3: Integrated Vehicle Safety Project of FTG, TU Graz [Eic10]

1.3 Target of this Thesis

The target of this thesis is to investigate the stae-of-the-art in vehicle tracking and collision estimation and to implement some of them by applying some enhancements or modifications. The thesis is part of an on-going strategic project being carried out at the Automotive Engineering Institute of Graz University of Technology. In Fig 1.3 we see the different modules of the mentioned project and this thesis specifically deals with the pre-collision model .

1.4 Chapter Overview

Here in this section a summarized chapter overview is given in order to let the reader have the complete overview this Master's Thesis

Chapter 1 is a brief introduction to Vehicle Safety, Pre-Crash Systems and Time to Collision. The main goal of this chapter is to start with the broader fields of Vehicle Safety and to come through different branches, to reach to the point that we specifically discuss about the concept of Time-to-Collision and its relevance.

Chapter 2 deals with different kinds of state estimators needed for vehicle tracking.

Chapter 3 introduces different models used for tracking with the use of state estimators. Here some of the models, implemented for vehicle tracking are discussed and their mathematical derivations is also presented in detail.

Chapter 4 deals with different models used for collision prediction. Some modifications are suggested for the collision prediction model, which is based on drivers stochastic input and for the one, which is based on overlap, additional mathematical derivations are included, to realize its implementation in matlab programming framework.

Chapter 5 presents the simulations and results of the applications of some of the selected models.

Chapter 6 is dedicated for Discussion and Conclusion.

1.4.1 Preventive Safety Applications

Preventive safety applications are used to mitigate an accident through the use of in-vehicle systems which assess the nature and significance of the danger with sensors, while taking the driver's state into consideration. Depending on the weight and timing of the accident threat, the active and preventive safety systems will [Köh04]

- Inform the driver as early as possible with advanced driver information systems
- Warn the driver, if the driver is not reacting to the information of the threat, and
- Actively assist or ultimately intervene in order to avoid an accident or mitigate its consequences.

Preventive safety applications support the drivers with various functions [Köh04] :

- Keep a safe speed, safe distance
- Drive by keeping the vehicle within the lane
- Avoid overtaking in dangerous situations
- Ensure safe crossing of road intersections
- Protect pedestrians and vulnerable road users
- Mitigate an accident, if it still occurs.
- Collision avoidance and intervention
- Braking and vehicle control through Advanced Driver Assistance Systems
- Providing brake assistance
- Preventing Roll-over or providing safety protection, if it happens

Preventive safety uses information, communications and positioning technologies to improve road safety. Some of these technologies are fully or partial autonomous with systems operating on-board and others operate on cooperative basis with a vehicle-to-vehicle or vehicle-to-infrastructure communication. This is believed to reduce the number of accidents and their severity.

1.5 Pre-Crash Systems

In recent years, as there is an increased demand for vehicle safety features and better crash-rating systems from legislation and customers, the vehicle manufacturers have quickly begun to incorporate more advanced technology into their designs. Among these advanced technologies, which are believed to reduce the number of fatalities and injuries on the road further, the introduction of a pre-collision system (PCS) can be mentioned. PCS constantly evaluates a ego vehicles's position and motion as well as any objects on the road, in order to prevent or mitigate damage that might arise due to an accident. On one hand some systems warn the driver with acoustic signal that a collision might be imminent –and this is with the thought of alerting the driver and thereby making him ready to take evasive action. On the other hand other systems take partial or full control of certain parts of the car. Some pre-crash brake systems apply additional pressure to the car's brake in addition to that of the driver's to assist the driver in reducing the speed of the car as fast as possible and mitigate the damage that might result from the accident. In some other systems, the PCS is connected to a pre-crash seat belt system and when the PCS predicts that a crash is imminent a higher tension is applied to the passenger's seat belts by the use of seatbelt pretensioners. These systems are called brak Brake Assist (BA). In some vehicles this is also accompanied by automatic sunroof closing and raising of reclined seats. Also systems capable of lane deviation and frontal collision avoidance are implemented in some cars. The PCS do make use of radar, video, lidar, infrared imagery or a fused combination of any of them, to asses the situation of the environment. Such systems should be reliable, robust, fine-tuned and highly accurate, since any malfunction could by itself be disastrous. As can be imagined, PCS passes through thorow rigorous testing to ensure failure-free functionality using best practice design and testing methods. Failure-safe strategies in case of system malfunction minimise the consequences.

Fig 1.5 shows an example of a PCS. The Milliwave radar makes assesment on the road environment and sends signals to the Pre-crash, seat belt and radar cruise control units to make decision, whether to activate the Brake actuators and the Motor Pretensioners.

1.5.1 Parts of a Pre-Crash System

A Pre-Crash System consists of the following main parts [MTO04].

Sensor System The Sensor system percepts the traffic environment and provides dynamic object data of the environment.

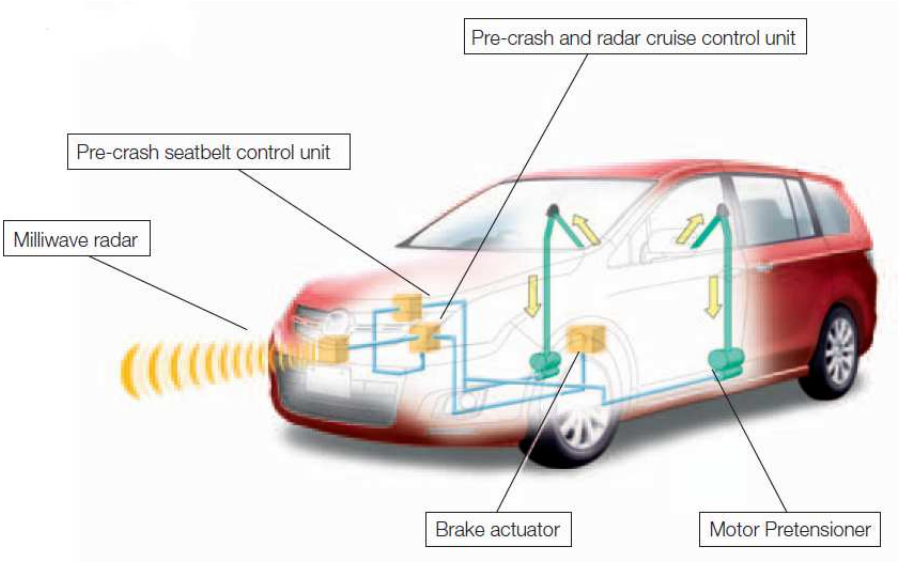


Figure 1.4: Mazda pre-crash safety system [Mazda]

Assessment and Deployment Algorithm The algorithm processes the data obtained from the sensor system, makes analysis of the traffic situation to decide, whether a crash is avoidable or not and if the accident is not avoidable it sends a trigger signal to actuators for activating the safety system.

Actuators Actuators serve for performing different kinds of safety starategies. As an example a belt pre-tenstioner can be taken. It is used to limit slack from the seat-belt when a crash is detected and this slack reduction reduces the belt load on the occupant during crash. Belt Slack represents a displacement, which is cost for "ride-down" of the occupant. ride-down is the displacement to deccelerate the occupant from collisison speed to zero motion. Belt slack results in higher belt forces and occupant loadings. To obtain an optimized protection of the occupant, the firing of the pre-tensioners and airbags should be well synchronized.

The following flow diagram, Fig 1.5, shows how Pre-Crash systems are interconnected to each other and the associated signal flow direction. The Radar Sensors assess the environment and collect Object-data, then in the next module the Object-data gets processed, then the processed Obeject-data is used as input to the algorithm, which decides, whether the crash is avoidable or not. If the crash is avoidable, the safety system is activated and if it is not avoidable, no measure will be taken.

The Following diagram, Fig 1.6, shows various pre-crash actuators, which are used for

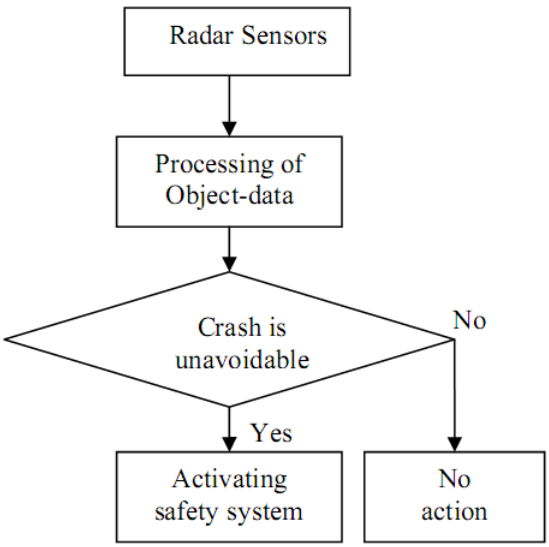


Figure 1.5: Block diagram of deployment strategy [MTO04]

adaptive loading of the occupant, when a crash is inevitable.

1.5.2 Precrash phases

Rather than using impact based sensors for activation some safety systems, when the crash contact takes place, the precrash system using vehicle tracking and collision prediction, helps to activate all the active safety system in time, like the belt pre-tensioners and the airbags.

A crash has typically three phases [Köh04].

1. The phase till the no return situation
2. The phase from no return situation to the time of first crash contact
3. The crash phase after the first contact took place

Fig 1.7, shows all the crash phases. The blue part shows, the time when the precrash safety system gets activated.

The actions that can be taken during the pre-crash phase can be

- Autonomous intervention like automatic braking and evasive maneuver
- Warning with signals (acoustic or visible signs)

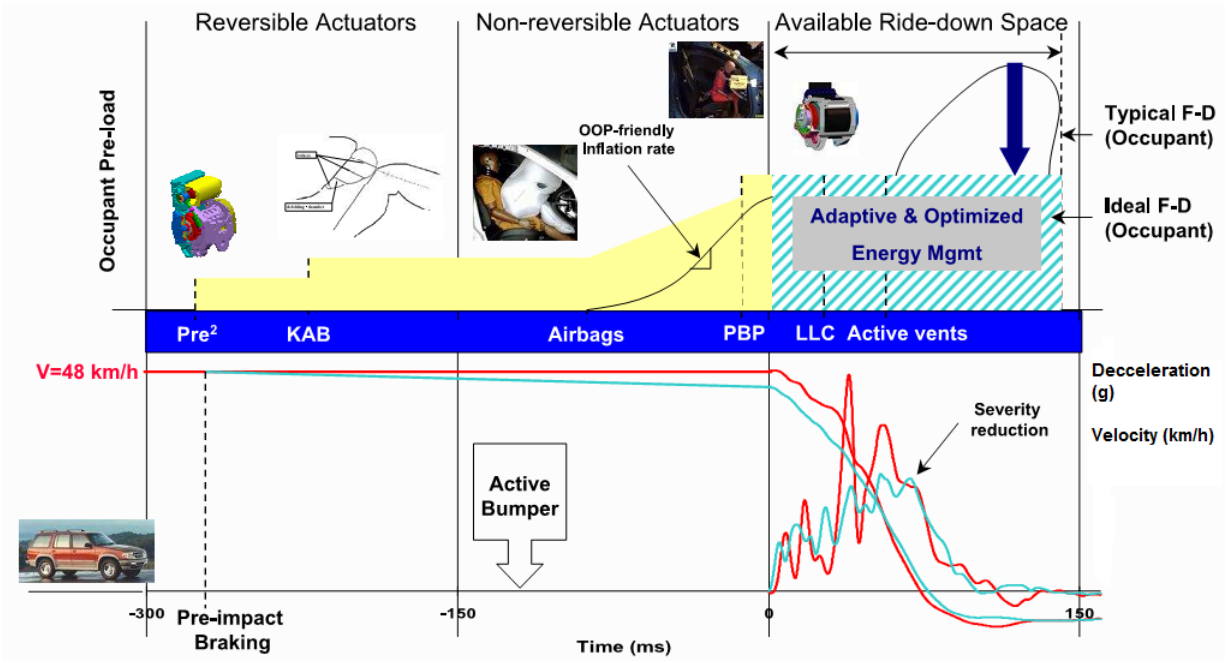


Figure 1.6: Pre-Crash Actuators and Adaptive Occupant Loading During a Crash [Hal05]

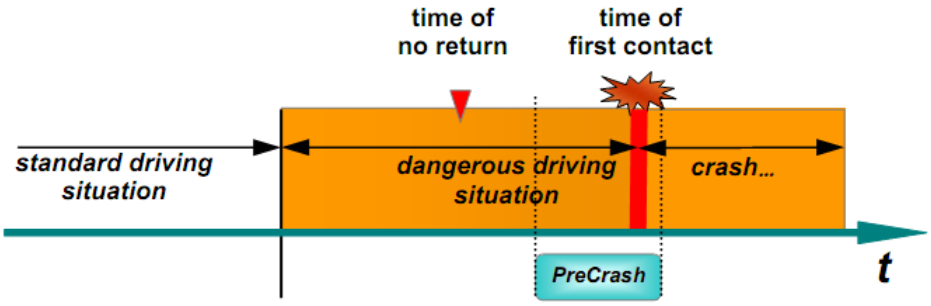


Figure 1.7: Phases of a crash [Köh04]

- Activating reversible restraint systems to reduce occupant displacement in the forward direction due to the braking force (Reversible restraint systems are like belt pretentioners, knee or thorax impact absorbers(the last two are known as active mechanical systems))
- Starting to record with black box data recorder
- Assessment and prediction of crash severity
- Active restraint systems (e.g. large volumed air bags with smart venting) are fired earlier than the normal time after getting crash confirmation.

1.5.3 Time to collision

Here the Crash Phase is further broken down into quantified time ranges of different pre-collision activities according to Prevent [Pre].

- In the Time-to-Collision range period between 10 seconds and roughly one second (the human reaction time), is the driver warning and assistance phase. The warning might be with optical, acoustic and/or haptic signals.
- The active assistance starts to act, when the time to collision goes down under one second. A haptic warning system can be taken as an example. In this system, the active accelerator starts to develop force, which gets intensified more and more over time during the warning phase - calling for corrective measures. Then breaking assistance follows and ultimately autonomous braking gets into action in order to avoid or mitigate the crash.
- During the last second before the collision, active and autonomous safety measures of a pre-crash system,consisting of suitable actuators will act to reduce the impact and the consequences of the crash. Two kinds of intervening measures achieve this goal: Autonomous or semi-autonomous (with amplification) braking which results in reduced crash kinetic energy.
- When time less than 500 milliseconds is remaining before the impact, activation of pre-crash systems, which make use of reversible structural actuators such as controllable bumpers, crash boxes and motor hoods takes place.
- When time less than 100 milliseconds is remaining before the impact, the phase known as pre-fire period starts. In this pre-fire period, the pre-crash system could provide the appropriate data and trigger in order to optimize the protections function of reversible restraint systems (e.g. belt-pre-tensioning) with the aim of reducing the overall occupant loading.
- Finally, during the last 10 milliseconds known as pre-set period, parametrization of the electronic control unit of the airbag system, based on the perceptive information

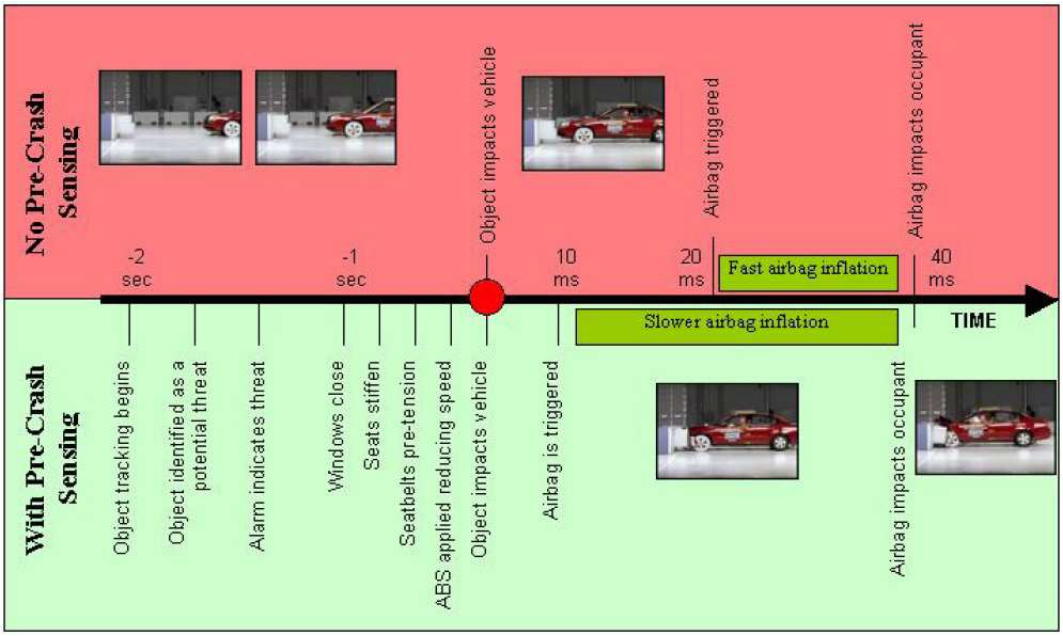


Figure 1.8: Timelines for collisions with and without precrash sensing [DD07]

gathered about the imminent collision, takes place. This will result in the best and appropriate protection of the occupants.

Fig 1.8, compares two crash scenarios, with no pre-crash sensing and with pre-crash sensing. A time scale similar to the one discussed above is shown on the horizontal axis. The diagram shows that, whenever no pre-crash sensing is used, nothing happens before impact happens and after the impact it takes 20 ms before the airbag is triggered. The airbag inflation happens in 20 ms, resulting in a very agressive impact of the airbag with the occupant. On the other hand in the case of pre-crash sensing 2 s earlier of the crash, start of object tracking, identification of the object as a potential threat and threat indication by the alarm happen sequentially. Then 1s earlier of the crash, closing of windows, siffening of seats, pretensioning of seat belts, application of ABS in reducing speed and then finally imact of the identified object with the vehicle take place respectively. After about 10ms the airbag starts to inflate and the airbag impacts the occupant after 40 ms of the occurence of the crash. The advantage of using a pre-crash sensing is the identification of the risk earlier and taking measures in time that would protect the occupant. The airbag is also slowly inflated in the case of pre-crash sensing.

2 State Estimators for Vehicle Tracking

This chapter presents the concept of state estimation and some filters used for this purpose. The dicussion and the equations are based on [Ord05][HS07] and [WB95].

Estimating the state of a process from noisy and/or inaccurate process variables measurement is known as State Estimation [Ord05]. Estimating the state of a process, which we cannot directly observe, involves probabilistic estimation of the state, i.e., getting the probability density function (pdf) of the state of the process. The state estimation has two in a cycle recursively operating steps, namely prediction of the future state from the current value and updating/correction of the predicted value with the measurement taken.

2.1 State Space Models

A state space models a process, where the process's state $\mathbf{x}$ is designated by a numerical vector. In a state space models there are actually two separate models, known as Process Model and Measurement Model. The Process Model models how the state propagates with time under the influence of external disturbances like input and noise, whereas the Measurement Model models how the process is measured.

2.1.1 General State Space Models

The non-linear model is considered as the general form of state-space model and it normally has two functions, f and h:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1}),$$

$$\mathbf{z}_k = h(\mathbf{x}_k, v_k), \tag{2.1}$$

where f is Propagation Function, h is Measurement Function, u is Process input, w is state noise vector, v is process noise vector, k is discrete time, $\mathbf{x}_k$ is state vector and $\mathbf{z}_k$ is measurement vector.

The functions, f and h , are normally discretized differential equations representing the dynamics of both the process of the system and observation on the system.

2.1.2 Linear State Space Models

A state-space model is a linear state space model if both f and h are linear. Using matrices F, B and H , it can be written in the following form.

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}u_{k-1} + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.2}$$

Such models are used for inherently linear systems or nonlinear systems, which are linearized by means of Taylor approximation.

2.1.3 Discretization of continuous-time linear time-invariant systems

Continuous time invariant models can be written in the following form [HS07]:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}\,x(t) + \mathbf{L}\,\mathbf{w}(t) \tag{2.3}$$

Where

- the initial conditions are $\mathbf{x}(0) \sim \mathbf{N}(m(0), P(0))$ (i.e. The initial state is defined by the initial state mean and initial state covariance of the normally distributed states.)
- $\mathbf{F}$ and $\mathbf{L}$ are constant matrices, which characterize the behavior of the model
- $\mathbf{w}(t)$ is a white noise process with a power spectral density $\mathbf{Q}_c$

The model is discretized with the following formula resulting in the matrices $\mathbf{A}_k$ and $\mathbf{Q}_k$ [HS07]

$$\mathbf{A}_k = exp(\mathbf{F}\,\Delta t_k),$$

$$\mathbf{Q}_k = \int_0^{\Delta t_k} exp(\mathbf{F}(\triangle t_k - \tau))\mathbf{L}\,\mathbf{Q}_c\mathbf{L}^T exp(\mathbf{F}(\Delta t_k - \tau))^T d\tau, \tag{2.4}$$

where$\Delta t_k = t_{k+1} - t_k$ is the discretization step size, τ is time, $\mathbf{F}$ and $\mathbf{L}$ are constant matrices, which characterize the behavior of the model and $\mathbf{Q}_c$ is the power spectral density of $\mathbf{w}(t)$.

If Q_k cannot be computed analytically with the above formula, then we can calculate it with the following matrix decomposition:

$$\begin{pmatrix} \mathbf{C}_k \\ \mathbf{D}_k \end{pmatrix} = exp\left\{ \begin{pmatrix} \mathbf{F} & \mathbf{L}\mathbf{Q}_c\mathbf{L}^T \\ 0 & -\mathbf{F}^T \end{pmatrix} \Delta t_k \right\} \begin{pmatrix} 0 \\ \mathbf{I} \end{pmatrix}, \tag{2.5}$$

where$\Delta t_k = t_{k+1} - t_k$ is the discretization step size, τ is time, $\mathbf{F}$ and $\mathbf{L}$ are constant matrices, which characterize the behavior of the model and $\mathbf{Q}_c$ is the power spectral density of $\mathbf{w}(t)$.

From $\mathbf{C}_k$ and $\mathbf{D}_k^{-1}$, then Q_k is calculated as

$$\mathbf{Q}_k = \mathbf{C}_k\mathbf{D}_k^{-1}. \tag{2.6}$$

2.2 Recursive Bayesian Estimation

Recursive Bayesian Estimation simulates a certain process in a recursive manner, while at the same manner it makes corrections based on the new measurements taken on the process. The two recursive calculation steps are [Ord05]

1. Extrapolation of the next state from the current one with the use of state propagation belief $p(\mathbf{x}_k/\mathbf{x}_{k+1})$ obtained from the function f
2. Correction of the predicted value by making use of the measurement likelihood $p(\mathbf{z}_k, \mathbf{x}_k)$ obtained from the function h

The prior pdf of the state x_k of the process is calculated with the Chapman-Kolmogorov equation based on the measurement value, that we have up to time k-1:

$$p(\mathbf{x}_k/\mathbf{z}_{k-1}) = \int p(\mathbf{x}_k/\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}/\mathbf{z}_{k-1})\, d\mathbf{x}_{k-1} \tag{2.7}$$

Then the next step is to calculate the updated pdf of state x_k taking into account the measurement taken up to time k

$$p(\mathbf{x}_k/\mathbf{z}_k) = \frac{p(\mathbf{z}_k/\mathbf{x}_k)p(\mathbf{x}_k/\mathbf{z}_{k-1})}{\int p(\mathbf{z}_k/\mathbf{x}_k)p(\mathbf{x}_k/\mathbf{z}_{k-1})\, d\mathbf{x}_k} \tag{2.8}$$

Unfortunately apart from serving as theoretic foundation for state estimation, the equation cannot be used to solve real world problems, due to the multidimensional integrals involved in case of multidimensional state vectors and also because of the reason that computers can calculate pdf only for discretized state space. Bayesian Estimation can be implemented only in cases, where the state space is discretized or with certain limitation assumptions.

2.3 Kalman Filter

The Kalman Filter (KF) is a set of mathematical equations that gives an efficient computational (recursive) means to estimate the state of a process, by minimizing the mean of the squared error [WB95]. The Kalman Filter gives optimal results for linear systems, though many systems in real world are in fact nonlinear. To use the Kalman Filter for non-linear systems, the method widely used is first to linearize the system and then to use the Extended Kalman Filter (EKF). Unfortunately the linearization can cause unstable estimation. To solve this problem in nonlinear system state estimation the Unscented Kalman Filter (UKF) is used, which totally eliminates and replaces linearization of the functions with Unscented Transformation, which makes use of propagation of a carefully selected set of points through the actual nonlinear function.

The Bayesian estimation technique is reduced to the Kalaman Filter, if the following assumptions are taken [Ord05]:

- f and h are linear functions
- the noise vectors w and v are assumed to be uncorrelated, Gaussian and white with a zero mean

The constraints are written in mathematical notation a follows

The functions f and h are linear functions

$$f(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1}) = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k u_{k-1} + \mathbf{w}_{k-1}$$

$$h(\mathbf{x}_k, v_k) = \mathbf{H}\, x_k + \mathbf{v}_k$$

The noise vectors w and v are assumed to be uncorrelated, Gaussian and white

$$\mathbf{w}_k \sim \mathbf{N}(0, Q_k) \quad v_k \sim \mathbf{N}(0, R_k) \tag{2.9}$$

The mean of the distribution of the noise vectors w and v is assumed to be zero

$$\mathbf{E}(w, w_j^T) = \mathbf{Q}_i \delta(i-j) \quad \mathbf{E}(v, v_j^T) = \mathbf{R}_i \delta(i-j)$$

$$\mathbf{E}(w_k v_k^T) = 0$$

Where $\mathbf{Q}$ and $\mathbf{R}$ are covariance matrices representing both the state and measurement noise in the second-order. Applying the above constraints results in the following linear state space model:

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k u_{k-1} + \mathbf{w}_{k-1},$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + v_k, \tag{2.10}$$

where A , B and H are matrices, which can also depend on time.

The state and the output of this model is Gaussian distributed, because of the fact that the linearity of the model and and the Gaussian input, which means that it is enough to calculate the mean vector m and covariance matrix P rather than calculating a full state pdf.

The Kalman filter consists of two steps, Prediction and Update steps and the equations used are [HS07]:

KF Prediction Equations:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}u_{k-1}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.11}$$

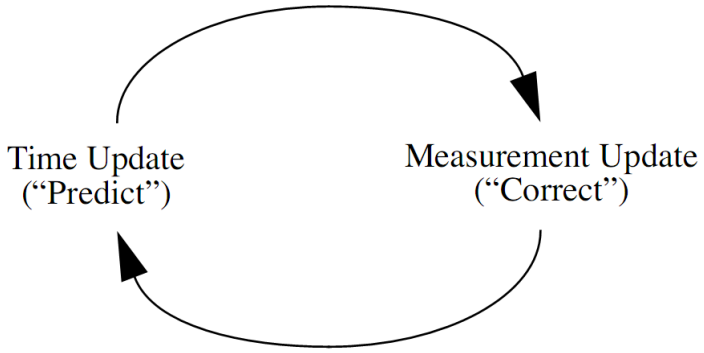


Figure 2.1: Discrete Kalman filter cycle [WB95]

KF Update Equations:

$$v_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-$$

$$K_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \tag{2.12}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k v_k$$

$$\mathbf{P}_k = \left(\mathbf{I} - K_k \mathbf{H}_k \right) \mathbf{P}_k^-$$

where

- $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$ are the predicted mean and covariance of the state respectively, on time step k before the measurement
- $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ are the estimated mean and covariance of the state, respectively on time step k after seeing the measurement
- v_k is the innovation or the measurement residual on time step k
- K_k is the filter gain, which sizes how much the predictions should be corrected on time step k

Fig 2.2 shows the complete picture of the operation of the Kalman filter. There are two recursively happening steps. The first one is the Time Update one, which consists of projection of the state and the error ahead. The second one is the Measurement Update one, which involves kalman gain computation, updating of estimate with measurement and updating of error covariance.

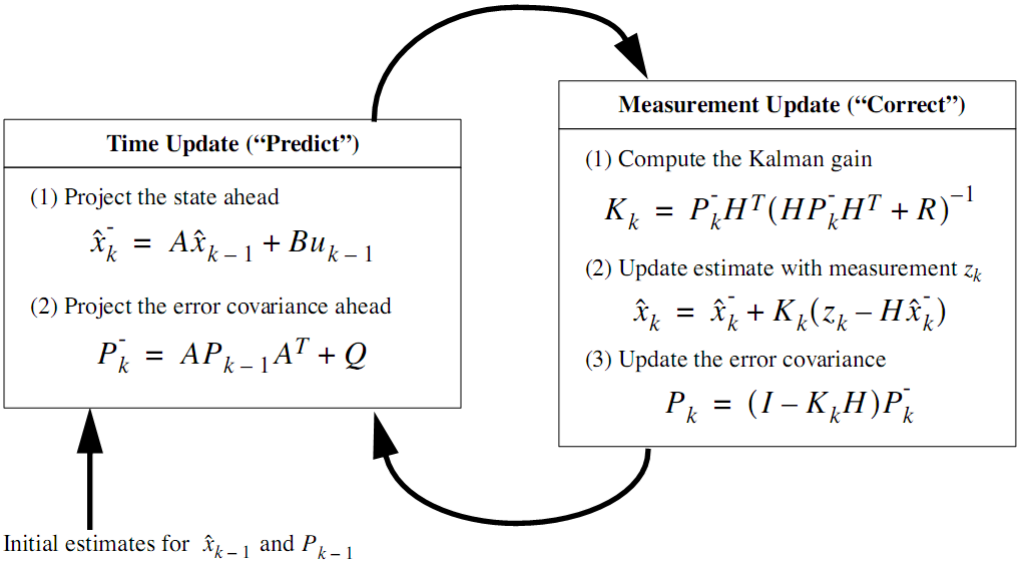


Figure 2.2: A complete picture of the operation of the Kalman filter [WB95]

2.4 Extended Kalman Filter

Most of the processes in real world are nonlinear and need to get linearized, before getting their state estimated with Kalman filter. Here all the derivations of the algorithms are taken from [WB95].

The type of Kalman filter, which uses the Jacobian of f and h at the estimated state resulting in the trajectory of the model function centered around this particular state is the Extended Kalman Filter.

Normally we do not know the individual values of the noise w_k and and v_k at each time step, but still we can approximate the state and measurement vector in their absence as

$$\tilde{\mathbf{x}}_k = f(\tilde{\mathbf{x}}_{k-1}, u_{k-1}, 0) \tag{2.13}$$

and

$$\tilde{\mathbf{z}}_k = h(\tilde{\mathbf{x}}_k, 0) \tag{2.14}$$

where $\tilde{x}_k$ is a posterior estimate of a state from the previous time step k

Then we can write a linearized estimate of the non-linear system as follows

$$\mathbf{x}_k \approx \tilde{x}_k + \mathbf{A}(x_{k-1} - \hat{x}_{k-1}) + \mathbf{W}\, w_{k-1} \tag{2.15}$$

$$\mathbf{z}_k \approx \tilde{z}_k + \mathbf{H}(x_k - \tilde{x}_k) + \mathbf{V}\, v_{k-1} \tag{2.16}$$

- $\mathbf{x}_k$ and $\mathbf{z}_k$ are the actual state and measurement vectors,
- $\tilde{\mathbf{x}}_k$ and $\tilde{\mathbf{z}}_k$ are the approximate state and measurement vectors
- $\hat{\mathbf{x}}_k$ is an posterior estimate of the state at step k
- the random variables w_k and v_k represent the process and measurement noise
- $\mathbf{A}$ is the Jacobian matrix of partial derivatives of f with respect to x, that is

$$\mathbf{A}_{[i,j]} = \frac{\partial f_{[i]}}{\partial x\,[j]}(\hat{\mathbf{x}}_{k-1}, u_{k-1}, 0), \tag{2.17}$$

where i and j are the row and column number of $\mathbf{A}$.

- W is the Jacobian matrix of partial derivatives of f with resprect to w

$$\mathbf{W}_{[i,j]} = \frac{\partial f_{[i]}}{\partial w\,[j]}(\hat{\mathbf{x}}_{k-1}, u_{k-1}, 0), \tag{2.18}$$

where i and j are the row and column number of $\mathbf{W}$.

- H is the Jacobian matrix of partial derivatives of h with respect to x

$$\mathbf{H}_{[i,j]} = \frac{\partial h_{[i]}}{\partial x\,[j]}(\tilde{\mathbf{x}}_k, 0), \tag{2.19}$$

where i and j are the row and column number of $\mathbf{H}$.

- V is the Jacobian matrix of partial derivatives of h with respect to$v$

$$\mathbf{V}_{[i,j]} = \frac{\partial h_{[i]}}{\partial v\,[j]}(\tilde{\mathbf{x}}_k, 0), \tag{2.20}$$

where i and j are the row and column number of $\mathbf{V}$.

The Extended Kalman filter consists of two steps, Prediction and and Update steps like the Kalman filter and the equations used are [HS07], see Fig 2.3

EKF Prediction Equations:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, u_{k-1}, 0)$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \tag{2.21}$$

EKF Update Equations:

$$\mathbf{v}_k = \mathbf{z}_k - h_k(\hat{\mathbf{x}}_k^-, 0)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T \right) \tag{2.22}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{v}_k$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

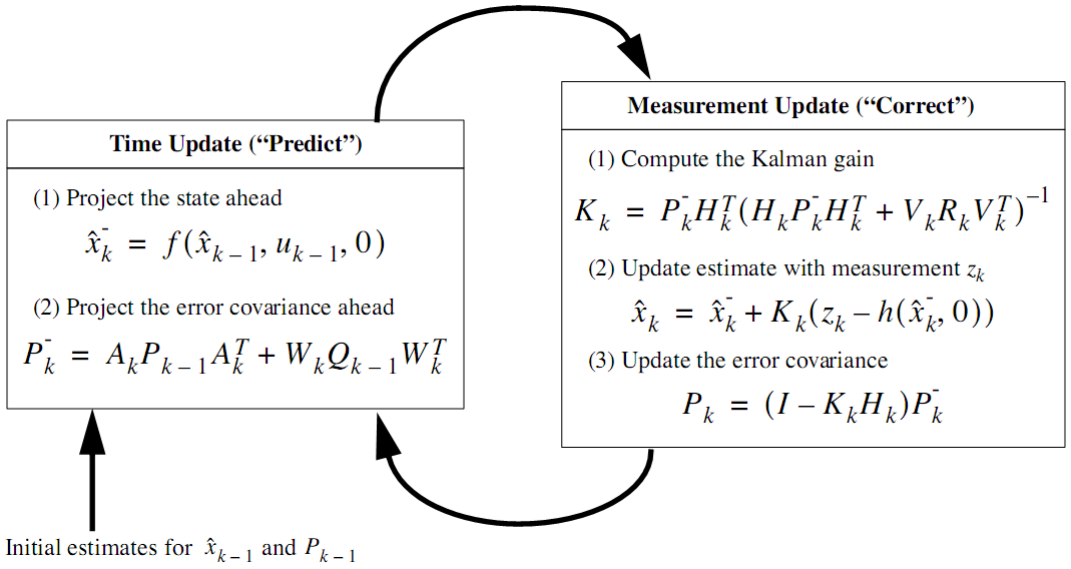


Figure 2.3: A complete picture of the operation of the extended Kalman filter [WB95]

The Limitations of EKF

- The linear transformation produces reliable results only in cases when the error propagation can be well approximated by a linear function, otherwise EKF might perform poorly, at worst divergence in estimation might also happen.
- The Jacobian matrices might not exist or can have singularities.
- Calculating the Jacobian matrices might need a long cpu time and error prone.

2.5 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is another version of the Kalman filter, which gives a Gaussian approximation to nonlinear filtering problems by making use of the unscented transformation. The algorithm is taken from [HS07].

Unscented Transform (UT) In UT, first a fixed number of samples known as sigma points are carefully selected to capture the true mean and covariance of the original Gaussian distribution of the state variable, which is random. Then the sigma points are propagated through the non-linear function and finally we capture the posterior mean and covariance accurately to the third order from the the propagated sigma points. UT captures the higher order moments of the non-linear transformation better that the Taylor series and it also does not need calculation of the Jacobian matrices,which makes it simpler and less error-prone.

If a state vector of $\mathbf{x}$ of dimension n gets propagated through a non-linear function $y = f(x)$ and its mean and covariances are m and P respectively, a matrix consisting of $2n+1$ sigma vectors $x^{(i)}$ having a corresponding weight $W^{(i)}$ is formed. The calculations involved are shown in the following steps

- Compute the set of $2n+1$ sigma points

$$\mathbf{x}^{(0)} = \mathbf{m},$$

$$\mathbf{x}^{(i)} = \mathbf{m} + \left[\sqrt{(n+\lambda)\,\mathbf{P}}\right]_i, \; i = 1, ..., n, \tag{2.23}$$

$$\mathbf{x}^{(i)} = \mathbf{m} - \left[\sqrt{(n+\lambda)\,\mathbf{P}}\right]_i, \; i = n+1, ..., 2n,$$

and the corresponding weights are

$$W_m^{(0)} = \lambda/(n+\lambda),$$

$$W_c^{(0)} = \lambda/\left(n+\lambda\right) + (1-\alpha^2+\beta),$$

$$W_m^{(i)} = 1/\left\{2\left(n+\lambda\right)\right\},\ i = 1, ..., n, \tag{2.24}$$

$$W_c^{(i)} = 1/\left\{2\left(n+\lambda\right)\right\},\ i = 1, ..., n,$$

$$\lambda = \alpha^2(n+k) - n,$$

where

λ is used for scaling,

α expresses the spread of the sigma points around the mean (small number like 10^{-3}),

β used to incorporate the prior knowledge of the random variables distribution (2 is an optimal value for Gaussian distribution),

k is a secondary scaling factor (usually it is taken to be equal to zero).

- Propagate every sigma point through the non-linear function

$$\mathbf{y}^{(i)} = f(\mathbf{x}^{(i)}),\ i = 0, ..., 2n \tag{2.25}$$

- Calculate the mean,μ_U, and covariance estimates for y, $\mathbf{S}_U$

$$\mu_U \approx \sum_{i=0}^{2n} W_m^{(i)} \mathbf{y}^{(i)} \tag{2.26}$$

$$\mathbf{S}_U = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathbf{y}^{(i)} - \mu_U\right)\left(\mathbf{y}^{(i)} - \mu_U\right)^T \tag{2.27}$$

- Calculate the cross-covariance between x and y

$$\mathbf{C}_U \approx \sum_{i=0}^{2n} \mathbf{W}_c^{(i)} \left(\mathbf{x}^{(i)} - \mathbf{m}\right)\left(\mathbf{y}^{(i)} - \mu_U\right)^T \tag{2.28}$$

The steps for UKF are given below

UKF Prediction Equations: Calculate the predicted state mean $\mathbf{m}_k^-$and the predicted covariance $\mathbf{P}_k^-$ as follows

$$\mathbf{X}_{k-1} = [\mathbf{m}_{k-1} ... \mathbf{m}_{k-1}] + \sqrt{c}\left[0\ \sqrt{\mathbf{P}_{k-1}}\ -\sqrt{\mathbf{P}_{k-1}}\right]$$

$$\hat{\mathbf{x}}_k^- = f(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1})$$

$$\mathbf{m}_k^- = \hat{\mathbf{x}}_k^- w_m \tag{2.29}$$

$$\mathbf{P}_k^- = \hat{\mathbf{x}}\mathbf{W}\left[\hat{x}_k\right]^T + \mathbf{Q}_{k-1}$$

UKF Update Equations: In the update part we calculate the predicted mean μ_k,the measurement covariance S_k and the cross covariance of the state and the measurement C_k

$$\mathbf{X}_{k-1} = [\mathbf{m}_k ... \mathbf{m}_k] + \sqrt{c}\left[0\ \sqrt{\mathbf{P}_k}\ -\sqrt{\mathbf{P}_k}\right]$$

$$\mathbf{Y}_k^- = h(\mathbf{x}_k^-, k)$$

$$\mu_k = \mathbf{Y}_k^- w_m \tag{2.30}$$

$$\mathbf{S}_k = \mathbf{Y}_k^-\mathbf{W}\left[\mathbf{Y}_k^-\right]^T + \mathbf{R}_k$$

$$\mathbf{C}_k = \mathbf{X}_k^-\mathbf{W}\left[\mathbf{Y}_k^-\right]^T$$

The filter gain K_k, the updated mean m_k and covariance P_k are then calculated:

$$\mathbf{K}_k = \mathbf{C}_k\mathbf{S}_k^-$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k\left[\mathbf{y}_k - \mu_k\right] \tag{2.31}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T$$

Fig 2.4 shows how a normal sampling, EKF and UKF propagate the mean in different ways. In EKF the mean and the covariances are approximated using Taylor Series, whereas in UKF the mean and covariance are approximated using carefully selected sigma points.

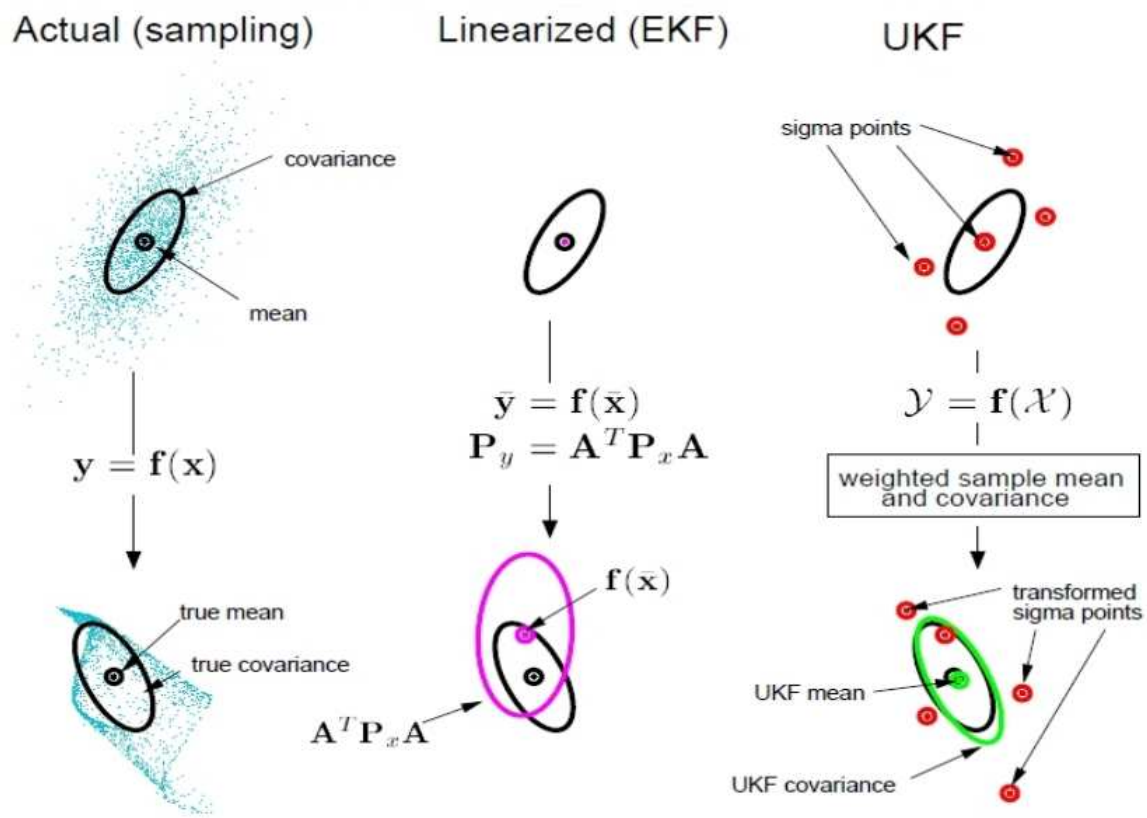


Figure 2.4: Comparison of mean propagation in EKF, UKF and Sampling [RvdM97]

2.6 Interacting Multiple Model Filter

In many practical applications the system's model can change in time. The varying system characterstics is not effectively represented with one model, that means that in estimation the model also changes with this change. Here all the important algorithms of the model are given as presented in [HS07].

Let out of the n models that are available $M = \{M^1 ... M^n\}$, one of them be the current model, $\mu_0^j = P\left\{M_0^j\right\}$ is the prior probability of M^j, and $p_{ij} = \left\{M_k^j / M_{k-1}^j\right\}$ is the transition probability matrix of a first order Markov chain of switching from model i to model j, which characterizes the mode transitions. If in a discrete random process the next state depends only on the current state, it is known as a Makrov Chain. Such systems are known as Markovian switching systems.

IMM-filter is a computationally efficient filter and most of the times it gives a suboptimal estimation for Markovian switching systems, which consists of three main steps, interaction (mixing), filtering and combination.

- Interacting (Mixing)

Given that μ_{k-1}^i is the probability of model M^i in the time step $k-1$, the mixing probabilities $\mu_k^{i/j}$ for each model M^i and M^j are computed as

$$\bar{c}_j = \sum_{i=1}^{n} p_{ij} \mu_{k-1}^i \tag{2.32}$$

$$\mu_k^{i/j} = \frac{1}{\bar{c}_j} p_{ij} \mu_{k-1}^i \tag{2.33}$$

Where $\bar{c}_j$ is a normalization factor

The mixed means and covariances used as input for each filter are computed from the updated mean $\left(\hat{\mathbf{x}}_{k-1}^i\right)$ and covariance $\left(\mathbf{P}_{k-1}^i\right)$ for model i at time step $k-1$ as follows

$$\hat{\mathbf{x}}_{k-1}^{0j} = \sum_{i=1}^{n} \mu_k^{i/j} \hat{\mathbf{x}}_{k-1}^i \tag{2.34}$$

$$\mathbf{P}_{k-1}^{0j} = \sum_{i=1}^{n} \mu_k^{i/j} \left\{ \mathbf{P}_{k-1}^i + \left[\hat{\mathbf{x}}_{k-1}^i - \hat{\mathbf{x}}_{k-1}^{0j}\right] \left[\hat{\mathbf{x}}_{k-1}^i - \hat{\mathbf{x}}_{k-1}^{0j}\right]^T \right\} \tag{2.35}$$

- Filtering

Filtering for each model M^i is carried out as follows

$$\left[\hat{\mathbf{x}}_k^{-,i}, \mathbf{P}_k^{-,i}\right] = KF_p\left(\hat{\mathbf{x}}_{k-1}^{0j}, \mathbf{P}_{k-1}^{0j}, \mathbf{A}_{k-1}^i, \mathbf{Q}_{k-1}^i\right) \tag{2.36}$$

$$\left[\hat{\mathbf{x}}_k^i, \mathbf{P}_k^i\right] = KF_p\left(\hat{\mathbf{x}}_k^{-,i}, \mathbf{P}_k^{-,i}, \mathbf{y}_k, \mathbf{H}_k^i, \mathbf{R}_k^i\right) \tag{2.37}$$

The measurement likelihood of each filter is calculated as

$$\Lambda_k^i = \left(v_k^i; 0, \mathbf{S}_k\right), \tag{2.38}$$

where v_k^i is the measurement error and S_k^i is model M^i's covariance

Each model's (M^i) probability at time k is computed as

$$c = \sum_{i=1}^{n} \Lambda_k^i \bar{c}_i \tag{2.39}$$

$$\mu_k^i = \frac{1}{c} \Lambda_k^i \bar{c}_i \tag{2.40}$$

c is a normalizing factor

- Combination: is the final step, which combines the estimate state means and covariances.

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{n} \mu_k^i \hat{\mathbf{x}}_k^i \tag{2.41}$$

$$\mathbf{P}_k = \sum_{i=1}^{n} \mu_k^i \left\{ \mathbf{P}_k^i \left[\hat{\mathbf{x}}_k^i - \hat{\mathbf{x}}_k\right] \left[\hat{\mathbf{x}}_k^i - \hat{\mathbf{x}}_k\right]^T \right\} \tag{2.42}$$

2.7 EKF/UKF Toolbox- Tool Used for State Estimation

EKF/UKF Toolbox is an optimal filtering toolbox, which works under Matlab Environment [HS07].The Kalman filters and smoothers are the main elements of this toolbox, which are the most widely used methods in stochastic state-space estimation. The developers developed it with the aim to provide a simple framework for building proof-of-concept implementations of optimal filters and smoothers for use in practical applications. The toolbox is developed by the Laboratory of Computational Engineering at Helsinki University of Technology (HUT). It is freely available and can be downloaded from their website. In the content of this thesis, this toolbox has been used instead of programming the equations of chapter 2.

3 Models for Vehicle Tracking and Motion Estimation

Here in this chapter different models that are used in estimating the future path of a vehicle are presented. Any of these models can be used for vehicle tracking purpose depending on the need. Each one of them has its own strong sides and weaknesses. Some of the models try to combine two or more models to get a better performance. The last 5 models also implement very simplified Vehicle Dynamic Models and some of them can be used for motion estimation without the use of state estimators.

3.1 Constant Velocity Model (CV)

The following mathematical formulas are based on [PTAA07] and [Jan05]

The state vector $\mathbf{x}_k$ reads

$$\mathbf{x}_k = [p_x \; p_y \; v_x \; v_y]^T, \tag{3.1}$$

where p_x and p_y are the x and y coordinate values respectively and v_x and v_y are the x and y components of the velocity respectively.

Assuming a constant acceleration and yaw rate derivatives, the transition matrix $\mathbf{A}$ of size 4×4 used for the state equation update reads

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

The covariance matrix of the process noise $\mathbf{Q}$ of size 4×4

$$\mathbf{Q} = \mathbf{G}.q.\mathbf{G}^T, \tag{3.3}$$

where the matrix $\mathbf{G}$ of size 4×2 is written as

$$\mathbf{G} = \begin{bmatrix} 0.5T^2 & 0 & T & 0 \\ 0 & 0.5T^2 & 0 & T \end{bmatrix}^T. \tag{3.4}$$

Assuming a white, Gaussian process noise with zero mean and variances $\sigma^2_{v_x}$ and $\sigma^2_{v_y}$ for v_x and v_y respectively, the matrix q of size 2×2 is given as

$$q = \begin{bmatrix} \sigma^2_{v_x} & 0 \\ 0 & \sigma^2_{v_y} \end{bmatrix}. \tag{3.5}$$

3.2 Constant Acceleration Model (CA)

This is one of the models used for predicting the future trajectory of vehicles,which assumes that the acceleration of the vehicle as constant in both axes and the state vector consists of the variables position, velocity and acceleration in both axes. The following mathematical formulas are based on [PTAA07] and [Jan05].

The state vector of size 6×1 $\mathbf{x}_k$ reads

$$\mathbf{x}_k = [p_x\ p_y\ v_x\ v_y\ a_x\ a_y]^T, \tag{3.6}$$

where p_x and p_y are the x and y coordinate values respectively and v_x and v_y are the x and y components of the velocity respectively and a_x and a_y are the x and y components of the accelerations respectively.

Assuming a constant acceleration and yaw rate derivatives, the transition matrix $\mathbf{A}$ of 6×6 size used for the state equation reads

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T & 0 & 0.5T^2 & 0 \\ 0 & 1 & 0 & T & 0 & 0.5T^2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.7}$$

The covariance matrix $\mathbf{Q}$ of size 4×2 of the process noise reads

$$\mathbf{Q} = \mathbf{G}.q.\mathbf{G}^T. \tag{3.8}$$

The matrix $\mathbf{G}$ of size 4×2 is written as

$$\mathbf{G} = \begin{bmatrix} 0.5T^2 & T & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5T^2 & T & 1 \end{bmatrix}^T . \tag{3.9}$$

Assuming a white, Gaussian process noise with zero mean and variances $\sigma^2_{a_x}$ and $\sigma^2_{a_y}$ for a_x and a_y respectively, the matrix q is given as

$$q = \begin{bmatrix} \sigma^2_{a_x} & 0 \\ 0 & \sigma^2_{a_y} \end{bmatrix} . \tag{3.10}$$

The initial state vector used to predict the future path of the vehicle is given as

$$\mathbf{x}_0 = [0\ 0\ v_v\ 0\ a_v\ \omega_v v_v] , \tag{3.11}$$

where v_v is speed of the vehicle, $a_v$is acceleration of the vehicle and ω_v is yaw rate of the vehicle.

3.3 Constant Acceleration and Yaw rate Derivatives Model

This is a model used to track the observer's vehicle itself. This might be important, if we think about ego motion compensation, or handing over the information to another target vehicle through vehicle to vehicle communication system. All the analysis is taken from [PTAA07]

The state vector $\mathbf{x}_k$ reads

$$\mathbf{x}_k = [v\ a\ \dot{a}\ \varphi\ \omega\ \dot{\omega}]^T, \tag{3.12}$$

where v is speed of the vehicle, a is the acceleration of the vehicle, $\dot{a}$ is the rate of change of acceleration, ω is the yaw rate and $\dot{\omega}$ is the rate of the yaw rate.

The measurement vector from the CAN bus reads

$$\mathbf{y} = [v\ \omega]^T, \tag{3.13}$$

where v is speed of the vehicle and ω is the yaw rate.

The matrix, $\mathbf{H}$ of size 2×6, which maps the measurement space to the state space is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.14}$$

The covariance matrix $\mathbf{R}$ of size 2×2 of the white and Gaussian measurement noise with a zero mean is

$$\mathbf{R} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix}, \tag{3.15}$$

where σ_v^2 is standard deviation in velocity and σ_ω^2 is standard deviation in yaw rate.

Assuming a constant acceleration and yaw rate derivatives, the transition matrix $\mathbf{A}$ of size 6×6 used for the state equation update is

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0.5T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & T & T & 0.5T^2 \\ 0 & 0 & 0 & 1 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.16}$$

where T is time span.

The covariance matrix $\mathbf{Q}$ of size 6×6 of the process noise is given as

$$\mathbf{Q} = \mathbf{G}.q.\mathbf{G}^T. \tag{3.17}$$

The matrix $\mathbf{G}$ of size 2×6 is written as

$$\mathbf{G} = \begin{bmatrix} 0.5T^2 & T & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5T^2 & T & 1 \end{bmatrix}. \tag{3.18}$$

Assuming a white, Gaussian process noise with zero mean and variances $\sigma_{\dot{a}}^2$ and $\sigma_{\dot{\omega}}^2$ for $\dot{a}$ and $\dot{\omega}$ respectively, the matrix q of size 2×2 is given as

$$q = \begin{bmatrix} \sigma_{\dot{a}}^2 & 0 \\ 0 & \sigma_{\dot{\omega}}^2 \end{bmatrix}, \tag{3.19}$$

where $\sigma_{\dot{a}}^2$ is th variance of the acceleration and $\sigma_{\dot{\omega}}^2$ is the variance of the rate of change of yaw rate.

3.4 Constant Turn Model with Unknown Turn Rate

This model assumes the yaw rate and the speed of the vehicle as constant and the state vector consists of positions and velocities in both direction and its turn rate. Below all the mathematical derivations are given based on [PTAA07].

The state vector $\mathbf{x}_k$ is given as

$$\mathbf{x}_k = [p_x \; p_y \; v_x \; v_y \; \omega]^T, \tag{3.20}$$

where p_x and p_y are the x and y coordinate values respectively and v_x and v_y are the x and y components of the velocity respectively and ω is the yaw rate respectively.

The rate of change of acceleration, $\dot{a}$, is give as

$$\dot{a} = -\omega^2 v, \tag{3.21}$$

where ω is the yaw rate and v is the velocity of the vehicle.

The yaw rate ω is given as

$$\omega = \frac{v\, a}{/v/^2}, \tag{3.22}$$

where ω is the yaw rate, v is the velocity and a is the acceleration of the vehicle.

Decomposing the velocity v along the two axes using the heading angle φ gives us

$$v_x = v \cos(\varphi), \tag{3.23}$$

and

$$v_y = v \sin(\varphi). \tag{3.24}$$

Assuming a CTR, $\omega = \dot{\varphi}$, the prediction equations for position coordinates p_x and p_y read

$$p_{x,k+1} = x_k + \int_{t_k}^{t_{k+1}} v_x(\tau/t_k)\, d\tau = p_{x,k} + T\left[SW.v_{x,k} - CW.v_{y,k}\right], \tag{3.25}$$

and

$$p_{y,k+1} = y_k + \int_{t_k}^{t_{k+1}} v_y\left(\tau/t_k\right) d\tau = p_{y,k} + T\left[CW.v_{x,k} + SW.v_{y,k}\right], \tag{3.26}$$

where

$$SW = \sin\left(\omega T\right)/\omega T, \tag{3.27}$$

and

$$CW = \frac{\left[1 - \cos\left(\omega T\right)\right]}{\omega T}. \tag{3.28}$$

The prediction equation for the velocity components in the two axes is given as

$$v_{x,k+1} = v.\cos\left(\varphi + \omega T\right) = v_x\left(k\right)\cos\left(\omega T\right) - v_y\left(k\right)\sin\left(\omega T\right), \tag{3.29}$$

and

$$v_{y,k+1} = v.\sin\left(\varphi + \omega T\right) = v_x\left(k\right)\sin\left(\omega T\right) + v\left(k\right)\cos\left(\omega T\right), \tag{3.30}$$

In a matrix form, the state equation is given as

$$\begin{bmatrix} p_{x,k+1} \\ p_{y,k+1} \\ v_{x,k+1} \\ v_{y,k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} p_{x,k} + T\left[SW.v_{x,k} - CW.v_{y,k}\right] \\ p_{y,k} + T\left[CW.v_{x,k} + SW.v_{y,k}\right] \\ v_{x,k}\cos\left(\omega T\right) - v_{y,k}\sin\left(\omega T\right) \\ v_{x,k}\sin\left(\omega T\right) - v_{y,k}\cos\left(\omega T\right) \\ \omega\left(k\right) \end{bmatrix}. \tag{3.31}$$

Linearizing the above state equation gives the following transition matrix $\mathbf{A}$ of size 5×5

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega T)}{\omega} & \frac{-(1-\cos(\omega T))}{\omega} & f_{15} \\ 0 & 1 & \frac{1-\cos(\omega T)}{\omega} & \frac{\sin(\omega T)}{\omega} & f_{25} \\ 0 & 0 & \cos(\omega T) & -\sin(\omega T) & f_{35} \\ 0 & 0 & \sin(\omega T) & \cos(\omega T) & f_{45} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.32}$$

where f_{15} through f_{45} read:

$$f_{15} = \frac{\partial p_x}{\partial \omega} = T\left(v_x dsw - v_y dcw\right), \tag{3.33}$$

$$f_{25} = \frac{\partial y}{\partial \omega} = T\left(v_x dcw + v_y dsw\right), \tag{3.34}$$

$$f_{35} = \frac{\partial v_x}{\partial \omega} = -T\left(v_x \sin\left(\omega T\right) + v_y \cos\left(\omega T\right)\right), \tag{3.35}$$

$$f_{45} = \frac{\partial v_y}{\partial \omega} = T\left(v_x \cos\left(\omega T\right) - v_y \sin\left(\omega T\right)\right), \tag{3.36}$$

$$dsw = \frac{\cos\left(\omega T\right)}{\omega} - \frac{\sin\left(\omega T\right)}{\omega^2 T}, \tag{3.37}$$

$$dcw = \frac{\sin\left(\omega T\right)}{\omega} - \frac{1}{\omega^2 T} + \frac{\cos\left(\omega T\right)}{\omega^2 T}. \tag{3.38}$$

The covariance matrix $\mathbf{Q}$ of size 5×5 of the process noise is given as

$$\mathbf{Q} = \mathbf{G}.q.\mathbf{G}^T \tag{3.39}$$

The matrix $\mathbf{G}$ of size 3×5 is written as

$$\mathbf{G} = \begin{bmatrix} 0.5T^2 & 0 & T & 0 & 0 \\ 0 & 0.5T^2 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & T \end{bmatrix}^T \tag{3.40}$$

Assuming a white, Gaussian process noise with zero mean and variances $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ for v_x and v_y respectively, the matrix q of size 3×3 is given as

$$q = \begin{bmatrix} \sigma_{v_x}^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_{\omega}^2 \end{bmatrix}. \tag{3.41}$$

The initial state vector, when the model is used for predicting the trajectory of the host vehicle is given as

$$\mathbf{x}_0 = \left[0\ 0\ v_v\ 0\ \omega_v\right], \tag{3.42}$$

where v_v and ω_v are current velocity and turn rate of the vehicle

When the model is used for target vehicle trajectory prediction, the initial state vector is formed from the current position, velocity and acceleration components of the tracked vehicle and is given as

$$\mathbf{x}_0 = \left[x_0 \; y_0 \; v_x \; v_y \; \frac{v_x a_y - v_y a_x}{v_x^2 + v_y^2} \right]. \tag{3.43}$$

3.5 Constant Turn Model with Known Turn Rate

This model assumes the yaw rate and the speed of the vehicle as constant and the state vector consists of positions and velocities in both direction. The formulas are taken from [Jan05] and [PTAA07].

State vector $\mathbf{x}_k$ of size 4×1 reads

$$\mathbf{x}_k = [p_x \; p_y \; v_x \; v_y]^T, \tag{3.44}$$

where p_x and p_y are the x and y coordinate values respectively and v_x and v_y are the x and y components of the velocity respectively.

The analysis is similar like in that of the Constant Turn Model with Unknown Turn Rate and the linearized transition matrix $\mathbf{A}$ of size 4×4 is given as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega T)}{\omega} & \frac{-(1-\cos(\omega T))}{\omega} \\ 0 & 1 & \frac{1-\cos(\omega T)}{\omega} & \frac{\sin(\omega T)}{\omega} \\ 0 & 0 & cos(\omega T) & -\sin(\omega T) \\ 0 & 0 & \sin(\omega T) & \cos(\omega T) \end{bmatrix}. \tag{3.45}$$

The covariance matrix $\mathbf{Q}$ of the process noise is given as

$$\mathbf{Q} = \mathbf{G}.q.\mathbf{G}^T. \tag{3.46}$$

The matrix $\mathbf{G}$ of size 3×2 is written as

$$\mathbf{G} = \begin{bmatrix} 0.5T^2 & 0 & T & 0 \\ 0 & 0.5T^2 & 0 & T \end{bmatrix}^T. \tag{3.47}$$

Assuming a white, Gaussian process noise with zero mean and variances $\sigma_{v_x}^2$ and $\sigma_{v_y}^2$ for v_x and v_y respectively, the matrix q is given as

$$q = \begin{bmatrix} \sigma_{v_x}^2 & 0 \\ 0 & \sigma_{v_y}^2 \end{bmatrix}. \tag{3.48}$$

3.6 Constant Turn Rate and Constant Tangential Acceleration Model (CTRA)

This models combines the advantages of the CA and the CT models and the following mathematical derivations are taken from [TPA05]

This represents the true motion of vehicle as the model considers the vehicle to move with constant tangential acceleration, a_t and constant yaw rate, ω and the state vector is given as

$$\mathbf{x} = [x\ v_x\ a_{tx}\ y\ v_y\ a_{ty}\ \omega]^T \tag{3.49}$$

The rate of change of speed with time, $\dot{v}$, is constant and is given as the tangential acceleration.

$$a_t = \dot{v} \tag{3.50}$$

The rate of change of the heading angle with time, $\dot{\varphi}$, is constant and is given as the tangential acceleration.

$$\omega = \dot{\varphi} \tag{3.51}$$

The x component of the acceleration, a_x, is given as

$$a_x = \dot{v}_x, \tag{3.52}$$

$$a_x = \dot{v}\cos(\varphi) - v\dot{\varphi}\sin(\varphi), \tag{3.53}$$

$$a_x = a_t\cos(\varphi) - \omega v\sin(\varphi), \tag{3.54}$$

$$a_x = a_t\cos(\varphi) - \omega v_y, \tag{3.55}$$

$$a_x = a_{tx} + a_{nx}, \tag{3.56}$$

where $\dot{v}_x$ is the rate of change of velocity in x direction, $\dot{v}$ is the rate of change of velocity, φ is the heading angle, a_t is the tangential acceleration, ω is the yaw rate v is the velocity, v_y is the y component of the velocity, a_{tx} is the tangential component of the acceleration in x direction and a_{nx} is the normal component of the acceleration in x direction.

The y component of the acceleration, a_y, is given as

$$a_y = \dot{v}_y \tag{3.57}$$

$$a_y = \dot{v}\sin(\varphi) - v\dot{\varphi}\cos(\varphi) \tag{3.58}$$

$$a_y = a_t\sin(\varphi) - \omega v\cos(\varphi) \tag{3.59}$$

$$a_y = a_t\sin(\varphi) - \omega v_x \tag{3.60}$$

$$a_y = a_{ty} + a_{ny} \tag{3.61}$$

The centripital acceleration components, a_{nx} and a_{ny}, are given as

$$a_{nx} = -\omega v_y \tag{3.62}$$

$$a_{ny} = \omega v_x \tag{3.63}$$

Then their resultant, the centripital acceleration, which is normal to vehicle's motion

$$a_n = \sqrt{a_{nx}^2 + a_{ny}^2} = \sqrt{\omega^2\left(v_x^2 + v_y^2\right)} = \omega v \tag{3.64}$$

The tangential components of the acceleration, a_{tx} and a_{ty}, are calculated as

$$a_{tx} = a_t\cos(\varphi), \tag{3.65}$$

$$a_{ty} = a_t\sin(\varphi), \tag{3.66}$$

where φ is the heading angle.

The linearized transition matrix $\mathbf{A}$ of this model is given as

$$\mathbf{A} = \begin{bmatrix} 1 & \frac{\sin(\omega T)}{\omega} & f_{13} & 0 & \frac{\cos(\omega T)-1}{\omega} & f_{16} & f_{17} \\ 0 & \cos(\omega T) & T\cos(\omega T) & 0 & -\sin(\omega T) & -T\sin(\omega T) & f_{27} \\ 0 & 0 & \cos(\omega T) & 0 & 0 & -\sin(\omega T) & f_{37} \\ 0 & \frac{1-\cos(\omega T)}{\omega} & f_{43} & 1 & \frac{\sin(\omega T)}{\omega} & f_{46} & f_{47} \\ 0 & \sin(\omega T) & T\sin(\omega T) & 0 & \cos(\omega T) & T\cos(\omega T) & f_{57} \\ 0 & 0 & \sin(\omega T) & 0 & 0 & \cos(\omega T) & f_{67} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.67}$$

Where f_{13}, f_{43}, f_{16}, f_{46} and f_{17} through f_{67} are calculated with the following equations:

$$c1 = \frac{\sin(\omega T)}{\omega T}, \tag{3.68}$$

$$c2 = \frac{\cos(\omega T) - 1}{\omega T}, \tag{3.69}$$

$$c3 = \cos(\omega T), \tag{3.70}$$

$$c4 = \sin(\omega T), \tag{3.71}$$

$$d1 = \frac{\omega T c3 - c4}{\omega^2}, \tag{3.72}$$

$$d2 = \frac{Tc4 - c2}{\omega}, \tag{3.73}$$

$$d3 = \frac{(\omega T^2 c3 - 2Tc4 - 2c2)}{\omega^2}, \tag{3.74}$$

$$d4 = \frac{\left(\left[2 - (\omega T)^2\right] c4 - 2\omega T c3\right)}{\omega^3}, \tag{3.75}$$

$$d5 = -d2,\ d6 = d1, \tag{3.76}$$

$$d7 = -d4,\ d8 = d3, \tag{3.77}$$

$$f_{13} = \frac{\partial x}{\partial a_{tx}} = \frac{T\,(c2 + c4)}{\omega}, \tag{3.78}$$

$$f_{43} = \frac{\partial y}{\partial a_{tx}} = \frac{(c4 - \omega\,T\,c3)}{\omega^2}, \tag{3.79}$$

$$f_{16} = \frac{\partial x}{\partial a_{ty}} = -f_{(43)}, \tag{3.80}$$

$$f_{46} = \frac{\partial y}{\partial a_{ty}} = f_{(13)}, \tag{3.81}$$

$$f_{17} = \frac{\partial x}{\partial \omega} = v_{x,k}d1 + v_{y,k}d2 + a_{tx,k}d3 + a_{ty,k}d4, \tag{3.82}$$

$$f_{47} = \frac{\partial y}{\partial \omega} = v_{x,k}d5 + v_{y,k}d6 + a_{tx,k}d7 + a_{ty,k}d8, \tag{3.83}$$

$$f_{27} = \frac{\partial v_x}{\partial \omega} = -T(v_{x,k+1}c4 + v_{y,k+1}c3), \tag{3.84}$$

$$f_{57} = \frac{\partial v_y}{\partial \omega} = T(v_{x,k+1}c3 - v_{y,k+1}c4), \tag{3.85}$$

$$f_{37} = \frac{\partial a_{tx}}{\partial \omega} = -T(a_{tx,k}c4 - a_{ty,k}c3), \tag{3.86}$$

$$f_{67} = \frac{\partial a_{ty}}{\partial \omega} = -T(a_{tx,k}c3 - a_{ty,k}c4). \tag{3.87}$$

3.7 Adaptive Dynamic (AD) Model

In [PTAA07], the AD Model is discussed briefly and the following part is based on that.

The CA model is better in predicting the trajectory of a vehicle, which is moving with a constant longitudinal acceleration. On the other hand, the CT model suits to a vehicle

turning with a constant turn rate, whereas the CTRA model predicts the trajectory, if the vehicle is turning with a constant yaw rate, while it is accelerating at the same time. Heuristic rules can be used for applying the right model to the right scenario based on a data from real driving recordings. The tracking model is adapted to the derivative of the yaw rate and the longitudinal acceleration, resulting in optimized performance in the lateral and longitudinal axes respectively, as this makes it possible to apply the right model to the right scenario.

3.8 Common State Space Model

The following model is taken from [BY07]. The model considers the longitudinal and the angular motion of the vehicle being tracked separately. The model is linear, which makes it quite simple for implementation.

The state space model $\mathbf{x}_{k+1}$ of size 5×1 reads

$$\mathbf{x}_{k+1} = \begin{bmatrix} d_{k+1} \\ v_{ego,k+1} \\ v_{obj,k+1} \\ a_{ego,k+1} \\ a_{obj,k+1} \end{bmatrix} = \mathbf{A}\mathbf{x}_k + \mathbf{B}w_k, \tag{3.88}$$

where d_k is the distance between the observer and the object, $v_{ego,k+1}$ is the speed of the observer, $v_{obj,k+1}$ is the speed of the object, $a_{ego,k+1}$ is the acceleration of the observer and $a_{obj,k+1}$ is the acceleration of the observer.

The transition matrix $\mathbf{A}$ of size 4×5 is given as

$$\mathbf{A} = \begin{bmatrix} 1 & -T & T & -\frac{T}{2}^2 & \frac{T}{2}^2 \\ 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.89}$$

where T is the cycle time.

The matrix $\mathbf{B}$ of size 5×2 add noise elements of $w(k)$ to the last two state variables and is given as

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T. \tag{3.90}$$

The corresponding measurement equation $\mathbf{y}(k)$ of the longitudinal motion is given as

$$\mathbf{y}(k) = \begin{bmatrix} d_k^m \\ v_{rel,k}^m \\ v_{ego,k}^m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} x_k + v_k, \tag{3.91}$$

where d_k^m measured relative distance between the observer and the object, $v_{rel,k}^m$ measured relative speed of the observer with respect to the object and $v_{ego,k}^m$ the measured speed of the ego vehicle.

We assume that the vehicles are driving in the same direction though in reality it depends also on their driving directions, but it can still be used with the Kalman filter. The relative speed, $v_{rel,k+1}$, is calculated as

$$v_{rel,k+1} = v_{obj,k} - v_{ego,k}, \tag{3.92}$$

where $v_{ego,k}$ is the speed of the observer and $v_{obj,k}$ is the speed of the object.

The state transition equations for the angular dynamics model is given as

$$\mathbf{x}_{a,k+1} = \begin{bmatrix} \dot{\alpha}_{k+1} \\ \alpha_{k+1} \end{bmatrix} = x_{a,k} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_{a,k}, \tag{3.93}$$

where $\dot{\alpha}_k$ the relative angle from the vehicle (or sensor) axis to the object α_k the angular velocity or gyro and $w_{a,k}$ noise process which models changes in the accelerations of observer and object.

The corresponding measurement equation of the angular motion is

$$\mathbf{y}_a = \alpha_k^m = \begin{bmatrix} 1 & 0 \end{bmatrix} x_{a,k} + v_{a,k}, \tag{3.94}$$

where

α_m^m relative angle of the observer to the observed object and

$v_{a,k}$ angular measurement noise vector.

3.9 Global Tracking Model

The following part is taken from [BY07]. This is a tracking model devised to deal with some of the draw backs of other tracking models like the Common State Space Model. The drawbacks are the inabilities of the tracking model to fulfill the needs listed below.

The model addresses successfully the following needs

1. Separation of the two distinct types of movement changes (acceleration/deceleration and steering).
2. Representation of constant object motion with a constant state.
3. Correct information consideration of both the ego and target vehicles.
4. All the sensors have the same coordinate system.

The state vector, $\mathbf{x}_{k+1}$ reads

$$\mathbf{x}_{k+1} = \begin{bmatrix} s_{x,k+1} \\ s_{y,k+1} \\ v_{k+1} \\ \varphi_{k+1} \\ a_{k+1} \\ \delta_{k+1} \end{bmatrix} = f\left(\mathbf{x}_k\right) + B\mathbf{w}_k, \tag{3.95}$$

where $(s_{x,k}, s_{y,k})$ is the position of the vehicle, v_k is the tangential speed of the vehicle, φ_{k+1} is the heading angle, a_{k+1} is the acceleration of the vehicle and δ_k is the steering angle and w_k is the input process noise, which is a 2x1 vector that models changes in acceleration and steering angle.

Fig 3.1 shows the state and measurement variables of Global Tracking Model. All the variables are given in Global Coordinate System.

As shown below the derivation of the state transition function, $f\left(\mathbf{x}\left(k\right)\right)$ of equation (3.95) is solved below. The index c in this section indicates the variables, which have a continuous-time representation

The position variables, $s_{x,k+1}$and $s_{y,k+1}$ at time $t = (k+1)\,T$ are calculated as.

$$\begin{bmatrix} s_{x,k+1} \\ s_{y,k+1} \end{bmatrix} = \begin{bmatrix} s_{x,k} \\ s_{y,k} \end{bmatrix} + \int_{kT}^{(k+1)} v_c(t) \begin{bmatrix} \cos(\varphi_c(t)) \\ \sin(\varphi_c(t)) \end{bmatrix}, \tag{3.96}$$

For calculation convenience, speed and driving directions are assumed as constant during every interval of length T and the values, which we would have obtained at the center of the interval are taken and it is proven with test runs that this assumption do not cause any noticeable errors.

The time at the center of the interval between discrete time step k and $k+1$, t, is given as

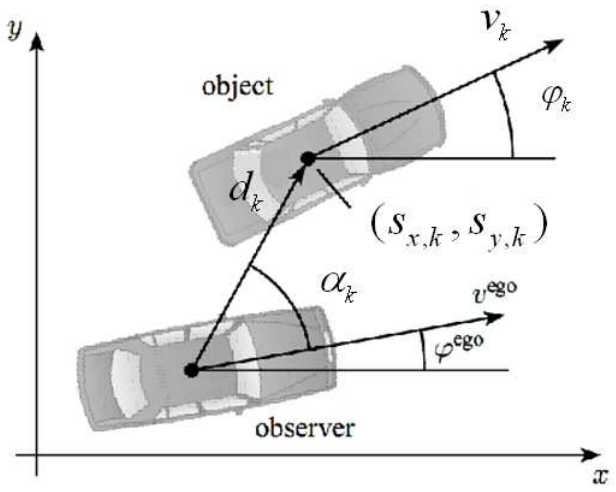


Figure 3.1: State and Measurement Variables of Global Tracking Model [BY07]

$$t = (k + \frac{1}{2})T \tag{3.97}$$

The resulting speed, $v_c(t)$ and driving direction, $\varphi_c(t)$ are given as

$$v_c(t) = \bar{v}_k = v_k + \frac{T}{2}a_k, \tag{3.98}$$

$$\varphi_c(t) = \bar{\varphi}_k + \frac{T}{2L}\delta_k v_k, \tag{3.99}$$

where T is time span, v_k is velocity at discrete time step k, a_k is acceleration at discrete time step k, $\bar{\varphi}_k$ is heading angle at discrete time step k, L is length of wheel base and δ_k is steering angle at discrete time step k.

The resulting simplified transition equations are

$$\begin{bmatrix} s_{x,k+1} \\ s_{y,k+1} \end{bmatrix} = \begin{bmatrix} s_{x,k} \\ s_{y,k} \end{bmatrix} + T\bar{v}_k \begin{bmatrix} \cos(\varphi_c(t)) \\ \sin(\varphi_c(t)) \end{bmatrix} \tag{3.100}$$

The transition equations, which consist of the speed, acceleration and steering angle are linear and they are given as follows

$$\begin{bmatrix} v_{k+1} \\ a_{k+1} \\ \delta_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ a_k \\ \delta_k \end{bmatrix} \tag{3.101}$$

The state transition equation for the driving direction, φ_{k+1}, which is a two-point single track model relating changes in driving directions to the speed, is given as

$$\varphi_{k+1} = \varphi_k + \frac{T}{L}\delta_k v_k + w_{2,k}, \tag{3.102}$$

where T is time span, v_k is velocity at discrete time step k,φ_k is heading angle at discrete time step k, L is length of wheel base, δ_k is steering angle at discrete time step k and $w_{2,k}$ is the noise.

The measurement vector, $\mathbf{y}(k)$, consisting of distance, d_k^m, relative angle, $\alpha_{rel,k}^m$ and the Doppler speed, $v_{rel,k}^m$, measured by the sensors is given as

$$\mathbf{y}(k) = \begin{bmatrix} d_k^m \\ \alpha_{rel,k}^m \\ v_{ego,k}^m \end{bmatrix} = g(\mathbf{x}_k) + v_k, \tag{3.103}$$

where $v\,(k)$ is a 3x1 vector, which models the measurement noise in the three directions

As shown below the derivation of the state transition function, $g\,(\mathbf{x}_k)$ is given.

The relative angle,α_k , the relative distance between the two vehicles, d_k and the Doppler speed, $v_{rel,k}$ are given as

$$\alpha_k = \arctan\left(\frac{s_{y,k} - s_y^{sen}}{s_{x,k} - s_x^{sen}}\right) - \varphi^{ego} - \theta^{sen}, \tag{3.104}$$

$$d_k = \sqrt{(s_{x,k} - s_x^{sen})^2 + (s_{y,k} - s_y^{sen})^2}, \tag{3.105}$$

$$v_{rel,k} = v^{ego}\cos(\alpha_k + \theta^{sen}) - v_k\cos(\alpha_k + \theta^{sen} + \varphi^{ego} - \varphi_k), \tag{3.106}$$

where $\left(s_x^{sen}, s_y^{sen}\right)$ is the current sensor position in the global coordinate system, φ^{ego} is the current driving direction of the observer, θ^{sen} is the sight direction of the sensor measured relative the longitudinal axis of the ego vehicle, $(s_{x,k}, s_{y,k})$ is position coordinate and v_k is tangential speed.

3.10 Ego-Motion Estimation Using Inertial Sensor (INS) data 1

The following model and discussion is taken [FDL03]. The following model is a simplified model compared to the single track model as it does not require many parameters, but it is proven that its accuracy is comparable [FDL03].In this model the pitch, roll and motion of the vehicle in the z direction are neglected. The motion of the vehicle in the x and y directions and its yaw motion are calculated with this model. As input, the steering angle and the speed of the vehicle are used, which are obtained from the INS data. The time T is the time between two consecutive scans in the case of the laser scanner, which has normally a typical value of 100ms . The slip angle is also calculated using the INS data. In this model condition of no change in wheel load, no cross acceleration, no accelerating or decelerating forces is assumed. The aerodynamic forces are also ignored. In this simplified model the side slip angle, β, depends only on the steering angle δ.

The relative positions Δx, Δy and $\Delta\psi$ between two time steps are given as

$$\Delta x = v.T.\cos(\beta), \tag{3.107}$$

$$\Delta y = v.T.\sin(\beta), \tag{3.108}$$

$$\Delta\psi = \frac{v.T}{l_h}.\beta \tag{3.109}$$

and

$$\beta = \frac{l_h}{l_v + l_h}.\delta, \tag{3.110}$$

where Δx is translation in x direction between two consecutive scans,Δy is translation in y direction between two consecutive scans, $\Delta\psi$ is yaw rotaion between two consecutive yaw scans,v is velocity of ego vehicle, T is sampling rate,β is body slip angle, l_h is distance from the center of gravity to the rear axis of the car, l_v is distance from the center of gravity to the front axis of the car andδ is front wheel steering angle.

3.11 Ego-Motion Estimation Using Inertial Sensor (INS) data 2

The following model and discussion is taken also from [FDL03]. The following model is a model, which have many parameters like both the front and the rear slip angles, α_v, α_h ,

cornering stiffness of both the front and the rear tires, C_v and C_h, mass of the vehicle, m and yaw moment of inertia of the vehicle in z-direction, J_z [FDL03].

The state vector for this model contains four variables and reads:

$$\mathbf{x} = (\beta\ \dot{\psi}\ \delta\ \dot{\delta})^T, \tag{3.111}$$

where β is the side slip angle, $\dot{\psi}$ is yaw rate, δ is the steering angle and $\dot{\delta}$ is the rate of change of the steering angle.

The state equation is then written as

$$\dot{\mathbf{x}} = \mathbf{A}.\mathbf{x}. \tag{3.112}$$

where $\mathbf{A}$ is the transition matrix and is given as

$$\mathbf{A} = \begin{bmatrix} -\frac{C_v+C_h}{m.v} & \frac{C_h.l_h-C_v.l_v-m.v^2}{m.v^2} & \frac{C_v}{m.v} & 0 \\ \frac{C_h.l_h-C_v.l_v}{J_z} & -\frac{C_h.l_h^2-C_v.l_v^2}{J_z.v} & \frac{C_v.l_v}{J_z} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.113}$$

where C_v and C_h are cornering stiffness of both the front and the rear tires, m is mass of the vehicle , J_z is the yaw moment of inertia of the vehicle in z-direction, v is velocity of the vehicle, l_v is the distnace between the front wheels and the center of gravity of the vehicle and l_h is the distnace between the rear wheels and the center of gravity of the vehicle.

This state equation can be used to find the state of a vehicle with the use of the Kalman filter.

The second model was used to evaluate the first simple model as it is presented in [FDL03]. Out of the different scenarios used for this evaluation purpose, the Authors of the mentioned work gave one as an example. The trajectories, which are produced, with both models are almost the same, only with a small error.

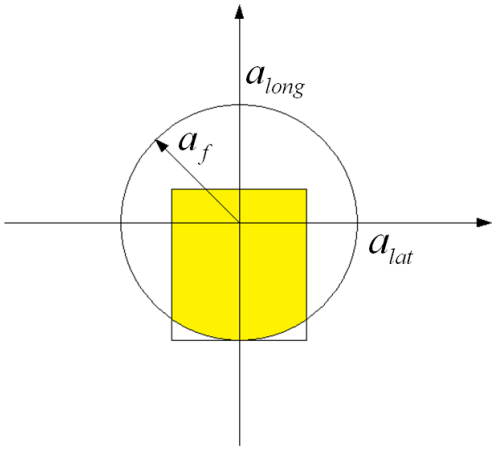


Figure 3.2: Friction elipse [DD07]

3.12 Simplified Dynamic Model of a Vehicle

The following model is taken from [DD07] and [Eid08].

For the following description, see Fig 3.2. In this model the vehicle is assumed to be controlled with the two control input values (u_1, u_2), where the first input is for the longitudinal motion and the second is for the lateral motion.The allowed acceleration on the friction ellipse is found out by intersecting the friction ellipse itself with a rectangle having maximum engine torque and maximum steering angle as sides. A separate treatment of the longitudinal and lateral acceleration is used and the limitation is based on the vehicle speed v and the switching threshold speed values in the longitudinal and lateral directions, v_{long} and v_{lat}.

If $v \leqslant v_{lat}$, the maximum steering angle φ_{max} and the wheel base L keep the lateral acceleration in a limited interval, and if $v > v_{lat}$ only the road friction a_f limits it. On the other hand, if $v > v_{long}$, engine torque $\frac{k}{v}$ (k is engine power divided by the mass of the vehicle) and road friction are the two limiting factors, which limit the longitudinal acceleration, whereas if $v \leq v_{long}$ the limiting factor is the road friction a_f. According to Eidehall's argument, it is the road friction, which limits the vehicle's braking acceleration rather than the engine torque.

As per the experimental investigations forwarded in [Eid08] , the expected mean longitudinal and lateral accelerations $(\bar{a}_{long}, \bar{a}_{lat})$ are zero, as calculated and confirmed from highway drive data.

The equations are given as

Symbol $\dot{x}$ denotes the x componenet of the velocity and reads:

$$\dot{x} = v \cos \theta, \tag{3.114}$$

where v is the velocity and θ is the heading angle.

Symbol $\dot{y}$ denotes the y componenet of the velocity and reads:

$$\dot{y} = v \sin \theta, \tag{3.115}$$

where v is the velocity and θ is the heading angle.

Symbol $\dot{v}$ denotes acceleration and reads:

$$\dot{v} = \{ \begin{array}{c} u_1 \frac{k}{v} \, if \, v > v_{long} \, and \, u_1 \geq 0 \\ u_1 a_f \, else \end{array} , \tag{3.116}$$

where u_1 is longitudinal motion control input value, $\frac{k}{v}$ is engine power divided by the mass of the vehicle, a_f is the maximum acceleration on the friction ellipse and v_{long} is the limiting longitudinal velocity.

Symbol $\dot{\theta}$ denotes the rate of change of heading angle and reads:

$$\dot{\theta} = \{ \begin{array}{c} v \sin \left(\left(\varphi_{max} u_2 \right) / L \right) \, if \, v \leq v_{lat} \\ u_2 a_f / v \, if \, v > v_{lat} \end{array} , \tag{3.117}$$

where δ_{max} is the maximum steering angle, u_2 is lateral motion control input value, L width of the wheel base of the vehicle, a_f is the maximum acceleration on the friction ellipse and v_{lat} is the limiting lateral velocity.

3.13 Simplified bicycle type vehicle model

The following part is taken from [GCT08].

The following model is a single track model and is usually called as the bicycle model, which takes into account the behavioral and stability issues listed below, thereby integrating constraints related to vehicle kinematics. The name bicycle is an arbitrarly given name and has no relation with the kinematics of a bicyle .

- No transfer of lateral tire load: the vehicle is assumed as if it is moving only on a single track
- No longitudinal transfer of tire loads, vehicles center of gravity is zero.

- No roll or pitch motion
- Tires are assumed to be a linear configuration
- Constant forward speed,v
- No aerodynamic effects
- Position control
- No effect from suspension and chassis flexibility

The implication of all the above assumptions is that the the acceleration is bound to remain below 0.4g, the steering and drift angles are small and no suspension displacement is assumed (as the road is assumed to be smooth). When the vehicle turns, it turns at low velocity resulting in negligible centrifugal forces and no development of tyre lateral forces, which means the tyre rolls without sliding or drifting and the instantaneous center of rotation of each tyre lies in the middle of the curve.

Fig 3.3 shows how this model models a four wheel vehicle as a single track. The symbols L and δ represent the distance between the front and rear wheels and the steering angle respectively.

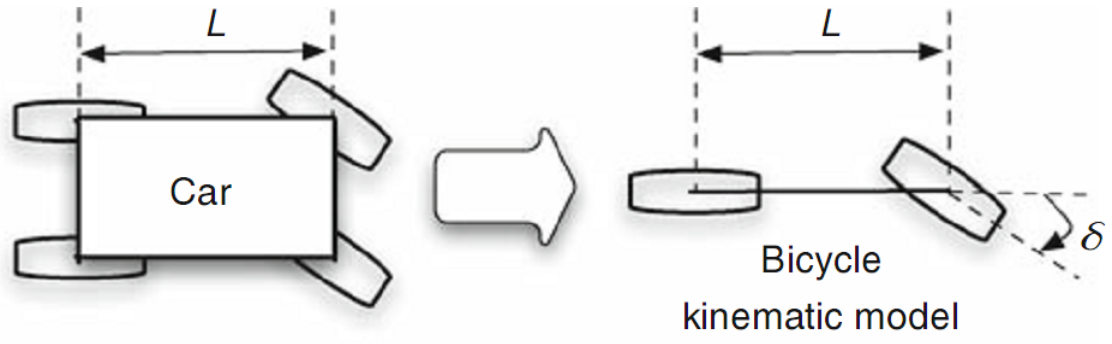


Figure 3.3: The "bicycle model" [GCT08]

Let (x,y) be the vehicle fixed coordinate system for this model, then its kinematic relations are given as follows.

Symbol $\dot{x}$ and $\dot{y}$ denote the x and y componenets of the velocity and read:

$$\dot{x} = v.\cos\,\beta, \tag{3.118}$$

$$\dot{y} = v.\sin\,\beta \tag{3.119}$$

and the rate of change of the slip angle $\dot{\beta}$ read:

$$\dot{\beta} = \frac{v}{L} \tan \delta, \tag{3.120}$$

whereβ is the vehicle's side slip, δ is the steering angle, v is the velocity of the vehicle and L is distance between the front and rear wheels.

4 Collision Prediction Models

In this chapter some representative models used for prediction of collision are discussed and the associated mathematical derivations are also shown in detail. Among them the 3rd, 4th, 6th and 7th models are implemented directly or with some needed modifications. Some similarity can be drawn among some of the models.

4.1 Global Approach

This is a model recently published in [SA09] and it incorporates several factors in it as discussed below

This approach has three sequential steps for mitigating a collision;

1. Environment Modelling
2. Trajectory Prediction-with state estimators like Kalman filter
3. Collision Detection-with the use of TTC and intersecting circles concept
4. Risk Management-with the use of belt pretentioners, automated brake systems, pre-firing of airbags and so on.

The whole process looks as illustarted in Fig 4.1. The environment is assessed with proproceptive and extroceptive sensors and the signal from them is fused in the data fusion model. The current environment state model makes evaluation of the static enviroment, dynamic objects and ego vehicle state using the the sensor data obtained from the data fusion mdel. Then the trajectories are checked, if they possibly result in collisison or not. If the situation is found to be dangerous the necessary actuators are activated.

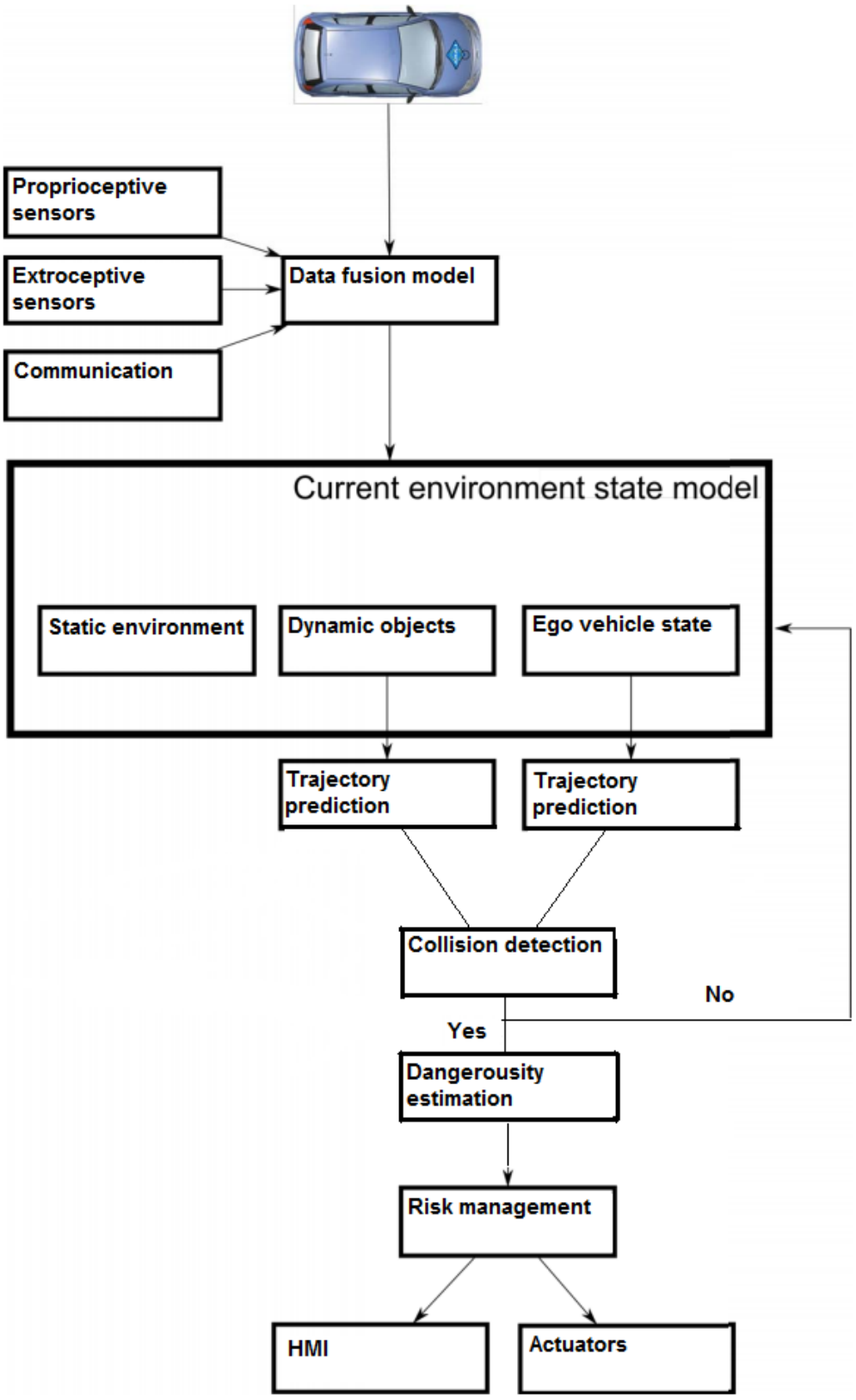


Figure 4.1: A block diagram summarizing the risk assessment steps [BY07]

The uncertainty of Gravitational Position System (GPS) based location of a vehicle is modeled with a Gaussian ellipse, which gives us the vehicle's position probability. The collision between two vehicles results in intersection of their corresponding ellipses. Calculation of the intersection of ellipses is unfortunately time consuming, rather it is much easier to perform intersection check up of series of circles inscribed in the ellipses as shown in Fig. 4.2. Collision estimation is therefore based on calculation of respective distances between the respective circles of the two vehicles. A collision is detected, if at least one circle of one vehicle intersects with one or more circles of the other vehicle.

The algorithm consists of the following steps:

1. TTC calculation-tells the time available for the drivers for taking action for collision avoidance or mitigation.
2. Number of circles intersecting-tells about the extent of the crash.
3. Duration of the circles intersection-the longer it takes,the higher the crash probability.

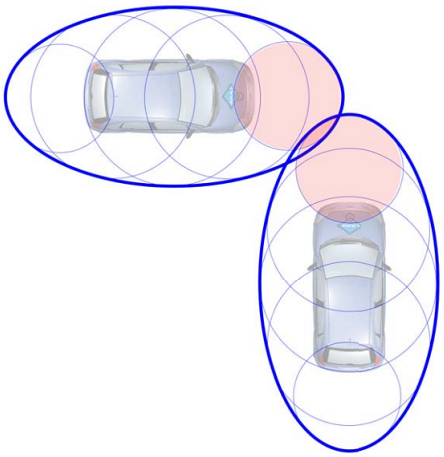


Figure 4.2: The collision detection is performed by detecting the collision between the circles around each vehicle [BY07].

4.2 Extended Area of Crash

The following model is taken from [MRR06] and the reader is encouraged to refer the mentioned literature for more details.

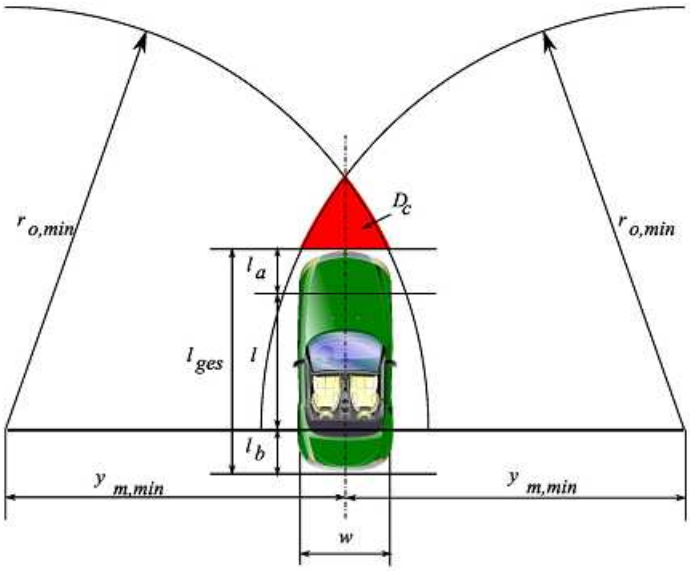
The maximum turning maneuver of a vehicle depends on the current speed v and the grip potential μ_r between the tires and the road. $\mu_r = 1$ is taken as most of the times

the real value is unknown. The two maximum turning maneuvers, one to the right and one to the left enclose, the crash area D_c.If an obstacle is in this area, it will definitely be hit by the vehicle. The two maximum turning maneuvers have a radius of $r_{o,min}$.

The x coordinate of no return point of an obstacle is calculated as

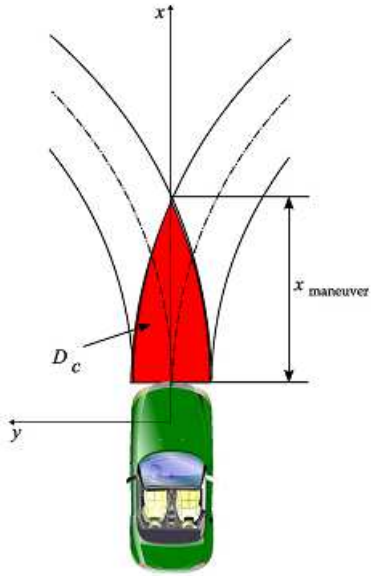
$$x_{crash}(y) = \sqrt{r_{o,min}^2 - (y_{m,min} + y)^2} - (l + l_a)\,, \tag{4.1}$$

where y is the y coordinate of the position of the obstacle, $r_{o,min}$ the outer turning radius of the maximum maneuver, $y_{m,min}$ distance between the center of the maneuver arc and the y axis, l distance between front and rear axle and l_a distance between the front axle and the front bumper.



(a) Geometric view

Figure 4.3: Crash area model [MRR06]



(b) Crash area

Figure 4.4: x coordinate of no return point of an obstacle [MRR06]

The time needed for the driver, to lift his/her foot from the throttle pedal and place it on the brake pedal is most of the times around 150 ms. This reaction time is modeled as dead time. Then the corresponding distance resulting is calculated as

$$x_{reaction} = v.t_{reaction} = v * 0.15s. \tag{4.2}$$

This reaction time and the corresponding area enlarges the crash area as shown in Fig 4.5. D_c now consists of the maneuver and the reaction areas. To calculate this area we need to do prediction of the path, which depends on the current speed and steering angles of the vehicles. For pre-crash cases, which involve very short time in the range of a few hundred milliseconds, linear path assumption is sufficient.

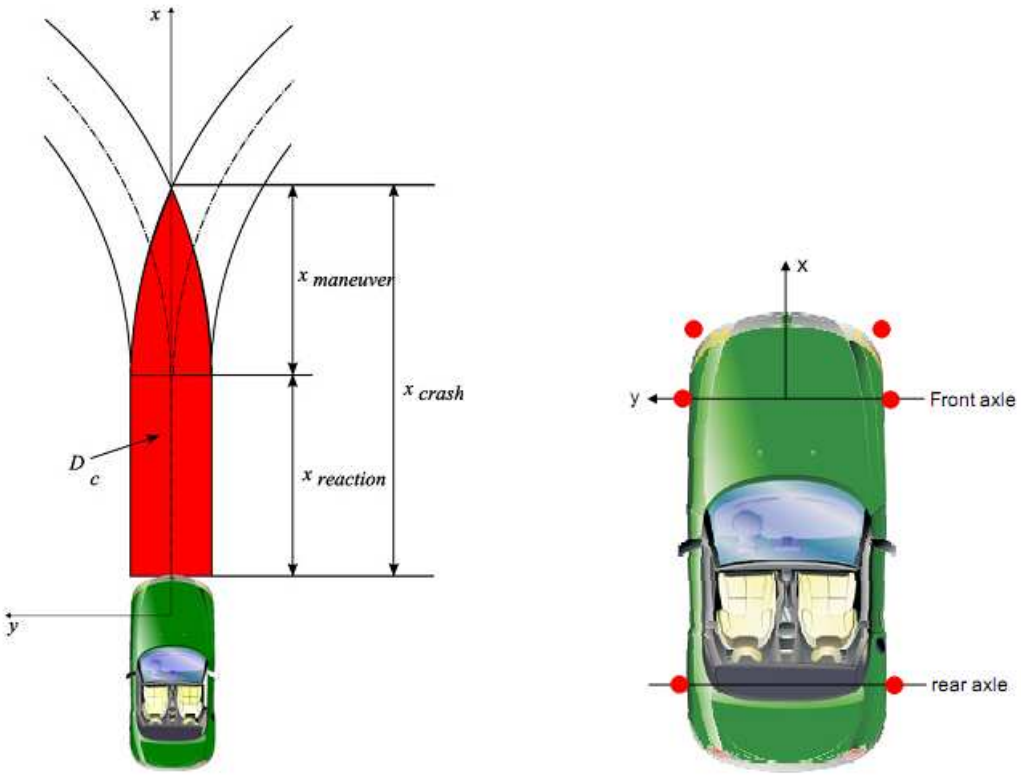
When the vehicle is moving along a curve its rear is closer to the center of the curve than its front. To find the resulting curved crash area, we need to do plotting of some selected representative boundary points of the vehicle as shown in Fig 4.5.

Three coordinate systems are used for detecting a vehicle, whether it is lying in the maneuver area or reaction area or in none of them.

1. S_{car} known as the car coordinate system fixed at the front of the car

2. S_{ra} known as reaction area coordinate system. It is fixed to the center of the curve.
3. S_{ma} known as the maneuver area coordinate system, which is fixed at the end of the reaction time

Checking whether a target is lying inside or outside of the crash area, involves the transformation of the targets vehicle into the other coordinate systems. Comparing the target's radial position obtained after transformation into the reaction area coordinate system, with the boundaries of the reaction area $r_{rear,inner}$ and $r_{bumper,outer}$ is used to detect, whether a target is inside the reaction area. To detect whether a vehicle is inside the maneuver area, we translate and rotate its coordinates from the vehicle coordinate system into the maneuver area coordinate system, then finally we do check like we did previously.



(a) Extended crash area in straight direction

(b) Relevant points of the car

Figure 4.5: Crash area model incorporting driver reaction time [MRR06]

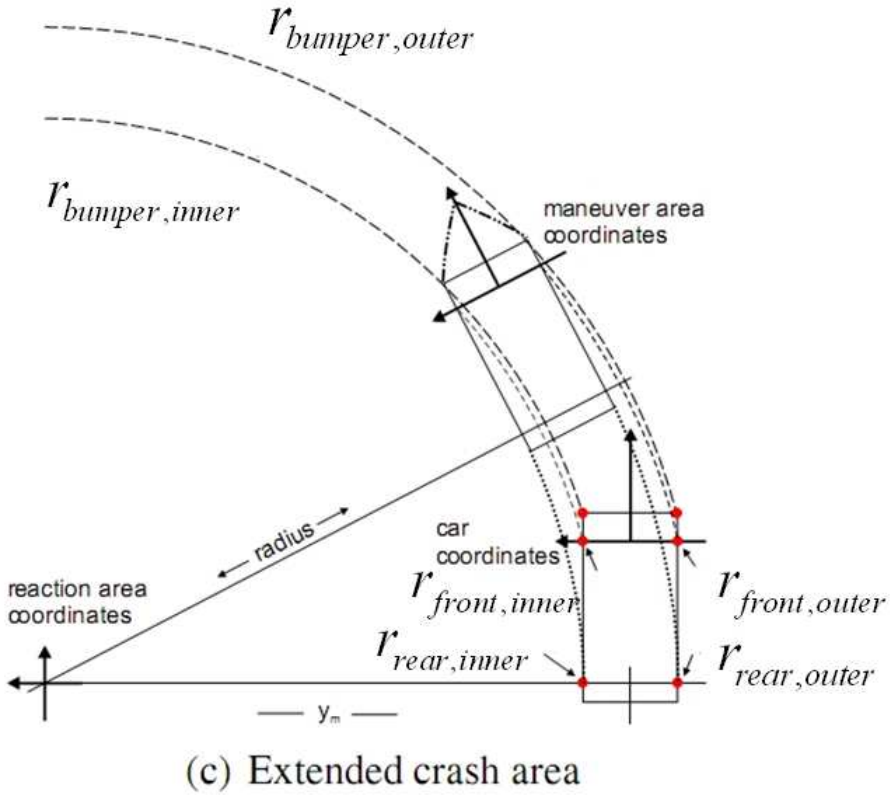


Figure 4.6: Crash area model incorporting driver reaction time along a curve [MRR06]

4.3 Projection Overlap based

This model is taken from [Köh04] and some geometrical derivations are added to make the general concept applicable. This algorithm is very fast and makes an accurate accident situation assessment.

Fig 4.10 is explained here. The first step in this method is to get the relative velocity direction and to define a projection line perpendicular to the relative velocity. Then all the corners of the two rectangular representations of the vehicles are projected on this line. The projection of each rectangle gives us a line. That means two lines from both of the rectangles, which lie on the same line. The intersection of the two lines is the overlap. To get this intersection, the method, which should be used, is to check, if one of the end points of one projection lies between the two end points of the other projected line. Thenafter it is possible to get the end points of the overlap line and to calculate this length. Based on the length and the TTC, it is possible to calculate the collision

probability.

The block diagram in Fig 4.7 shows the algorithm of collision prediction with the use of overlap calculation. The object contour data is obtained from the vehicle detection and tracking done by the sensors on all the objects viewed.

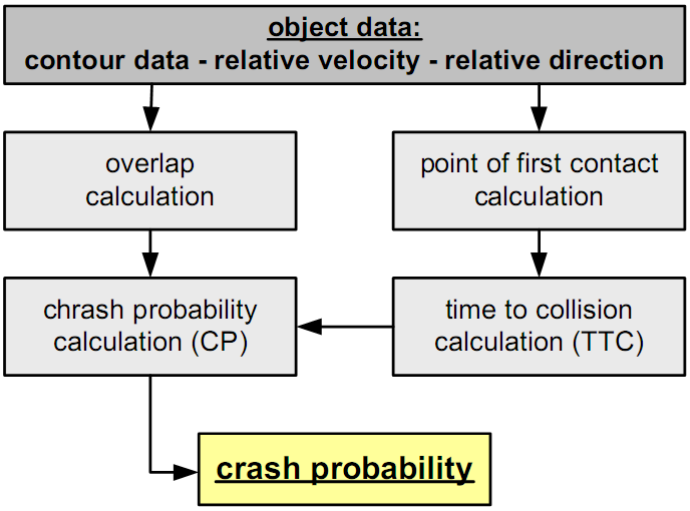


Figure 4.7: Crash Detection Algorithm [Köh04]

4.3.1 Overlap Calculation

Fig 4.6 shows the illustration of the original concept of overlap calculation. In the orignal concept the ovrlap is the smaller one of O_L and O_R. The overlap calculation involves calculating the relative velocity and its direction, finding the size of the projected size of the vehicles on a line perpendicular to the relative velocity and also the center to center distance of the two rectangles representing the vehicles and the last step is to calculate the overlap using all the necessary angles and dimensions.

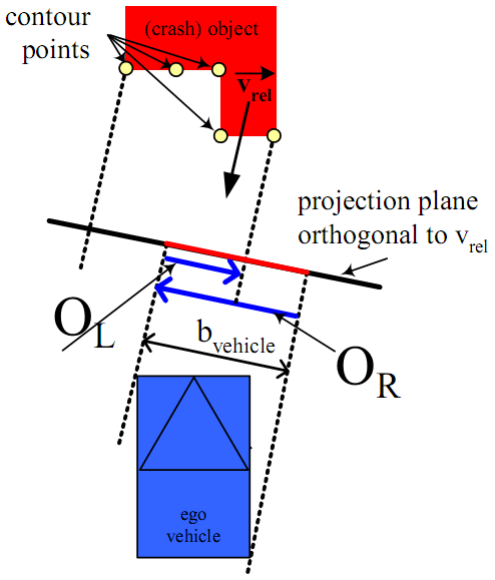


Figure 4.8: Overlap Calculation [Köh04]

4.3.2 Normalized Overlap Calculation

After getting the size of the projected overlap, ov, the next step would be to calculate the ratio of the overlap and the projected size of the vehicle. Here the size of the projected ego vehicle is enlarged by a safety factor of $\Delta p_e/2$, that means by a total value of Δp_e and this in turn leads to the enlargement of the overlap size, ov, by a safety factor of Δp_e.

The normalized overlap, ov', reads

$$ov' = \frac{ov + \Delta p_e/2}{p_e + \Delta p_e}. \tag{4.3}$$

4.3.3 Point of First Contact (PFC) and Time to Collision Calculation(TTC)

The point of the target vehicle (object) contour, which has the shortest distance from ego vehicle's contour is named as the point of first contact (PFC). We can also take an assumption that PFC is equivalent to the smallest distance between the two rectangular boundaries of the two vehicles. The point should be a member of the points, which form the overlap on the line perpendicular to the direction of the relative velocity. From PFC and v_{rel}, we can calculate the TTC value.

The TTC is calculated as,

$$TTC = \frac{d_{PFC}}{v_{rel}}, \tag{4.4}$$

where d_{PFC} is distance of first point of contact and v_{rel} is the relative velocity.

Here in this diagram an illustration is given. The variable d_{pfc} can be calculated with the algorithm given in [JAC08].

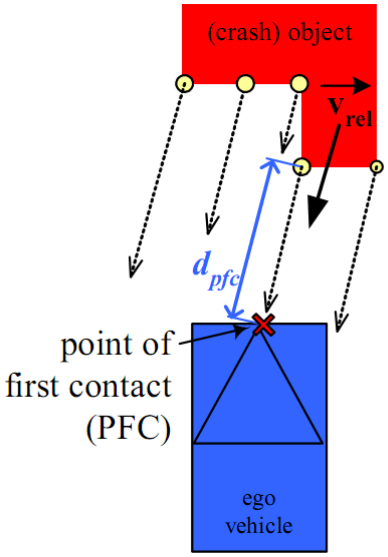


Figure 4.9: Calculation of PFC and TTC [Köh04]

The x component of the relative velocity, $v_{rel,x}$, is calculated as

$$v_{rel,x} = (v_{tx} - v_{ex}), \tag{4.5}$$

where v_{tx} and v_{ex} are respectively the target and ego vehicles' x component.

The y component of the relative velocity, $v_{rel,y}$, is calculated as

$$v_{rel,y} = (v_{ty} - v_{ey}), \tag{4.6}$$

where v_{ty} and v_{ey} are respectively the target and ego vehicles' y components.

If we know the slip angles of both vehicles, we can calculate the relative velocity components, $v_{rel,x}$ and $v_{rel,y}$, as

$$v_{rel,x} = v_t \cos \beta_t - v_e \cos \beta_e, \tag{4.7}$$

$$v_{rel,y} = v_t \sin \beta_t - v_e \sin \beta_e, \tag{4.8}$$

where v_t and v_e are the target and ego vehicles' velocities and β_t and β_e are the target and ego vehicles' side slip angles.

The relative velocity, v_{rel}, can be calculated from the above components as

$$v_{rel} = \sqrt{(v_{rel,x})^2 + (v_{rel,y})^2}, \tag{4.9}$$

$v_{rel,x}$ and $v_{rel,y}$ are respectively the x and y components of the velocities.

4.3.4 Crash Likeness(CL)

with the following metrics the likeness of crash, CL, can be calculated as

$$CL = \frac{ov'}{TTC - \Delta t}, \tag{4.10}$$

where Δt is delay time, which might be vehicle specific and is determined experimentally, ov' is the nomalized overlap and TTC is the time to collision.

4.3.5 Geometrical Analysis of Overlap Calculation

This part is a further contribution to the orginal work and it is derived with the thought of making it applicable. Fig 4.10 shows us all the important dimensions and angles used to calculate the overlap of the two vehicles.

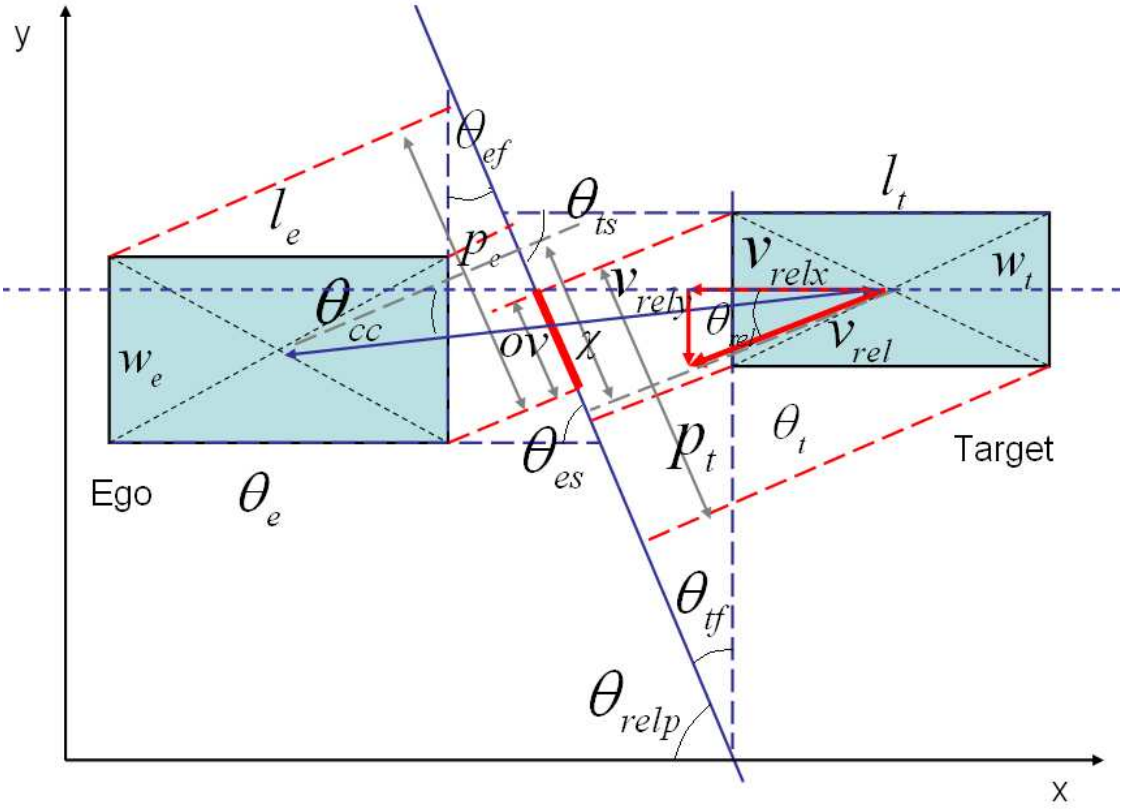


Figure 4.10: Overlap calculation geometry

The angle that the relative velocity makes with the horizontal axis, θ_{rel}, reads:

$$\theta_{rel} = \arctan\left(\frac{v_{rel,y}}{v_{rel,x}}\right), \tag{4.11}$$

where $v_{rel,y}$ and $v_{rel,x}$ are the y and x components of the relative velocity respectively.

The angle of the line perpendicular to the relative velocity, θ_{relp}, reads:

$$\theta_{relp} = \theta_{rel} + \frac{\pi}{2}, \tag{4.12}$$

where θ_{rel} is the angle that the relative velocity makes with the horizontal axis.

The angle between the side of the ego vehicle and perpendicular line, θ_{es}, reads:

$$\theta_{es} = |\theta_{relp} - \theta_e|, \tag{4.13}$$

where θ_{relp} is the angle of the line perpendicular to the relative velocity andθ_e is the heading angle of the ego vehicle.

The angle between the front of the ego vehicle and perpendicular line, θ_{ef}, reads:

$$\theta_{ef} = |\theta_{relp} - \theta_e| + \frac{\pi}{2}, \tag{4.14}$$

where θ_{relp} is the angle of the line perpendicular to the relative velocity andθ_e is the heading angle of the ego vehicle.

The angle between the side of the ego vehicle and perpendicular line, θ_{ts}, reads:

$$\theta_{ts} = |\theta_{relp} - \theta_t|, \tag{4.15}$$

where θ_{relp} is the angle of the line perpendicular to the relative velocity andθ_t is the heading angle of the ego vehicle.

The angle between the front of the ego vehicle and perpendicular line, θ_{tf}, reads:

$$\theta_{tf} = |\theta_{relp} - \theta_t| + \frac{\pi}{2}, \tag{4.16}$$

where θ_{relp} is the angle of the line perpendicular to the relative velocity andθ_t is the heading angle of the ego vehicle.

The ego vehicle projection on the line perpendicular to the relative velocity, p_e, reads:

$$p_e = |l_e \cos\, \theta_{es}| + |w_e \sin\, \theta_{ef}|, \tag{4.17}$$

where l_e is length of the ego vehicle, w_e is width of the ego vehicle, θ_{es} is the angle between the side of the ego vehicle and the perpendicular line and θ_{ef} is the angle between the front of the ego vehicle and the perpendicular line.

The target vehicle projection on the line perpendicular to the relative velocity, p_t, reads:

$$p_t = |l_t \cos \theta_{ts}| + |w_t \sin \theta_{tf}|, \tag{4.18}$$

where l_t is length of the target vehicle, w_t is width of the target vehicle, θ_{ts} is the angle between the side of the target vehicle and the perpendicular line and θ_{tf} is the angle between the front of the target vehicle and the perpendicular line.

The minimum possible size of the projected center to center distance on the line perpendicular to the relative velocity that results in 0 overlap, χ_{min}, reads:

$$\chi_{min} = \frac{1}{2}\left(p_e + p_t\right), \tag{4.19}$$

where p_e and p_t are the ego and target vehicle projections on the line perpendicular to the relative velocity.

The actual size of the projected center to center distance on the line perpendicular to the relative velocity, χ, reads:

$$\chi = |d \cos\left(\theta_{relp} \pm \theta_{cc}\right)|, \tag{4.20}$$

where d is the center to center distance between the two vehicles, θ_{relp} is the angle of the line perpendicular to the relative velocity and θ_{cc} the angle of the line connecting the two CGs of the vehicles with the horizontal.

The projected overlap, ov, reads:

$$ov = |\chi_{min} - \chi| = |\frac{1}{2}\left(p_e + p_o\right) - |d \cos\left(\theta_{relp} \pm \theta_{cc}\right)||. \tag{4.21}$$

The center to center distance between the two vehicles, d, is calculated as

$$d = \sqrt{(x_t - x_e)^2 + (y_t - y_e)^2}, \tag{4.22}$$

where $(x_e, y_e)$and (x_t, y_t) are the x and y components of the ego and target vehicles respectively.

The tangent of the angle of the line connecting the two CGs of the vehicles with the horizontal, tan θ_{cc} is calculated as

$$\tan \theta_{cc} = \frac{(y_t - y_e)}{(x_t - x_e)}, \tag{4.23}$$

where $(x_e, y_e)$and (x_t, y_t) are the x and y components of the ego and target vehicles respectively.

4.4 Monte Carlo Simulation based

Monte Carlo Based Collision Probability Estimation The following collision prediction is based on the work presented in [LGPN08]. Unlike many authors, who implemented the estimation of the timely occupation of a vehicle by an ellipse, with a 90% probability accuracy of vehicle orientation and occupation representation, the Monte Carlo Based Collision Estimation presented in [LGPN08] makes use of the whole probability density functions (pdfs) of the volumes, positions and orientations of both the ego and target-vehicles, resulting in a more accurate collision prediction. The method takes into account both the three dimensional uncertainty of the vehicles and as well as their volumes. Moreover it can be used for collision prediction for the cases, when multiple obstacles are being tracked.

Vehicle Models The positions and orintations of the two vehicles, $\mathbf{x}_v$and $\mathbf{x}_o$, are expressed as

$$\mathbf{x}_v = (x_v, y_v, \theta_v)^T, \tag{4.24}$$

$$\mathbf{x}_o = (x_o, y_o, \theta_o)^T, \tag{4.25}$$

where the indexes v and o show those variables, which represent the ego and target vehicles respectively, (x_v, y_v) and$(x_o, y_o)$are the representative points of the vehicles and here considered as the geometric centers of the two rectangles representing the vehicles on a 2D map and $\theta_v$and $\theta_o$are the respective orientations of the ego vehicle and the target vehicle.

Uncertainty Modeling of the two vehicles From their respective mean state, $\hat{\mathbf{x}}$, and covariance,Σ, of the two vehicles, we can calculate their probability distribution function as

$$p(x) = \frac{1}{\left(\sqrt{2\pi}\right)^n \sqrt{det\left(\Sigma\right)}} e^{-\frac{1}{2}\left((\mathbf{x}-\hat{\mathbf{x}})\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\hat{\mathbf{x}})\right)}, \tag{4.26}$$

where n is the size of $\mathbf{x}$, i.e. 3 in this case and Σ is the covariance matrix.

The covariance, Σ, is given as

$$\boldsymbol{\Sigma} = \left(\left(\mathbf{x} - \hat{\mathbf{x}}\right)\left(\mathbf{x} - \hat{\mathbf{x}}\right)^T\right), \tag{4.27}$$

where $\mathbf{x}$ is the state vector and $\hat{\mathbf{x}}$ is the mean vector.

The mean and the covariance matrices can be obtained from Extended Kalman Filter or Unscented Kalman Filter.

A more complete state description of a vehicle can include the orientations, covariances and physical occupations in a vector.

Symbols v and o denote the ego vehicle and the target vehicle respectively and read:

$$v = (\mathbf{x}_v, \boldsymbol{\Sigma}_v, v_v), \tag{4.28}$$

$$o = (\mathbf{x}_o, \boldsymbol{\Sigma}_o, v_o), \tag{4.29}$$

where $\mathbf{x}_v$and $\mathbf{x}_o$ are the positions and orintations of the two vehicles, $\boldsymbol{\Sigma}_v$and $\boldsymbol{\Sigma}_o$ are the respective covariance matrices and ν_v and ν_o are used to denote the two polygonal representations of the vehicles.

Probability of collision for two vehicles with Gaussian uncertainties The probability of collision between two vehicles is the probability that their rectangular representations share the same space after some time. The collision probability, $p(v, o)$, can be written mathematically as

$$p(v,o) = \int_D p_v(x_v, y_v, \theta_v).p_o(x_o, y_o, \theta_o)\, dx_v dy_v d\theta_v dx_o dy_o d\theta_o, \tag{4.30}$$

where

$$D = \left\{(x_v, y_v, \theta_v, x_o, y_o, \theta_o)\, \epsilon R^6 | \nu_v(x_v, y_v, \theta_v) \cap \nu_o(x_o, y_o, \theta_o) \neq \phi\right\}. \tag{4.31}$$

This equation cannot be solved analytically. Nevertheless it can be solved using Monte Carlo Simulation.

Monte Carlo Solution for the Collision Probability The above equation can be written as

$$p(v,o) = \int_D \gamma\, p_v(x_v, y_v, \theta_v).p_o(x_o, y_o, \theta_o)\, dx_v dy_v d\theta_v dx_o dy_o d\theta_o, \tag{4.32}$$

where

$$\gamma = \left(v_v\left(x_v, y_v, \theta_v\right), v_o\left(x_o, y_o, \theta_o\right)\right) \tag{4.33}$$

is a collision test between the two geometries and is evaluated as

$$\gamma = \left\{1\, if\, \nu_v(x_v, y_v, \theta_v) \cap \nu_o(x_o, y_o, \theta_o) \neq \phi \quad 0\, if\, \nu_v(x_v, y_v, \theta_v) \cap \nu_o(x_o, y_o, \theta_o) = \phi\right. . \tag{4.34}$$

The algorithm is summarised in Algorithm 4.1. The algorithm generate radom states of both vehicles using their state means and covariances and check for intersection. perform this for N times. The collisison probability is calculated as the number of collision resulting states divided by N.

Algorithm 4.1 Faster computing of the probability of collision between v and o [LGPN08]

function FASTPROBABILITYOFCOLLISION(v, o)
  $p_{coll}(v, o) \leftarrow 0$
  for $j \leftarrow 1$ to N **do**
    $\mathbf{x}_v \leftarrow randc(\widehat{\mathbf{x}}_v, \mathbf{\Sigma}_v)$
    $\mathbf{x}_o \leftarrow randc(\widehat{\mathbf{x}}_o, \mathbf{\Sigma}_o)$
    if $\mathcal{V}_v(\mathbf{x}_v) \cap \mathcal{V}_o(\mathbf{x}_o) \neq \emptyset$ **then**
      $p_{coll}(v, o) \leftarrow p_{coll}(v, o) + 1$
    end if
  end for
  $p_{coll}(v, o) \leftarrow \frac{p_{coll}(v,o)}{N}$
  return$(p_{coll}(v, o))$
end function

4.5 Joint pdf Estimation based

The following model is presented in [Jan05]

The joint pdf of the ego and target vehicles is formed from their relative position to each other. The joint pdf is calculated from the future absolute position pdfs of both vehicles, which can be obtained by state estimation.The probability of collision is calculated by integrating the joint pdf over the area of their physical overlap, where collision would take place.

Symbol $Prob_{collision}$ is the probability of collision and reads:

$$Prob_{collision} = \max_{i}(Prob(x_{t+iT} \epsilon D)),\ i = 1, ..., N, \tag{4.35}$$

where$D$ represents the state space, which corresponds to collision.

Fig 4.11 shows the relative positional vector of the two vehicles and region of their physical overlap. The set of all possible orientations of collision are represented by D.

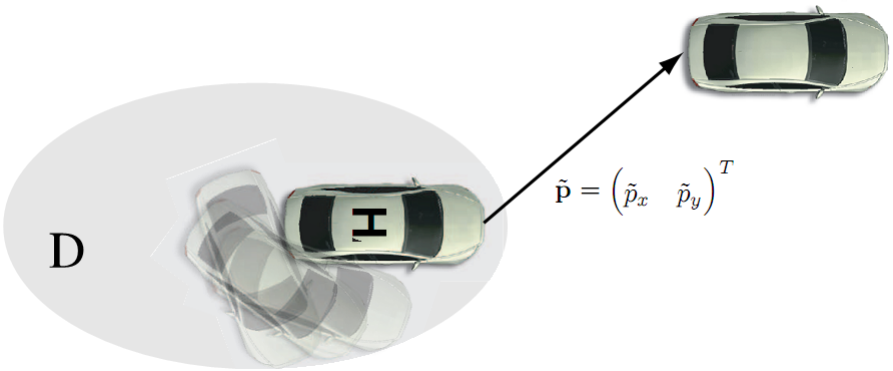


Figure 4.11: The set D corresponds to all possible configurations between the two vehicles, which results in collision [Jan05].

The Probability of collision for each time instant for three dimensional case, $Prob_{collision\,at\,time\,t+T}$, is calculated as

$$Prob_{collision\,at\,time\,t+T} = Prob(\tilde{p}_{x,t+T}, \tilde{p}_{y,t+T}, \tilde{\psi}_{t+T} \epsilon D), \tag{4.36}$$

and

$$Prob_{collision\,at\,time\,t+T} = \iiint_{\tilde{p}_x, \tilde{p}_y, \psi \tilde{\epsilon} D} p_{t+T}(\tilde{p}_x, \tilde{p}_y, \tilde{\psi}/Y) dx dy d\psi, \tag{4.37}$$

where p_{t+T} is the probability function of the relative position of the two vehicles, $\tilde{p}_x$ and $\tilde{p}_y$ are the x and y components of the relative position of the two vehicles, $\tilde{\psi}$ is the relative angle of the two vehicles, D is the region, where collision is expected to happen, Y is the measurement vector and T is the time span.

The above estimation problem can be reduced to a two dimensional problem by disregarding the heading angle and approximating D with a rectangle. The rectangle has width, which is equal to the sum of the widths of the two vehicles and length, which is equal to the length of the ego vehicle. An assumption of the in-dependency of the movement of the two vehicles in x and y direction is taken.

The probability of collision taking these assumptions, $Prob(collision\,at\,time\,t+i)$, is calculated as

$$Prob(collision\,at\,time\,t+i) = \iint_{\tilde{p} \epsilon D} p_{t+iT}(\tilde{p}_x, \tilde{p}_y/Y_t) dx dy, \tag{4.38}$$

where p_{t+T} is the probability function of the relative position of the two vehicles, $\tilde{p}_x$ and $\tilde{p}_y$ are the x and y components of the relative position of the two vehicles, D is the region, where collision is expected to happen, Y is the measurement vector and T is the time span.

Integrating the probability density on the area D , we get an expression, which involves the corresponding cumulative density functions. See the details in [Jan05].

4.6 Stochastic drivers' Input based

In this section the situation assessment of vehicle to vehicle collision in [KSD09] is discussed and some changes are suggested. The changes include use of a better simple realistic vehicle dynamic model for trajectory generation and also a collision index, which is quite easy to calculate from the measurable parameters of both vehicles.

4.6.1 Model I

In the mentioned work it is assumed that the inputs of the drivers of both the ego and the host vehicles are unknown. The algorithm considers all the combinations of the trajectories of the vehicles at the maximum handling limits of the vehicles. A collision is unavoidable, if any of these combinations doesn't result in collision avoidance. The algorithm takes the maximum possible acceleration of the vehicle and gets the longitudinal and lateral acceleration components of a certain maneuver with a given angle γ on the Kamm's circle for both the ego and target vehicles. This leads to the conclusion that $[\gamma_{ego}, \gamma_{target}]$ define the trajectory combinations of the two vehicles.

The longitudinal and lateral accelerations, a_l and a_c, are given as

$$a_l = a.\cos(\gamma), \tag{4.39}$$

$$a_c = a.\sin(\gamma), \tag{4.40}$$

where a is the acceleration of the vehicle and, γ maneuver angle on the Kamm's circle.

The limiting friction force, F_f , is the maximum force that the wheels can transmit to the surface of the road and is given as

$$F_f = \mu_0 mg, \tag{4.41}$$

where μ_0 road friction, m mass of the vehicle and g gravitational acceleration.

The force which is applied to move it, F_a, is calculated as

$$F_a = ma, \tag{4.42}$$

where m is the mass of the vehicle and, a is acceleration of the vehicle.

Then the maximum acceleration, a_{max}, is calculated as

$$a_{max} = \mu_0 g, \tag{4.43}$$

where μ_0 is the road friction, g is the gravitational acceleration.

The maximum acceleration of standard vehicles can reach up to 11 [m/s^2].

Fig 4.12 shows the Kamm circle. At 0°, the vehicle is fully accelerated , whereas at 180°, it is fully braking. At 90°, it is turning to the left, whereas at 270°, it is turning to the right. The acceleration, a, has longitudinal and lateral components, a_l and $a_c$respectively.

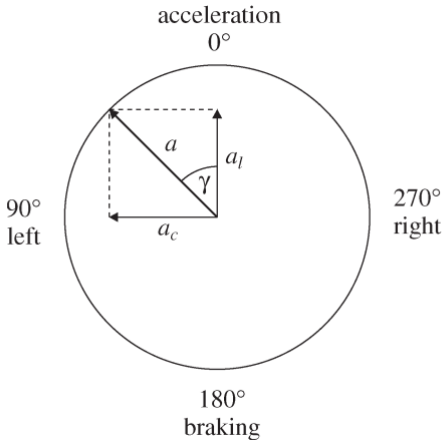


Figure 4.12: Kamm's circle[KSD09]

The motion model used for getting the trajectories is given with the following equations.

The acceleration, $\dot{v}(t)$, is given as

$$\dot{v}(t) = a_{max} \cos(\gamma), \tag{4.44}$$

where a_{max} is the maximum possible acceleration and γ is the maneuver angle on the Kamm circle.

Symbol $\dot{\psi}$denotes the yaw rate and reads

$$\dot{\psi} = min\left(\frac{a_{max}\sin(\gamma)}{v(t)}, \frac{v(t)}{r_{min}}\right), \tag{4.45}$$

where a_{max} is the maximum possible acceleration, γ is the maneuver angle on the Kamm's circle, $v(t)$ is the velocity of the vehicle and r_{min} is the minimum possible turning radius for the given velocity.

Symbols $\dot{x}_t$ and $\dot{y}_t$ denote the x and y components of the velocity respectively and read

$$\dot{x}_t = v(t)\cos(\psi(t)), \tag{4.46}$$

$$\dot{y}_t = v(t)\cos(\psi(t)), \tag{4.47}$$

where $v(t)$ is the velocity of the vehicle and $\psi(t)$ is the heading angle.

Fig 4.13 shows the possible trajectories for a =10 m/s2 and v_0 =20 m/s for a time prediction of 1 s on x-y coordinate system. The curves in blue color are the limiting trajectories.

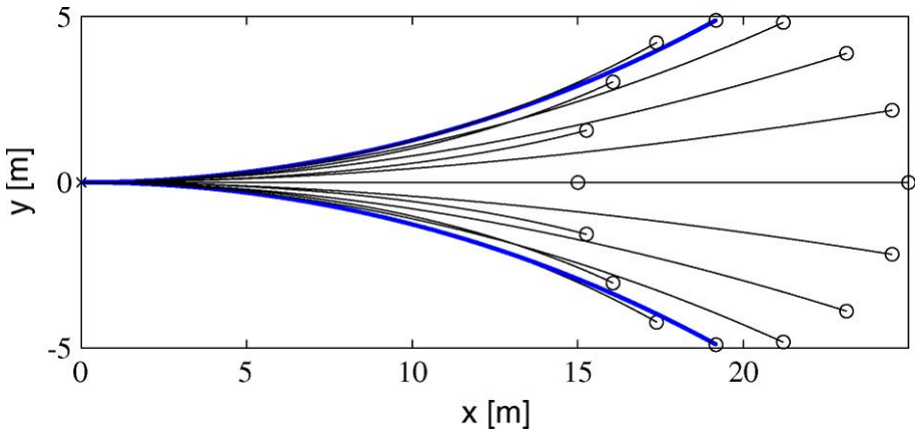


Figure 4.13: Possible trajectories for a =10 m/s2 and v_0 =20 m/s for a prediction of 1 s [KSD09] .

Kaemphen defined the size of the area of the opponent vehicle's body between the lines drawn from the two front corners of the ego vehicle towards the opponent vehicle in the direction of their relative velocity as a collision index. This collision index is directly

proportional to the extent of the crash, which means that it is very small in cases where the two vehicles touch barely to each other and large if they centrally crash. See Fig 4.14.

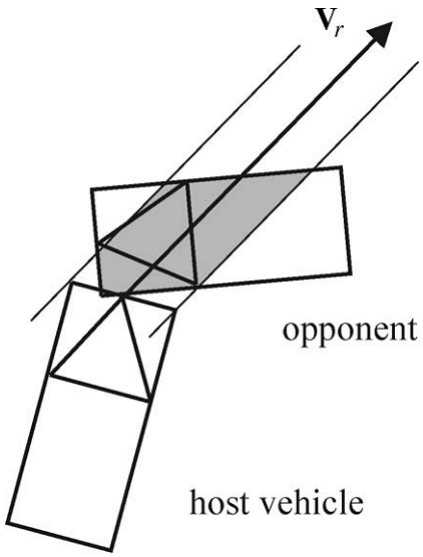


Figure 4.14: Collision index determination. The size of the gray area is the collision index [KSD09].

For a given trajectory combination with maneuver angles $[\gamma_{ego}, \gamma_{target}]$, the relative velocity $v_r = [v_x, v_y]$ is calculated from the velocities of both vehicles as

$$v_{r,x} = v_{x,ego} - v_{x,target} = v_{ego}\cos(\psi_{ego}) - v_{target}\cos(\psi_{target}), \tag{4.48}$$

$$v_{r,y} = v_{y,ego} - v_{y,target} = v_{ego}\sin(\psi_{ego}) - v_{target}\sin(\psi_{target}). \tag{4.49}$$

In the implementation of the algorithm searching for the the minimum collision index is crucial. If we find out that the collision matrix has a value of zero for a certain combination, it means that the collision can still be avoided. But if there is no any combination which leads to a collision index of zero the collision is inevitable and the safety system should be activated. Searching for the minimum collision index involves usage of gradient-descent algorithm in four different areas of the matrix. Each area corresponds to a certain trajectory combination of the vehicles. The number of of the local minima is always 1 or 2. If there are two local minima, they will be situated opposite to each other on the Kamm's circle.

4.6.2 Model II

In this model an alternative way of implementing the concept of the first model is suggested. The following changes are made:

A better motion model A better motion model, which considers the friction limits and engine torque is used. Basically the set of equations is quite similar to that of the one in section 3.11 but it is modified based on the model in [GCT08].

The rate of position of the vehicle in x and y direction, $\dot{x}$ and $\dot{y}$ are calculated as

$$\dot{x} = v \cos \beta, \tag{4.50}$$

$$\dot{y} = v \sin \beta, \tag{4.51}$$

where v is velocity and β is the side slip angle.

The acceleration of the vehicle in terms of the maximum engine power or the maximum acceleration, $\dot{v}$, which depends on the friction is given as

$$\dot{v} = \{ \begin{array}{c} u_1 \frac{k}{v} \, if \, v > v_{long} \, and \, u_1 \geq 0 \\ u_1 a_f \, else \end{array} , \tag{4.52}$$

where u_1 is longitudinal motion control input value, $\frac{k}{v}$ is engine power divided by the mass of the vehicle, a_f is the maximum acceleration on the friction ellipse and v_{long} is the limiting longitudinal velocity.

The rate of the slip with time of the vehicle in terms of the maximum steering angle or the maximum acceleration, $\dot{\beta}$, which depends on the friction is given as

$$\dot{\beta} = \{ \begin{array}{c} v \tan \left(\left(\delta_{max} u_2 \right) / L \right) \, if \, v \leq v_{lat} \\ u_2 a_f / v \, if \, v > v_{lat} \end{array} , \tag{4.53}$$

where δ_{max} is the maximum steering angle, u_2 is lateral motion control input value, L width of the wheel base of the vehicle, a_f is the maximum acceleration on the friction ellipse and v_{lat} is the limiting lateral velocity.

The rate of the heading angle with time of the vehicle in terms of the maximum steering angle or the maximum acceleration, which depends on the friction is given as

$$\dot{\theta} = \{ \begin{array}{c} v \sin\left(\left(\delta_{max} u_2\right)/L\right) \, if \, v \leq v_{lat} \\ u_2 a_f / v \, if \, v > v_{lat} \end{array} . \tag{4.54}$$

Using Euler's integration the above equations are integrated as follows

The position of the vehicle is calculated as

$$x_{k+1} = x_k + T \cdot v_k \cdot \cos(\beta_k), \tag{4.55}$$

$$y_{k+1} = y_k + T \cdot v_k \cdot \sin(\beta_k). \tag{4.56}$$

The velocity of the vehicle is calculated as

$$v_{k+1} = v_k + T\{ \begin{array}{c} u_1 \frac{k}{v} \, if \, v > v_{long} \, and \, u_1 \geq 0 \\ u_1 a_f \, else \end{array} . \tag{4.57}$$

The side slip angle of the vehicle, β, is calculated as

$$\beta_{k+1} = \beta_k + T\{ \begin{array}{c} v \tan\left(\left(\delta_{max} u_2\right)/L\right) \, if \, v \leq v_{lat} \\ u_2 a_f / v \, if \, v > v_{lat} \end{array} , \tag{4.58}$$

where T is time span, δ_{max} is the maximum steering angle, u_2 is lateral motion control input value, L width of the wheel base of the vehicle, a_f is the maximum acceleration on the friction ellipse and

v_{lat} is the limiting lateral velocity.

the heading angle of the vehicle, θ, is calculated as

$$\theta_{k+1} = \theta_k + T\{ \begin{array}{c} v \sin\left(\left(\delta_{max} u_2\right)/L\right) \, if \, v \leq v_{lat} \\ u_2 a_f / v \, if \, v > v_{lat} \end{array} , \tag{4.59}$$

where T is time span, δ_{max} is the maximum steering angle, u_2 is lateral motion control input value, L width of the wheel base of the vehicle, a_f is the maximum acceleration on the friction ellipse and v_{lat} is the limiting lateral velocity.

Trajectories of Arbitrary Inputs for both the Longitudinal and Lateral Motions All the input combinations both in the longitudinal and lateral directions are considered. In this model the vehicle is assumed to be controlled with the two control input values (u_1, u_2), where the first input is for the longitudinal motion and the second is for the lateral motion. Both have $[-1, 1]$ as value range.

Different Possible Input combination of both vehicles The probability of collision is calculated based on polygonal intersection algorithm and 16 possible combination of maneuver types of the two vehicles. The collision probability is calculated in a similar way to that implemented in 4.4. The advantage of doing this is that it makes it possible to assess, which combination results in high collision risk and if we couple this with tracking, it would be possible to estimate, the current combination of the inputs of the two vehicles making it possible for us to estimate the corresponding risk involved.

The possible combinations are given in Tabel 4.1.

Strategy	Ego Vehicle Motion	Ego Vehicle Turning	Target Vehicle Motion	Target Vehicle Turning
1	accelerated	right	accelerated	right
2	accelerated	left	deccelerted	right
3	accelerated	right	accelerated	right
4	accelerated	left	accelerated	left
5	accelerated	left	deccelerted	right
6	accelerated	right	deccelerted	left
7	accelerated	right	accelerated	left
8	accelerated	left	deccelerted	left
9	decceleraated	right	accelerated	right
10	decceleraated	right	deccelerted	right
11	decceleraated	left	accelerated	right
12	decceleraated	left	accelerated	left
13	decceleraated	left	deccelerted	right
14	decceleraated	right	deccelerted	left
15	decceleraated	left	accelerated	left
16	decceleraated	left	deccelerted	left

Table 4.1: Possible Maneuver combination

For each combination, equal number of arbitrary stochastic inputs are generated and then trajectories are produced. For each pair of trajectories generated from the initial conditions, polygon intersection check up is done. Monte Carlo simulation for a specified

number of input generation coupled with trajectory generation and collision check up is performed. The probability of collision for a certain maneuver combination is the number of input generation, which resulted in collision divided by the total number of input generations carried out. The same is done for all the possible combinations and also as a whole for any arbitrary maneuver combination.

Collision metrics based on transformation and intersection check-up of the rectangular representations The intersection of two rectangles is checked first by simulating both rectangles with the motion models and finding the final positions. Then multiplying all the four corners coordinates with both the translation and rotation matrices gives the final coordinates. But in the beginning each corner of the rectangle should be express in terms of the coordinates of the geometric centers, the dimensions of the rectangles and the angle that their longitudinal axes make with the horizontal. After getting the new coordinate values of the rectangles, the next step would be to check intersection. If there is intersection the function should return the value of 1 and if not it should return the value of 0.

The first step for the motion simulation is to randomly generate input values and based on the randomly generated input values corresponding trajectories are generated. Depending on whether this will cause the geometrical figures to intersect or not, the value of 1 or 0 is respectively assigned. By performing the simulation for as many input samples we want, lots of check up results are obtained. Finally when the sum of 1s of the the samples resulting in intersection is divided with the total sum of the the samples generated, the probability of collision is found.

Minimum Time of Collision for Each Pairs of Trajectories Generated The minimum time of collision is calculated for each pair of trajectories generated. And finally the minimum possible collision time of all the generated pair of trajectories is found out with Matlab Statistics Function min.

4.7 Model using covariance and space propagation

The following part is based on the work of [MZD09].

The ego-motion isignored for pre crash application, as the time duration is very short. The error can be considered in the error covariance with an assumption of error in acceleration with a magnitude of +/-10 m/s^2.

For activating the Pre-crash actuator, propagation of the states with their covariance matrices both in time and space, is an effective way of finding the probability of collision and knowing, when and where it is going to occur.

4.7.1 Measurement and State-Model

The measurement vector, $\mathbf{z}$, in polar coordinates for radar based tracking is

$$\mathbf{z} = [r_z \; \phi_z \; \dot{r}_z]^T \,, \tag{4.60}$$

where r_z is range, ϕ_z is azimuth angle and $\dot{r}_z$ is range rate.

The motion model used in this work is the nearly constant turn model and the state vector,$\mathbf{x}$, is given as

$$\mathbf{x} = [x_x \; y_x \; \dot{x}_x \; \dot{y}_x]^T \,, \tag{4.61}$$

where x_x is the x coordinate of the tracked object, y_x is the y coordinate of the tracked object, $\dot{x}_x$ the velocity of the tracked object in x direction and $\dot{y}_x$ the velocity of the tracked object in y direction.

4.7.2 Propagation to collision space

A non-linear distribution P_{space} of the state around the mean one is defined by taking N samples around the mean according to the the covariance matrix. The next step to get the non-Gaussian probability density is to non-linearly transform all the samples with the use of the following function, y_{hit}, which reads:

$$y_{hit}(x^{(i)}) = \left\{ \begin{array}{c} y_x - \frac{\dot{y}_x}{\dot{x}_x}.x_x \; \dot{x}_x \neq 0 \\ \infty \; otherwise \end{array} \right\}, \tag{4.62}$$

where $y_{hit}(x^{(i)})$ is the mean expected point of impact, x_x is the x coordinate of the tracked object, y_x is the y coordinate of the tracked object, $\dot{x}_x$ is the velocity of the tracked object in x direction and $\dot{y}_x$ is the velocity of the tracked object in y direction.

4.7.3 The collision space probability

The space covariance propagation p_{space} let us determine the collision space probability and reads:

$$p_{space} = \mathrm{P}\left(-\frac{w}{2} \leq y_{hit}(x^{(i)}) \leq \frac{w}{2}\right), \tag{4.63}$$

where w is the width of the target vehicle.

4.7.4 Propagation to collision time

Analyzing TTC of a vehicle tracked with a radar is very important to determine the right triggering time of the actuators in a pre-crash system of the ego vehicle. As it is mentioned above the transformation of the samples with the following non-linear functions results into the two main parameters used for defining the TTC. Then the TTC with its variance and mean is found out by calculation using the TTC formula.

The expected point of impact, see equation (4.69)

$$y_{hit}(x) = \left\{ \begin{array}{c} y_x - \frac{\dot{y}_x}{\dot{x}_x}.x_x \; \dot{x}_x \neq 0 \\ \infty \; otherwise \end{array} \right\} \tag{4.64}$$

The range of the target vehicle, r_{hit}, from the ego vehicle is calculated as

$$r_{hit} = \sqrt{x_x^2 + y_x^2}, \tag{4.65}$$

where x_x is the x coordinate of the tracked object and y_x is the y coordinate of the tracked object.

The mean angle of point of impact, $\phi_{hit}(x)$, is a calculated with the following formula

$$\phi_{hit}(x) = \left\{ \begin{array}{c} \arctan\left(\frac{y_{hit}(x) - y_x}{x_x}\right) \; if \; x_x \neq 0 \\ \infty \; otherwise \end{array} \right\}, \tag{4.66}$$

where $y_{hit}(x)$ is the mean expected point of impact, x_x is the x coordinate of the tracked object and y_x is the y coordinate of the tracked object.

The range rate, $\dot{r}_{hit}$ is calculated as follows

$$\dot{r}_{hit} = -\dot{x}_x . \cos(\phi_{hit}(x)) + \dot{y}_x . \sin(\phi_{hit}(x)), \tag{4.67}$$

where $\dot{x}_x$ is the velocity of the tracked object in x direction, $\dot{y}_x$ is the velocity of the tracked object in y direction and $\phi_{hit}(x)$ is the mean angle of point of impact.

The transformed TTC, $TTC_{hit}(x^{(i)})$, gives us the TTC distribution and is calculated as follows.

$$TTC_{hit}(x^{(i)}) = \frac{r_{hit}(x^{(i)})}{\dot{r}_{hit}\left(x^{(i)}\right)}, \tag{4.68}$$

where $r_{hit}(x^{(i)})$ is the mean range point of impact and $\dot{r}_{hit}\left(x^{(i)}\right)$ is the mean rate of change of range point of impact.

4.7.5 Actuator trigger

If the covariance propagation of the TTC lies between 200 and 400ms and the collision space probability is greater than experimental predetermined threshold, the actuator is activated [MZD09]. To make sure the trigger time accuracy, the variance of the TTC can also be evaluated and if its variance is below a certain threshold value, the actuators are triggered.

5 Simulation of Selected Tracking and Collision Models

In this chapter some of the models presented in the last two chapters are implemented. The first part deals with the implementation of the Kalman filter for tracking. In this part the purpose is to show the tracking model used for collision estimation and to show the suitability of the coordinated turn model due to its adaptive nature.

The second part deals with different vehicle collision prediction models. Some of the models are tested with different collision scenarios, which were generated with the use of the software package PC-Crash.

5.1 Simulation of Selected Vehicle Tracking Models

5.1.1 Radar based Constant Velocity Model with EKF and UKF

Fig 5.1 shows the result of tracking an object with the constant velocity model using a radar sensor. The trajectory is synthetically generated and the same model is used for collision assessment. The model is a modified version of the bearing only tracking demo model of ekf/ukf toolbox[HS07].

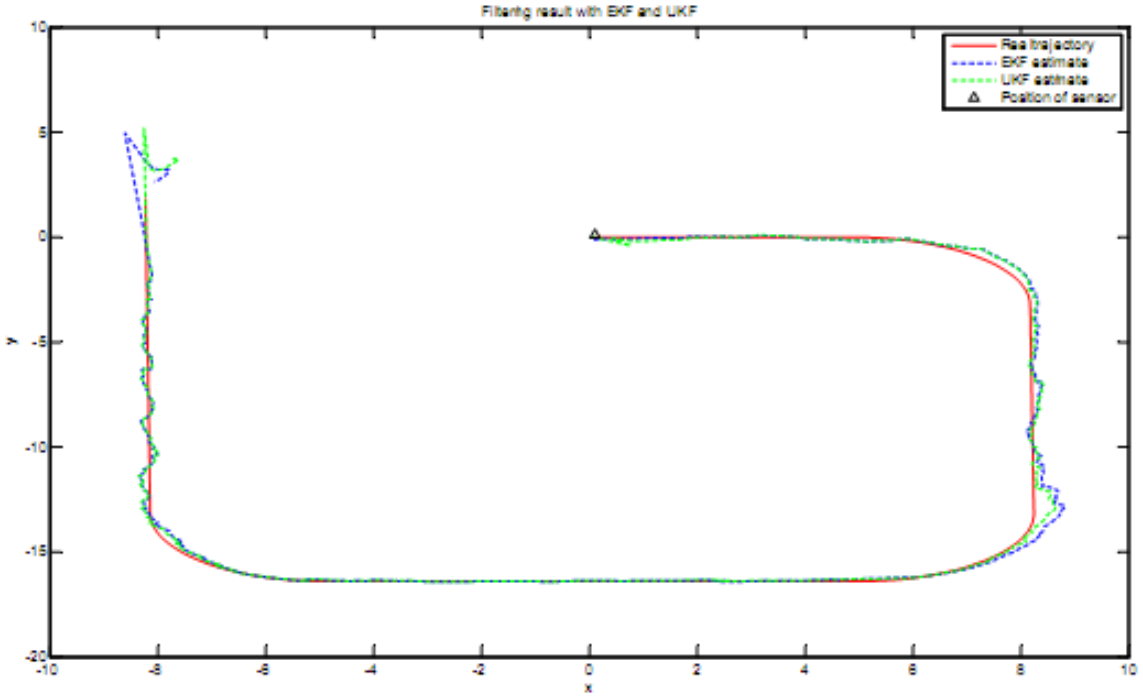


Figure 5.1: Radar measurement based tracking with a constant velocity motion model

Fig 5.2 and Fig 5.3 show the error involved in the tracking along x and y directions.

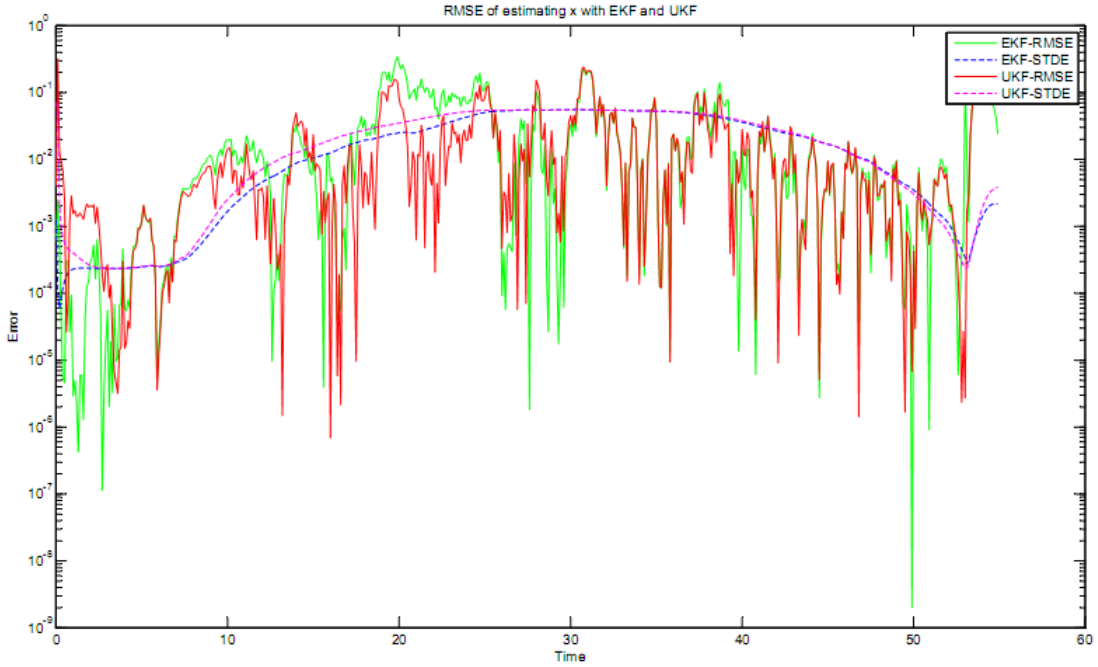


Figure 5.2: Error in tracking of x direction

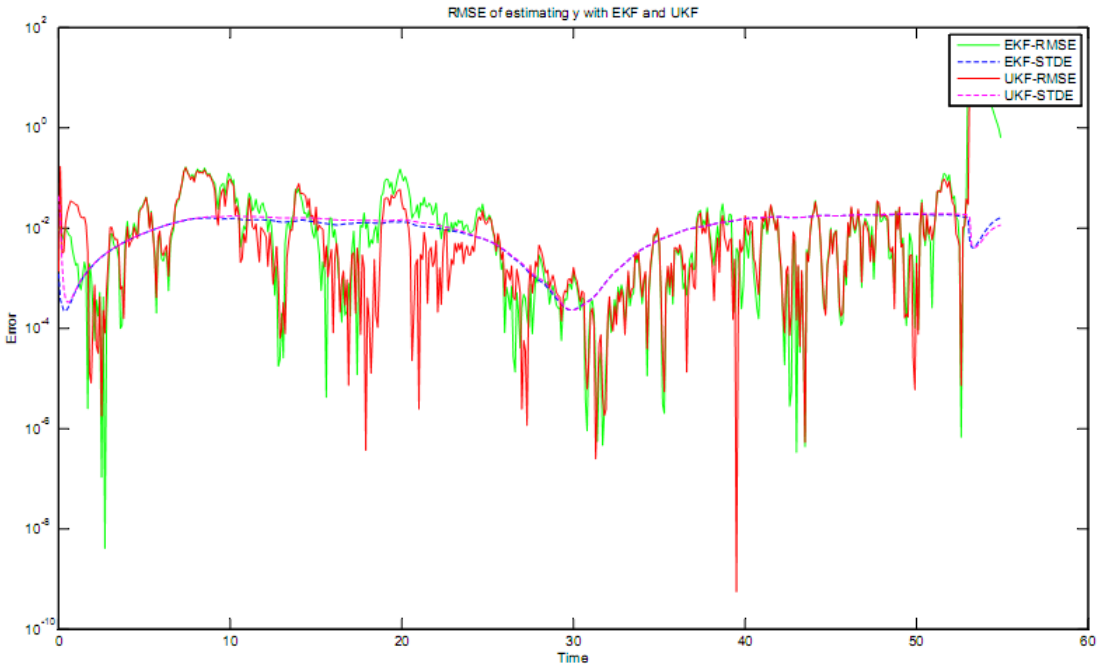


Figure 5.3: Error in tracking of y direction

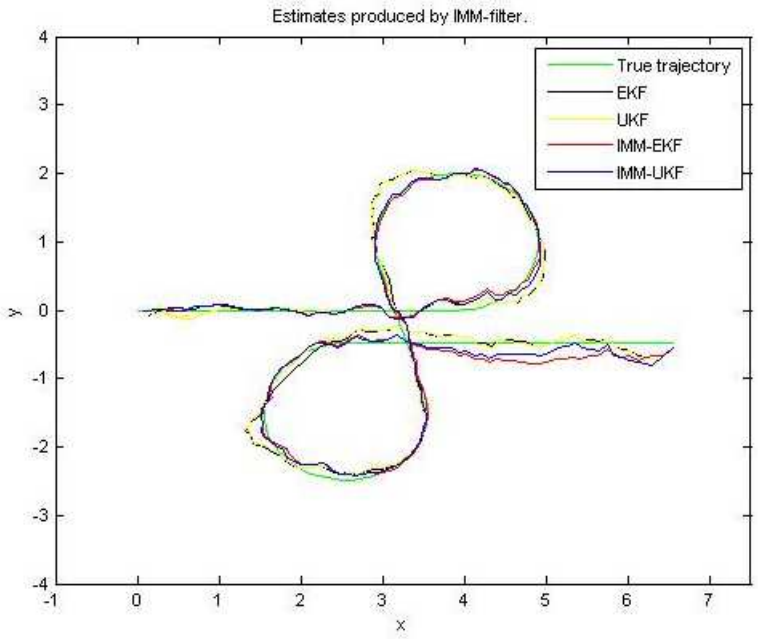


Figure 5.4: Radar measurement based tracking with an IMM Coordinated Turn model

5.1.2 Radar based Coordinated Turn Model with EKF and UKF

Fig 5.4, shows the result of radar measurement based tracking, which implements the IMM tracking. The model is a modified version of the bearing only coordinated tracking demo model of ekf/ukf toolbox [HS07]. Tracking vehicles along a straight road and a curved one is possible with this model. The following plot shows this.

Mean square errors of position estimates of this tracking example is given as follows and from the numbers we see that the IMM models perform better than the others. The Root Mean Square Error (RMSE) is used to measure the differences between the prediction values of an estimator and the actual values.

RMSE	value
EKF-RMSE	0.0108
UKF-RMSE	0.0106
EIMM-RMSE	0.0060
UIMM-RMSE	0.0046

Fig 5.4 shows that the turning rate is very closely tracked, exhibiting the advantage of the Coordinated Turn Model.

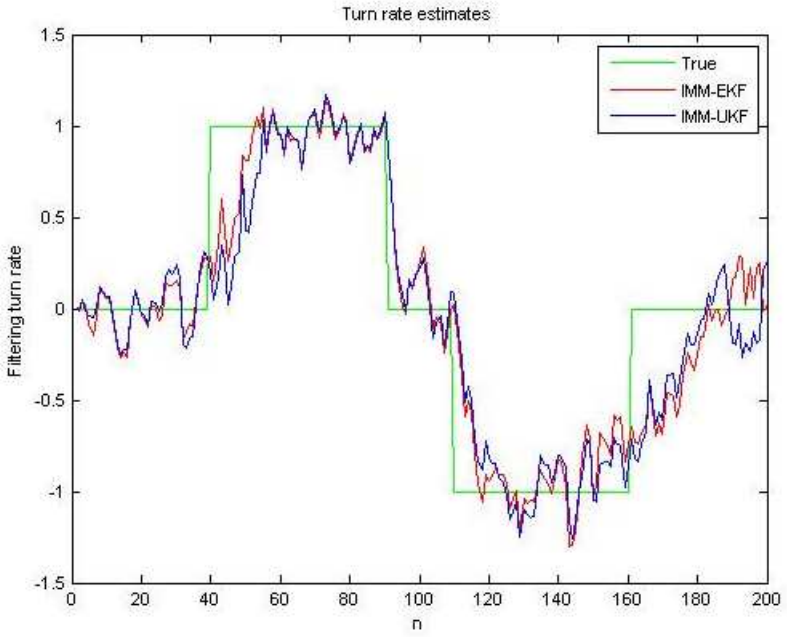


Figure 5.5: Estimation of the turning rate

5.2 Simulation of Selected Collision Estimation Models

5.2.1 Projection Overlap based

Here a simulation result of two scenarios are investigated with the use of collision prediction model based on projection overlap based, in which there is no offset in the first case and there is an offset in the other one. For the model the code in [JAC08] is used to calculate the minimum distance between the two rectangular representations of the vehicles.

Test case a In this case a full overlap head on collision is considered. In the table below we see the dependence of collision likeness on the distance between the two vehicles. Collision likeness increases as distance between the two vehicles gets less and less. Tabel 5.1 shows the output of this model by varying the distance between the two vehicles.

xe	ye	ve	betae	thetae	le	we	xt	yt	vt	beta	thetat	lt	wt	colliken	TTC	min_d	overlap	normalized overlap
0	0	50kph	0	0	2.5	1.515	10	0	70kph	0	0	4.65	1.84	4.955	0.197	6.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	9	0	70kph	0	0	4.65	1.84	5.865	0.163	5.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	8	0	70kph	0	0	4.65	1.84	7.19	0.133	4.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	7	0	70kph	0	0	4.65	1.84	9.29	0.103	3.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	6	0	70kph	0	0	4.65	1.84	13.121	0.073	2.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	5	0	70kph	0	0	4.65	1.84	22.328	0.043	1.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	4	0	70kph	0	0	4.65	1.84	74.866	0.013	0.425	1.515	0.954
0	0	50kph	0	0	2.5	1.515	3.875	0	70kph	0	0	4.65	1.84	106.06	0.009	0.3	1.515	0.954
0	0	50kph	0	0	2.5	1.515	3.775	0	70kph	0	0	4.65	1.84	159.09	0.006	0.2	1.515	0.954
0	0	50kph	0	0	2.5	1.515	3.675	0	70kph	0	0	4.65	1.84	318.18	0.003	0.1	1.515	0.954
0	0	50kph	0	0	2.5	1.515	3.575	0	70kph	0	0	4.65	1.84	Infinity	0	0	1.515	0.954

Table 5.1: Input data and simulation result of test Test case a

Where xe is x coordinate of ego vehicle, ye is y coordinate of ego vehicle, betae is the side slip angle of the ego vehicle, thetae is heading angle of the ego vehicle, le is length of the ego vehicle, we is width of the ego vehicle, xt is x coordinate of target vehicle, yt is y coordinate of target vehicle, betat is slip angle of the target vehicle, thetat is heading angle of the target vehicle, lt is length of the target vehicle, wt is width of the target vehicle, collikeness collision likeness, TTC is time to collision and min_d is the minimum distance between the two vehicles.

Fig 5.6 shows the relation between TTC and Overlap based Collision Likeness. The two variables are inversely related. As the TTC gets smaller and smaller, the collision likeness gets larger and larger.

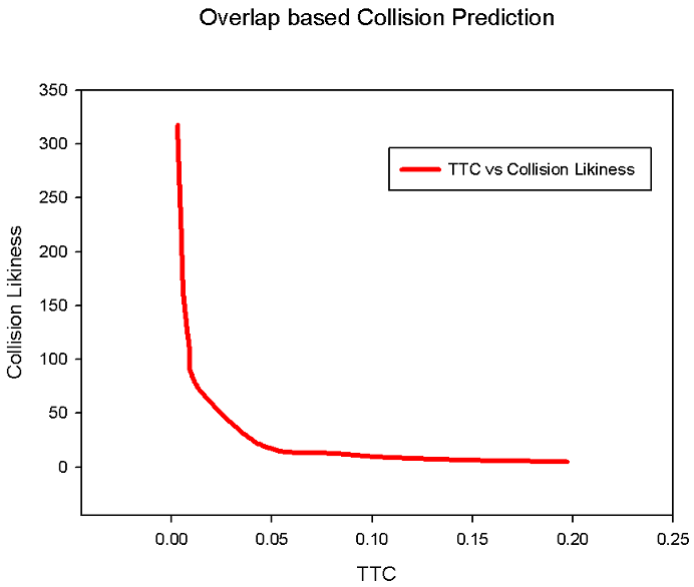


Figure 5.6: TTC versus Collision Likeness for a 0 offset case

Test Case b In this case a head on collision with an offset of 0.5m is considered. In comparison with Test Case b, the offset has an effect of lowering the collision likeness depending on its value. Tabel 5.2 shows the output of this model by varying the distance between the two vehicles in the presence of offset.

xe	ye	ve	betae	thetae	le	we	xt	yt	vt	betat	thetat	lt	wt	collikeness	TTC	min_d	overlap	normalized overlap
0	0	50kph	0	0	2.5	1.515	10	0.5	70kph	0	0	4.65	1.84	3.396	0.197	6.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	9	0.5	70kph	0	0	4.65	1.84	4.022	0.162	5.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	8	0.5	70kph	0	0	4.65	1.84	4.93	0.133	4.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	7	0.5	70kph	0	0	4.65	1.84	6.37	0.103	3.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	6	0.5	70kph	0	0	4.65	1.84	8.997	0.073	2.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	5	0.5	70kph	0	0	4.65	1.84	15.31	0.043	1.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	4	0.5	70kph	0	0	4.65	1.84	51.334	0.013	0.425	1.015	0.654
0	0	50kph	0	0	2.5	1.515	3.88	0.5	70kph	0	0	4.65	1.84	72.724	0.009	0.3	1.015	0.654
0	0	50kph	0	0	2.5	1.515	3.78	0.5	70kph	0	0	4.65	1.84	109.09	0.006	0.2	1.015	0.654
0	0	50kph	0	0	2.5	1.515	3.68	0.5	70kph	0	0	4.65	1.84	218.17	0.003	0.1	1.015	0.654
0	0	50kph	0	0	2.5	1.515	3.58	0.5	70kph	0	0	4.65	1.84	Infinity	0	0	1.015	0.654

Table 5.2: Input data and simulation result of test Test case b

For the definition of the variables, see Test Case a.

The effect of the offset is a reduced collision likeness as compared to the above one as shown in Fig 5.7, where there was no offset but the relation between TTC and Collision Likeness is similar.

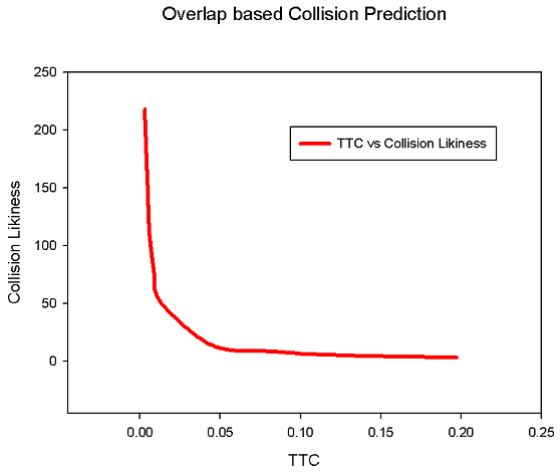


Figure 5.7: TCC versus Collision Likeness for an offset of 0.5m case

5.2.2 Monte Carlo Simulation based

Here a simulation scenario of a rear end collision of two vehicles in which a vehicle driving with a speed of 70 kph following a vehicle driving with 20 kph is presented. For the implementation of this model, the code in [Gia09] and [Fel07] is used for polygonal intersection check up and and creating a gaussian distribution of the states from the mean and covariance of each vehicle.

Fig 5.8 shows the situation after 0.3s. The collision probability is 0 at this moment. From the diagram, one can understand that the spread of each vehicle around the mean depends on its covariance matrix of state estimation.

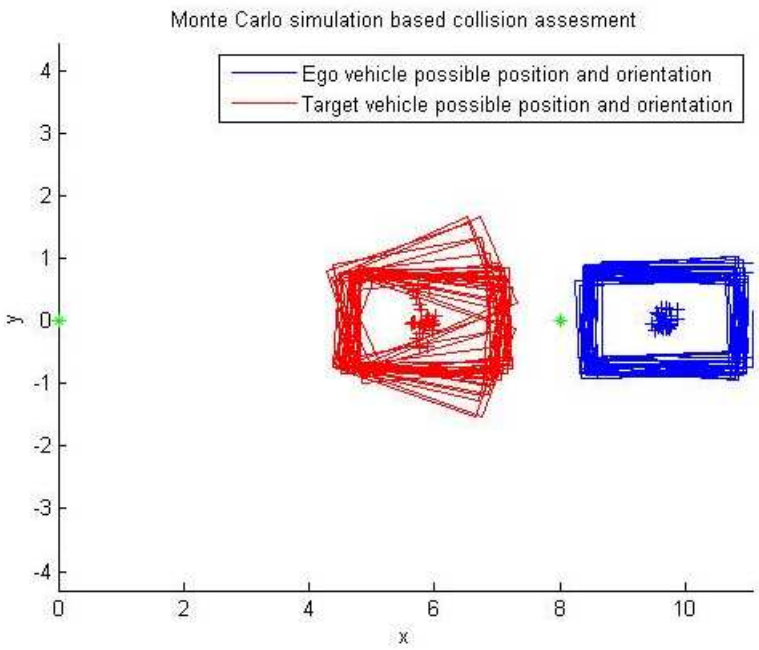


Figure 5.8: Orientation of the vehicles after 0.3s

Fig 5.9, shows the situation after 0.4s. The collision probability is 0.567 at this moment. More than half of the cases result in collision.

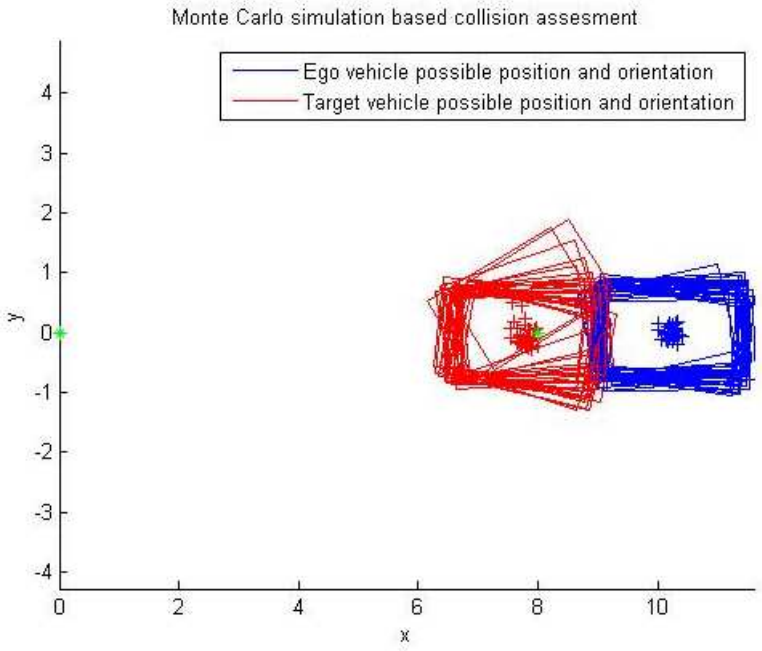


Figure 5.9: Orientation of the vehicles after 0.4s

Fig 5.9, shows the situation after 0.45s. The collision probability is 0.967 at this moment. At this moment the occurrence of collision is inevitable.

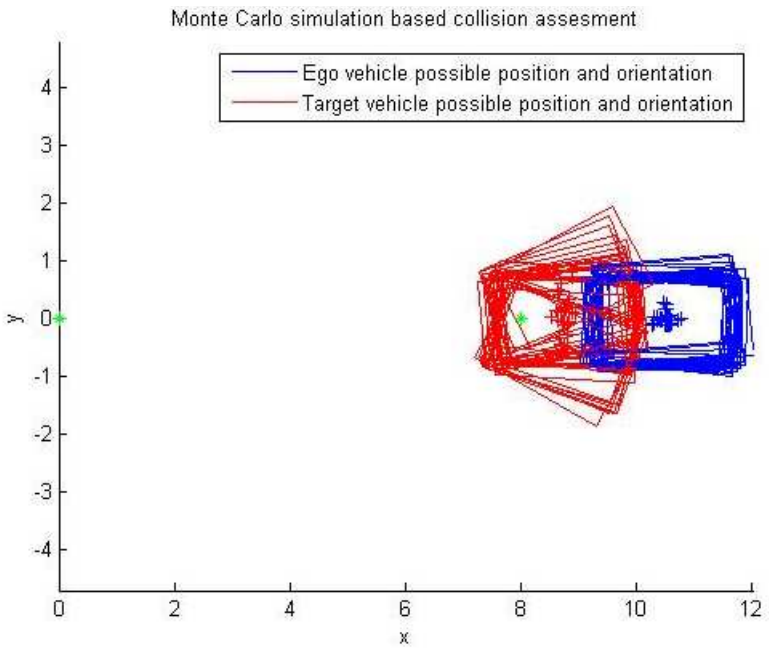


Figure 5.10: Orientation of the vehicles after 0.967

In this simullation , Le=2.4 is length of the ego vehicle, we=1.5 is width of the vehicle, toe=0 is initial time of the ego vehicle, xoe=0 is initial x value of the ego vehicle, yoe=0 is initial y value of the ego vehicle, voe=70kph is initial x value of the ego vehicle, phioe=0 is initial heading the ego vehicle, betaoe=0 is initial slip angle of the ego vehicle, lt=2.4 is length of the target vehicle, wt=1.5 is width of the target vehicle, tot=0 is initial time of the target vehicle, xot=8 is initial x value of the target vehicle, yot=0 is initial y value of the target vehicle, vot=20kph is initial x value of the target vehicle, phiot=0 is initial heading the target vehicle and betaot=0 is initial slip angle of the target vehicle.

5.2.3 collision prediction using stochastic drivers' Input

Here the simulation of two accident scenarios are discussed, in which one is without an offset and the other with an offset. For the implementation of this model, the code from [Gia09] is used for polygonal intersection check up.

Test Case 1a

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (10,0), the slip angle is 0, the heading angle is 180° and its speed is 70kph. The minimum possible collision time=0.217 and the total probability of collision=0.052. Overall the danger of collision at this moment is small. See Fig 5.11 and Fig 5.12.

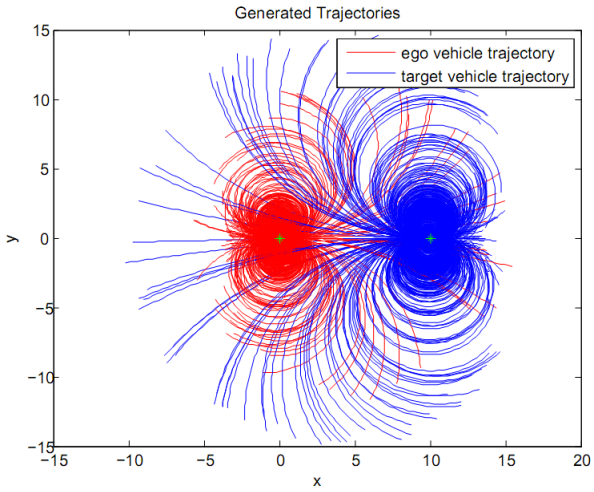


Figure 5.11: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

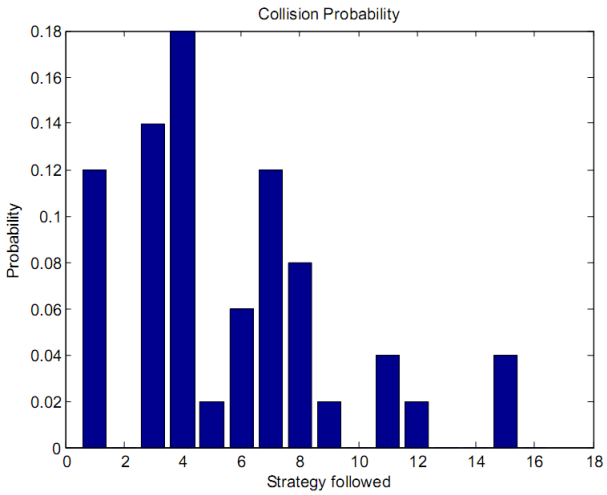


Figure 5.12: Collisison Probabilities of the Strategies of 1a

Test Case 1b

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (8,0), the slip angle is 0, the heading angle is 180° and its speed is 70kph.

The minimum possible collision time=0.15 and the total probability of collision=0.086. Overall the danger of collision at this moment is small. The one with the largest collision probability is strategy 4 with a corresponding value of 0.28. See Fig 5.13 and Fig 5.14.

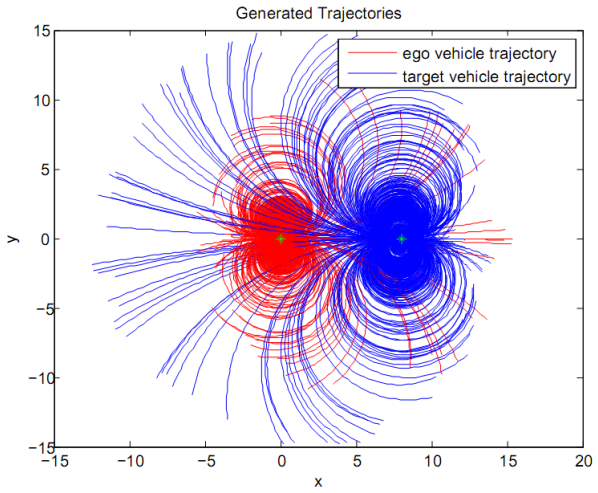


Figure 5.13: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1b

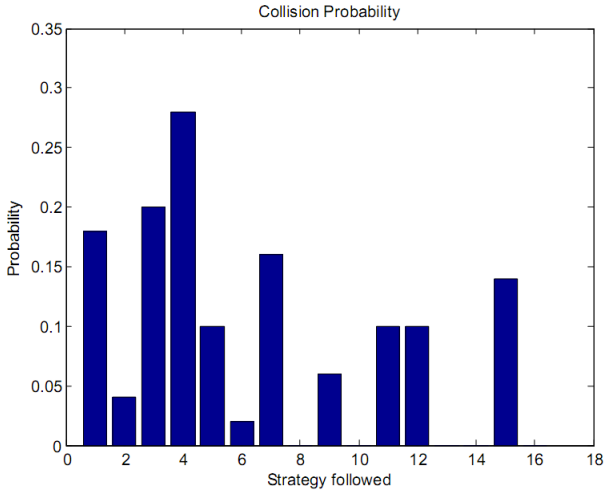


Figure 5.14: Collisison Probabilities of the Strategies 1b

Test Case 1c

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (6,0), the slip angle is 0, the heading angle is 180° and its speed is 70kph. The minimum possible

collision time=0.117 and the total probability of collision=0.216. See Fig 5.15 and Fig 5.16.

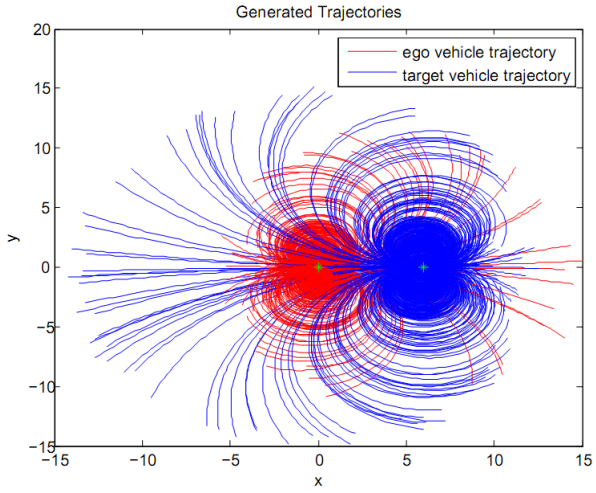


Figure 5.15: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1c

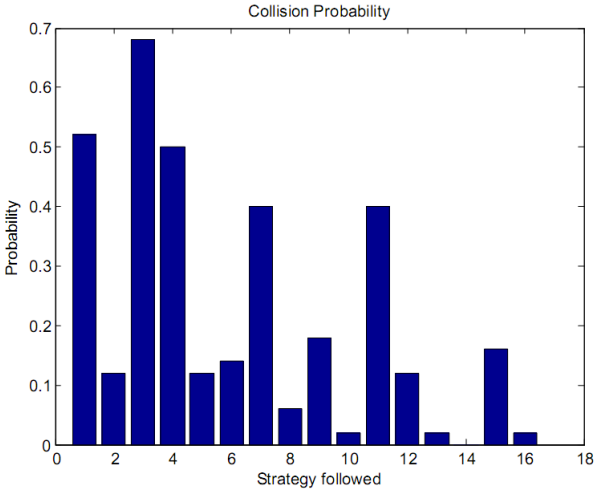


Figure 5.16: Collisison Probabilities of the Strategies 1c

Test Case 1d

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (4,0), the slip angle is 0, the heading angle is 180° and its speed is 70kph.

The minimum possible collision time=0.033, and the total probability of collision=0.999. Here collision is inescapable. See Fig 5.17 and Fig 5.18.

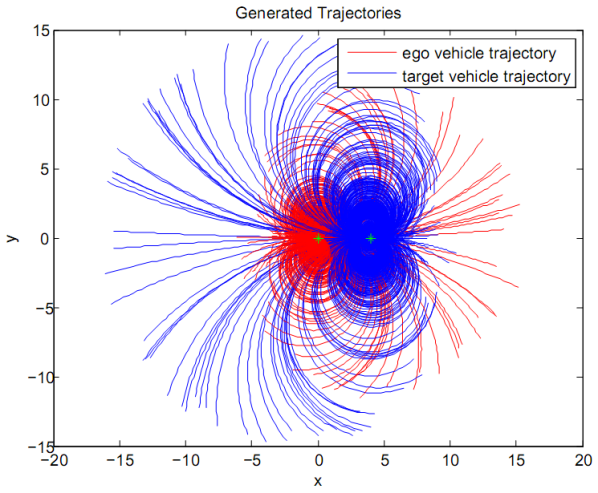


Figure 5.17: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1d

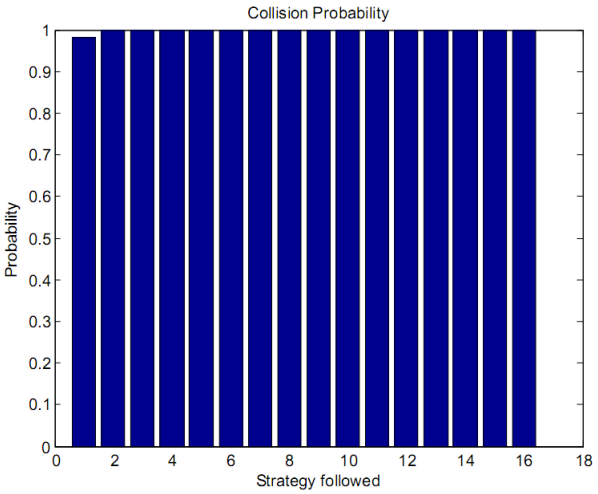


Figure 5.18: Collisison Probabilities of the Strategies 1d

As can be seen in the following table, Tabel 5.3, the minimum possible collision time and the total probability of collision increases as the distance between the vehicle gets less and less.

distance between the vehicle centers	10	8	6	4
The minimum possible collision time	0.217	0.15	0.117	0.033
The total probability of collision	0.052	0.086	0.216	0.999
Collision Probability of strategy 1	0.12	0.18	0.52	0.98
Collision Probability of strategy 2	0	0.04	0.12	1
Collision Probability of strategy 3	0.14	0.2	0.68	1
Collision Probability of strategy 4	0.18	0.28	0.5	1
Collision Probability of strategy 5	0.02	0.1	0.12	1
Collision Probability of strategy 6	0.06	0.02	0.14	1
Collision Probability of strategy 7	0.12	0.16	0.4	1
Collision Probability of strategy 8	0.08	0	0.06	1
Collision Probability of strategy 9	0.02	0.06	0.18	1
Collision Probability of strategy 10	0	0	0.02	1
Collision Probability of strategy 11	0.04	0.1	0.4	1
Collision Probability of strategy 12	0.02	0.1	0.12	1
Collision Probability of strategy 13	0	0	0.02	1
Collision Probability of strategy 14	0	0	0	1
Collision Probability of strategy 15	0.04	0.14	0.16	1
Collision Probability of strategy 16	0	0	0.02	1

Table 5.3: Simulation result of test Test case 1

Test Case 2a

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (10,3), the slip angle is 0, the heading angle is 180° and its speed is 70kph.(10m distance in between)

The minimum possible collision time=0.25 and the total probability of collision=0.04. The probability of collision is quite small at this specific moment. See Fig 5.19 and Fig 5.20.

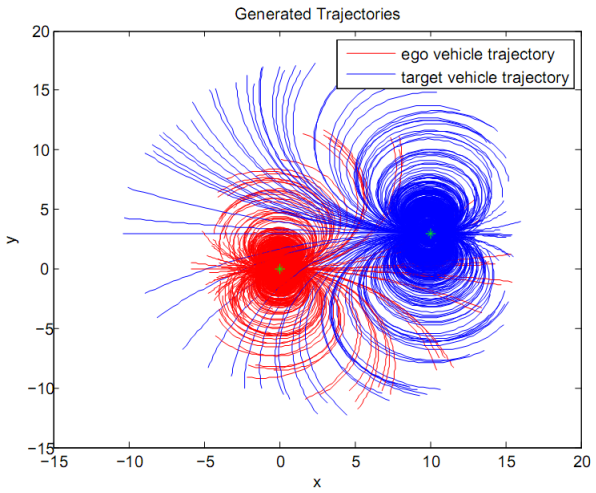


Figure 5.19: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

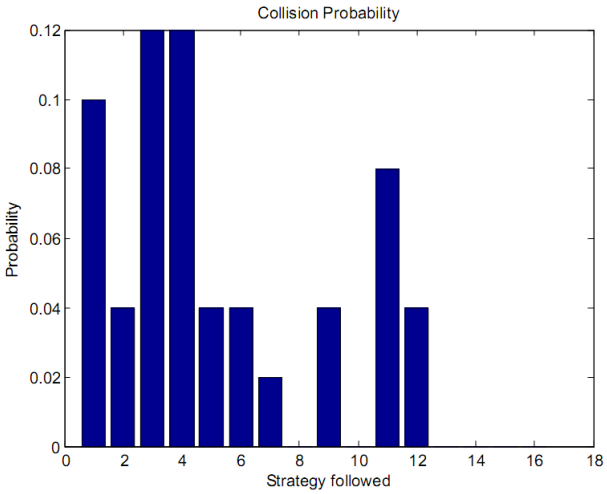


Figure 5.20: Collisison Probabilities of the Strategies 2a

Test Case 2b

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (8,3), the slip angle is 0, the heading angle is 180° and its speed is 70kph.(8m distance in between)

The minimum possible collision time=0.167 and the total probability of collision=0.065. See Fig 5.21 and Fig 5.22.

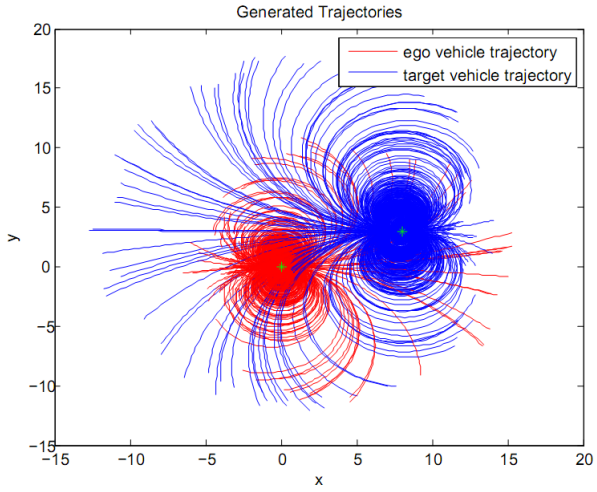


Figure 5.21: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

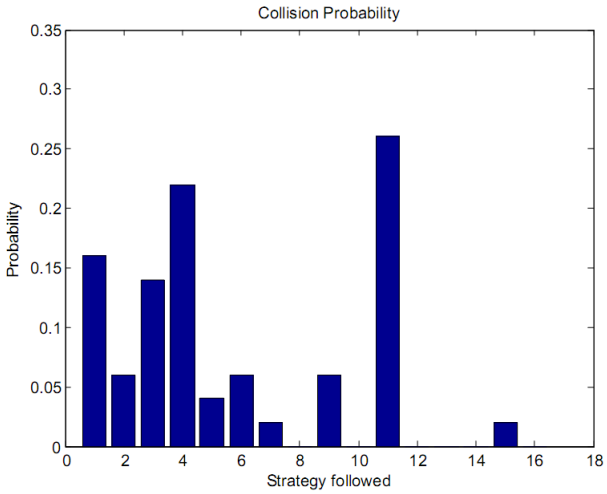


Figure 5.22: Collisison Probabilities of the Strategies 2b

Test Case 2c

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (6,3), the slip angle is 0, the heading angle is 180° and its speed is 70kph.(6m distance in between)

The minimum possible collision time=0.133 and the total probability of collision=0.123. Overall the danger of collision at this moment is small but in some specific cases is getting higher. See Fig 5.23 and Fig 5.24.

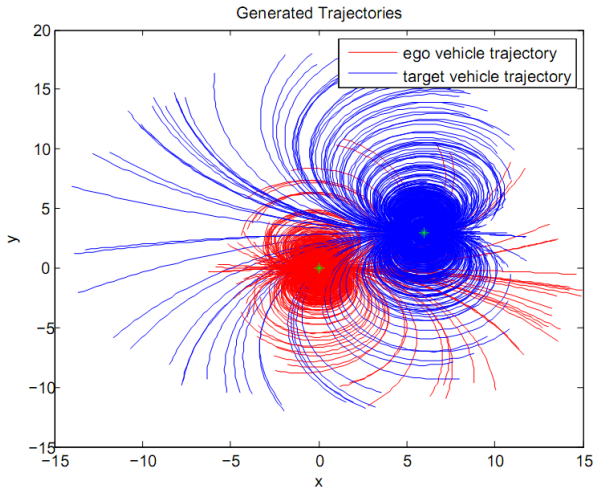


Figure 5.23: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

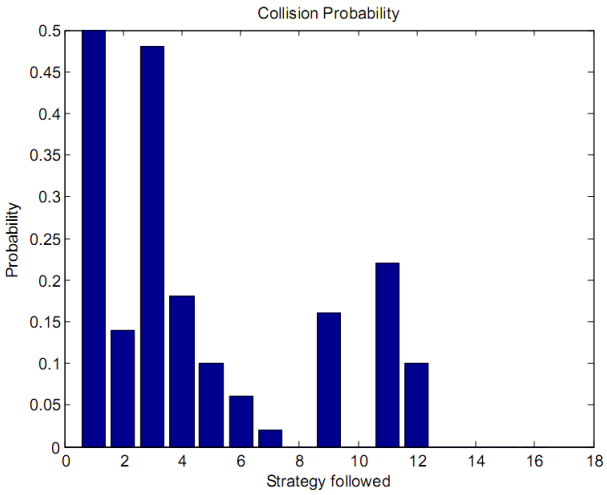


Figure 5.24: Collisison Probabilities of the Strategies 2c

Test Case 2d

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (4,3), the slip angle is 0, the heading angle is 180° and its speed is 70kph.(4m distance in between)

The minimum possible collision time=0.067 and the total probability of collision=0.429. Overall the danger of collision at this moment is getting higher and higher. Among all the strategies strategy 1,2,3,4 and 9 have the highest collision probability. See Fig 5.25 and Fig 5.26.

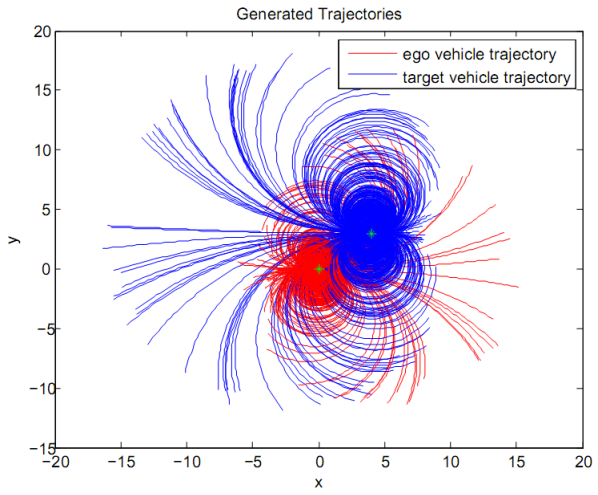


Figure 5.25: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

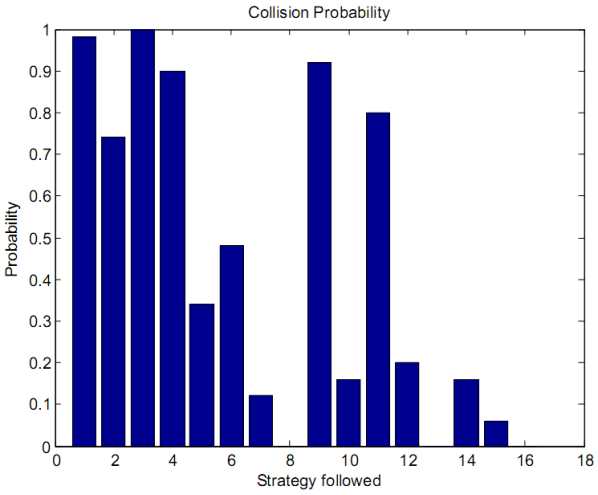


Figure 5.26: Collisison Probabilities of the Strategies 2d

Test Case 2e

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (3.6,3), the slip angle is 0, the heading angle is 180° and its speed is 70kph.(3.6m distance in between)

The minimum possible collision time=0.05 and the total probability of collision=0.522. Overall the danger of collision at this moment is getting higher and higher. Among all the strategies strategy 1,2,3,4,6,9 and 11 have the highest collision probability. See Fig 5.27 and Fig 5.28.

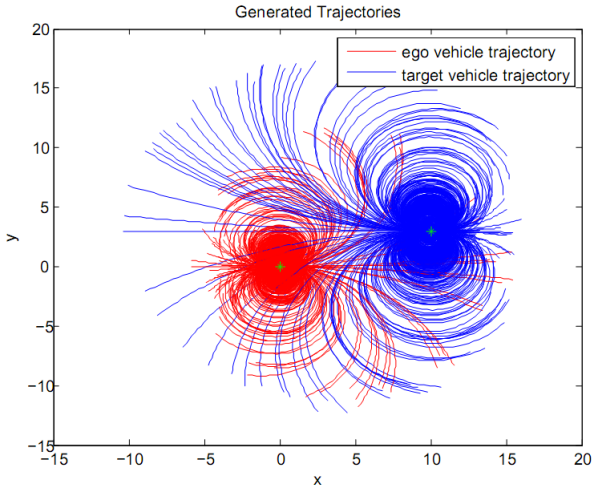


Figure 5.27: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

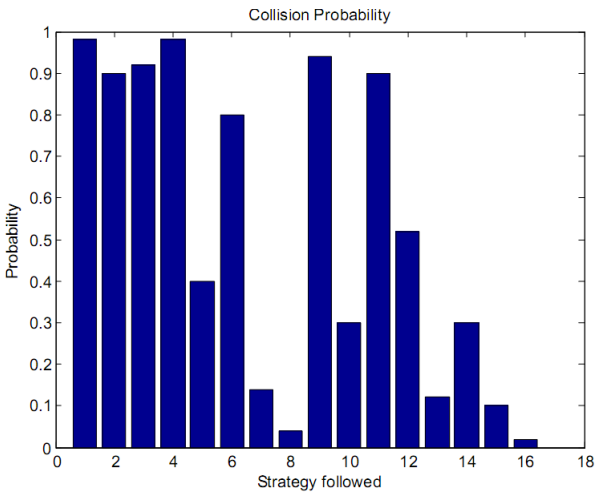


Figure 5.28: Collisison Probabilities of the Strategies 2e

Test Case 2f

The initial coordinate of the ego vehicle is (0,0), both the slip and heading angles are equal to 0 and its speed is 50kph. The initial coordinate of the target vehicle is (6,0), the slip angle is 0, the heading angle is 180° and its speed is 70kph.(3m distance in between)

The minimum possible collision time=0.05 and the total probability of collision=0.669. As can be seen in the following table the minimum possible collision time and the total probability of collision increases as the distance between the vehicle gets less and less. See Fig 5.29 and 5.30.

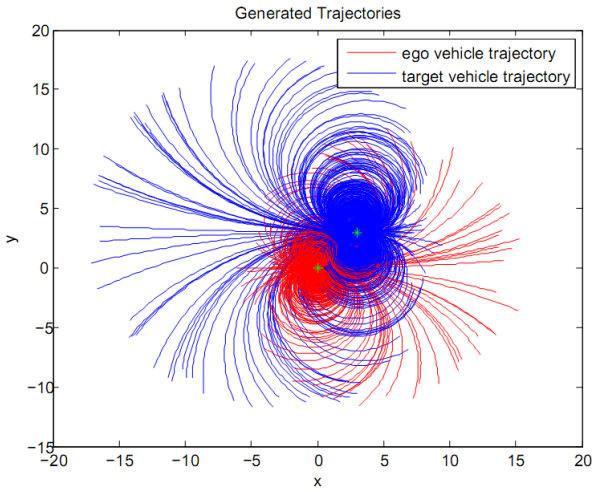


Figure 5.29: Trajectories generated and Collision probability of a strategy followed by the two vehicles of Test Case 1a

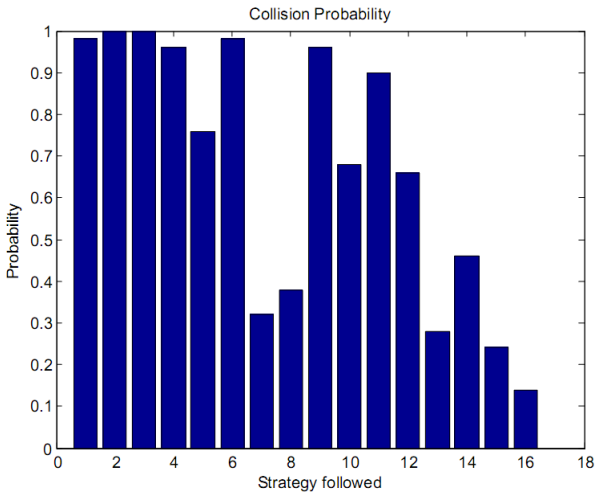


Figure 5.30: Collisison Probabilities of the Strategies 2f

As can be seen in the following table, Tabel 5.4, the minimum possible collision time and the total probability of collision increases as the distance between the vehicles gets less and less. Compared to the result of Table 5.3, we see the corresponding collision of probability to an equivalent distance in x direction is less due to the offset.

distance between the vehicle centers	10	8	6	4	3.6	3
The minimum possible collision time	0.25	0.167	0.133	0.067	0.05	0.05
The total probability of collision	0.04	0.065	0.123	0.429	0.522	0.669
Collision Probability of strategy 1	0.1	0.16	0.5	0.98	0.98	0.98
Collision Probability of strategy 2	0.04	0.06	0.14	0.74	0.9	1
Collision Probability of strategy 3	0.12	0.14	0.48	1	0.92	1
Collision Probability of strategy 4	0.12	0.22	0.18	0.9	0.98	0.96
Collision Probability of strategy 5	0.04	0.04	0.1	0.34	0.4	0.76
Collision Probability of strategy 6	0.04	0.06	0.06	0.48	0.8	0.98
Collision Probability of strategy 7	0.02	0.02	0.02	0.12	0.14	0.32
Collision Probability of strategy 8	0	0	0	0	0.04	0.38
Collision Probability of strategy 9	0.04	0.06	0.16	0.92	0.94	0.96
Collision Probability of strategy 10	0	0	0	0.16	0.3	0.68
Collision Probability of strategy 11	0.08	0.26	0.22	0.8	0.9	0.9
Collision Probability of strategy 12	0.04	0	0.1	0.2	0.52	0.66
Collision Probability of strategy 13	0	0	0	0	0.12	0.28
Collision Probability of strategy 14	0	0	0	0.16	0.3	0.46
Collision Probability of strategy 15	0	0.02	0	0.06	0.1	0.24
Collision Probability of strategy 16	0	0	0	0	0.02	0.14

Table 5.4: Simulation result of test Test case 2

5.2.4 TTC and Space Propagation based

Here in this section the simulation result of different accident scenarios, which are based on TTC and space propagation are presented. Here a modified version of the bearing only tracking demo model of ekf/ukf toolbox in [HS07] and the code in [Fel07] is used. The measurement update is 0.01s but it can be made larger depending on the time span that we want to look each time in the future. Here as the ekf/ukf toolbox of Kalman filtering is used for tracking, the time span for prediction and measurement are the same.

Tabel 5.5 and Tabel 5.6 are the test matrices of the test scenarios examined with the mentioned algorithm. The first tabel is for the ego vehicle data and the second tabel is for the corresponding data of the target vehicle. All together, it consist of 18 different scenarios. The simulations were performed with PC-Crash with a VW Golf, as an ego vehicle and and Opel Astra as a target vehicle. Symbol vx is the longitudinal velocity of each vehicle, pedal position value is in between 1 (full acceleration) and -1 (full braking). the steering angle is the angle measured at the wheel. Overlap shows, how much percent of the vehicles are to be involved in collision.

Test Matrix	ID	Ego-vehicle	vx	pedal position	steering angle	overlap	collision
Head on	001	VW Golf 5	60	0	0	100%	1.020 s
Rear-end	002	VW Golf 5	60	0	0	100%	1.350 s
Side impact	003	VW Golf 5	60	0	-4.2 deg after 1 sec	-	4.125 s
Head on	004	VW Golf 5	60	-1	0	100%	no collision
Head on	005	VW Golf 5	60	1	0	100%	0.780 s
Rear-end	006	VW Golf 5	60	-1	0	100%	no collision
Rear-end	007	VW Golf 5	60	1	0	100%	0.945 s
Head on	008	VW Golf 5	60	0	0	50%	1.020 s
Head on	009	VW Golf 5	60	0	0	10%	1.020 s
Head on	010	VW Golf 5	60	0	0	0%	no collision
Side to side	011	VW Golf 5	0	0.7	-4.2 deg after 1 sec	-	4.661 s
Avoidance	012	VW Golf 5	60	-1	evasive	-	no collision
Avoidance	013	VW Golf 5	60	0	evasive	-	0.735 s
Head on	014	VW Golf 5	60	0	0	100%	0.840 s
Sliding	015	VW Golf 5	40	0	0	-	no collision
Head on	016	VW Golf 5	50	0	5	100%	6.950 s
Side to side	017	VW Golf 5	50	0	-5		5.790 s
Stationary eg	018	VW Golf 5	0	0	0	100%	2.780 s

Table 5.5: Test Scenario Matrix a

Test Matrix	ID	Obstacle	vx	pedal position	steering angle	overlap	collision
Head on	001	Opel Astra	10	0	0	100%	1.020 s
Rear-end	002	Opel Astra	10	0	0	100%	1.350 s
Side impact	003	Opel Astra	11	0	14 deg after 1 sec	-	4.125 s
Head on	004	Opel Astra	10	-1	0	100%	no collision
Head on	005	Opel Astra	10	1	0	100%	0.780 s
Rear-end	006	Opel Astra	10	1	0	100%	no collision
Rear-end	007	Opel Astra	10	-1	0	100%	0.945 s
Head on	008	Opel Astra	10	0	0	50%	1.020 s
Head on	009	Opel Astra	10	0	0	10%	1.020 s
Head on	010	Opel Astra	10	0	0	0%	no collision
Side to side	011	Opel Astra	0	0.3	14 deg after 1 sec	-	4.661 s
Avoidance	012	Opel Astra	80	0	0	-	no collision
Avoidance	013	Opel Astra	80	0	0	-	0.735 s
Head on	014	Opel Astra	10	0	0	100%	0.840 s
Sliding	015	Opel Astra	40	0	0	-	no collision
Head on	016	Opel Astra	50	0	5	100%	6.950 s
Side to side	017	Opel Astra	50	0	5		5.790 s
Stationary eg	018	Opel Astra	30	0	follow	100%	2.780 s

Table 5.6: Test Scenario Matrix b

Test Scenario1 Fig 5.31 shows the vehicle collision orientation for Test 1.

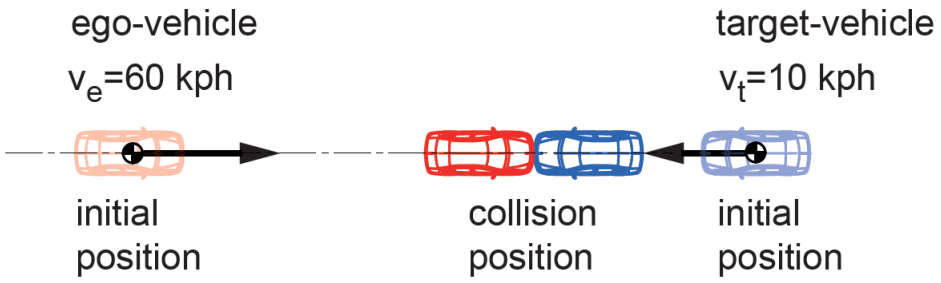


Figure 5.31: The final incidence orientation of test 1

Fig 5.32 shows the trajectory prediction of the last 0.6s before collision.

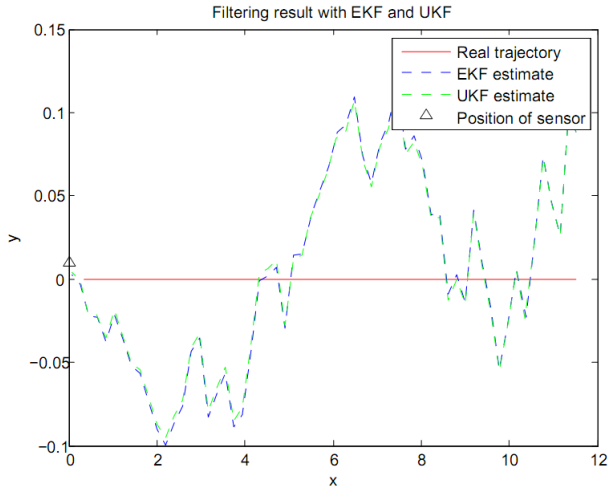


Figure 5.32: Trajectory of test 1

Fig 5.33 and Fig 5.34 show the error involved in the prediction of x and y coordinates respectively.

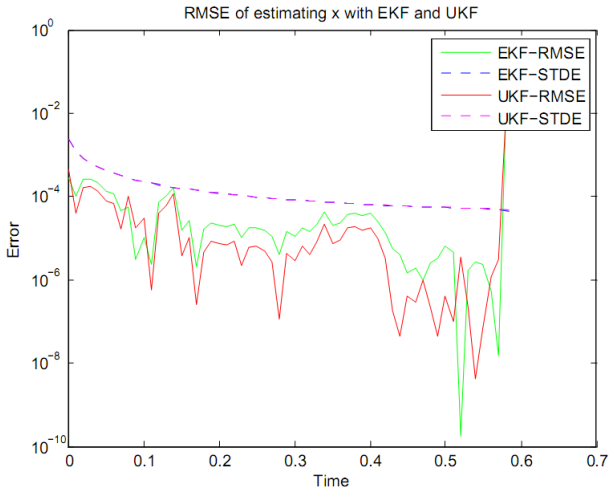


Figure 5.33: RMSE of x and y estimation of test 1

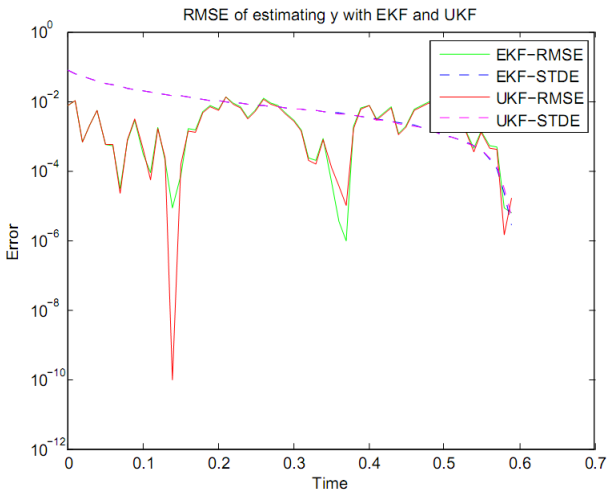


Figure 5.34: RMSE of y estimation of test 1

In Fig 5.35, TTC goes down to zero and in Fig 5.36, the point of first contact lies between -0.15 and 0.12, which all together indicates high probability of collision.

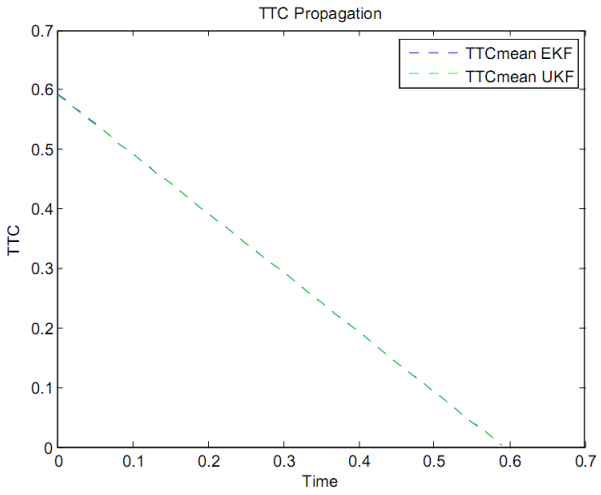


Figure 5.35: TTC of test 1

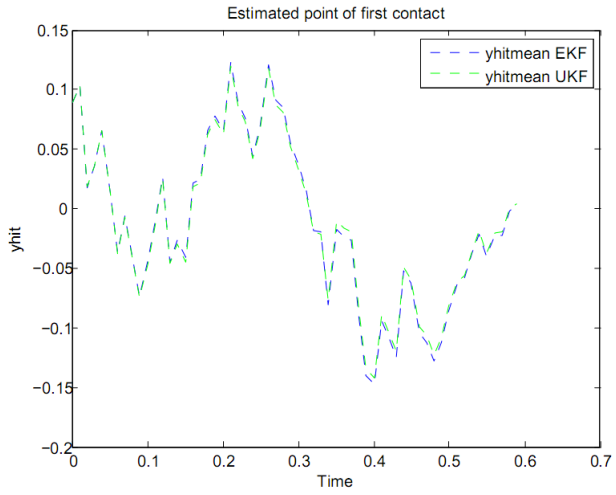


Figure 5.36: Point of First Contact estimation of test 1

Test Scenario 2 Fig 5.37, shows the vehicle collision orientation for Test 2.

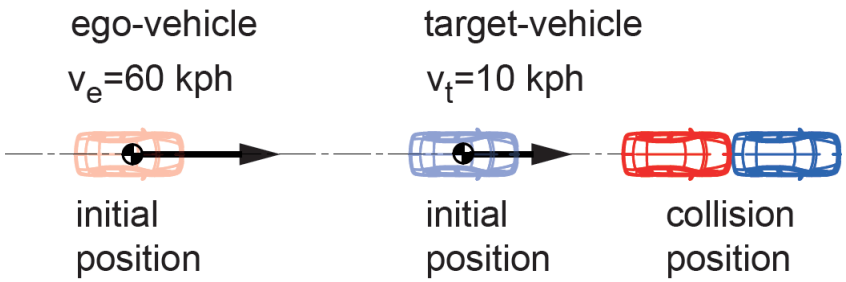


Figure 5.37: The final incidence orientation of test 2

Fig 5.38 shows the trajectory prediction of the last 0.6s before collision.

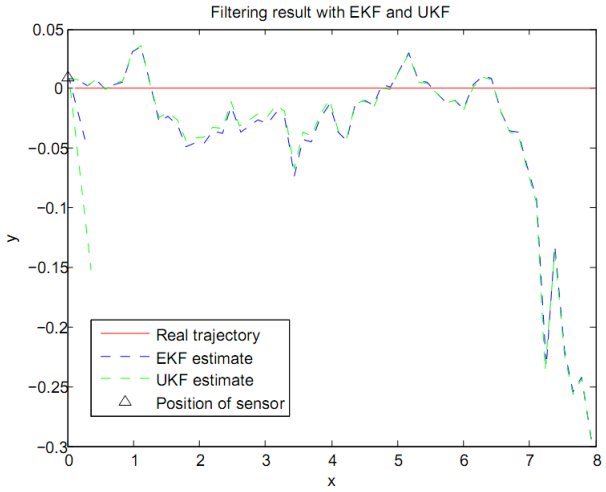


Figure 5.38: Trajectory of test 2

Fig 5.39 and Fig 5.40 show the error involved in the prediction of x and y coordinates respectively.

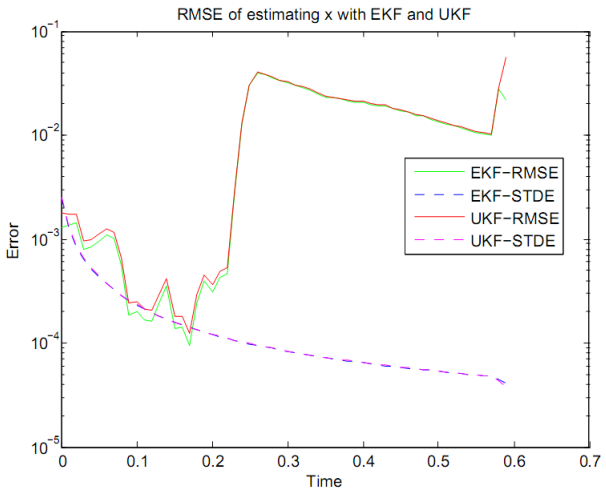


Figure 5.39: RMSE of x and y estimation of test 2

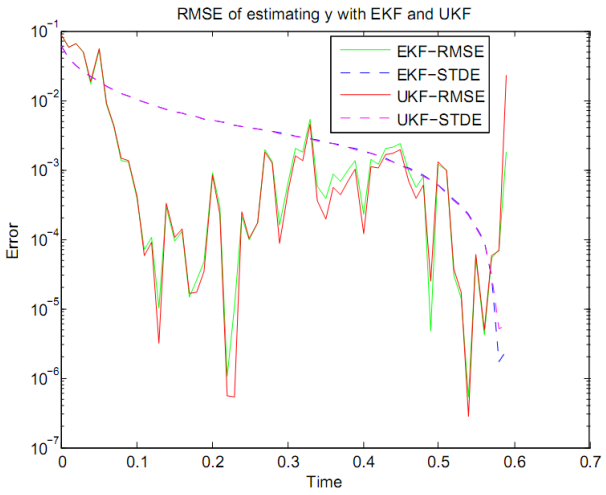


Figure 5.40: RMSE of y estimation of test 2

In Fig 5.41, TTC goes down to zero and in Fig 5.42, the point of first contact lies between -0.3 and 0.05, which all together indicates high probability of collision.

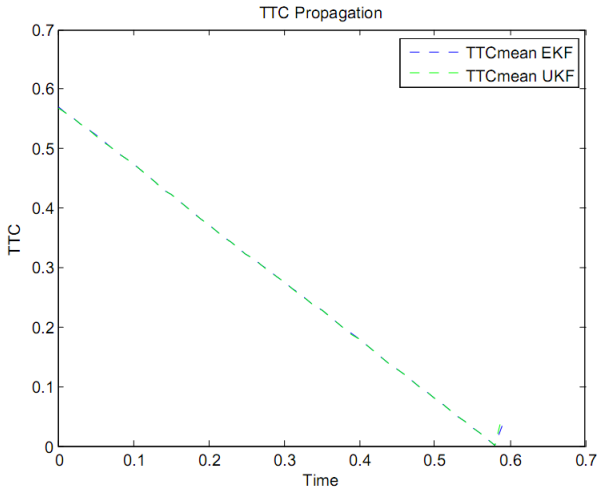


Figure 5.41: TTC of test 2

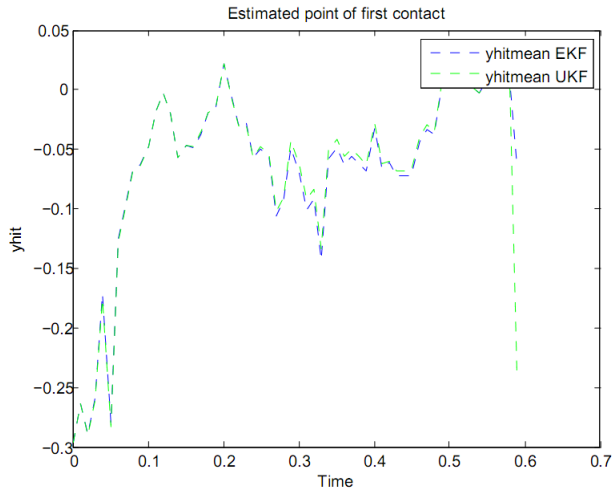


Figure 5.42: Point of First Contact estimation of test 2

Test Scenario 3 Fig 5.43 shows the vehicle collision orientation for Test 3.

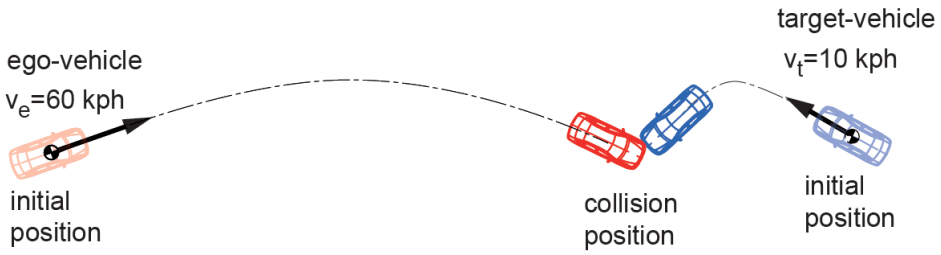


Figure 5.43: The final incidence orientation of test 3

Fig 5.44 shows the trajectory prediction of the last 0.6s before collision.

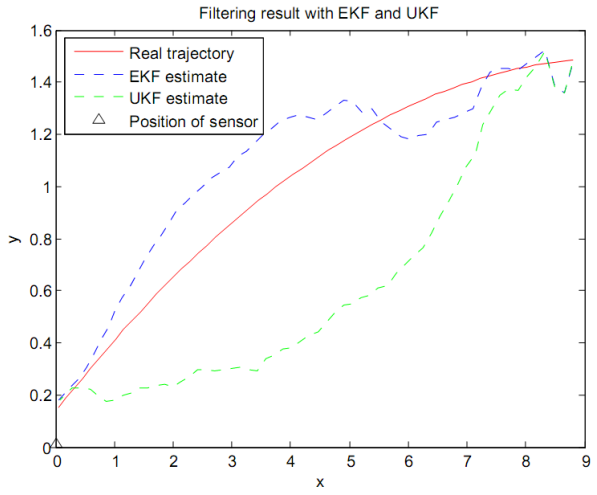


Figure 5.44: Trajectory of test 3

Fig 5.45 and Fig 5.46 show the error involved in the prediction of x and y coordinates respectively.

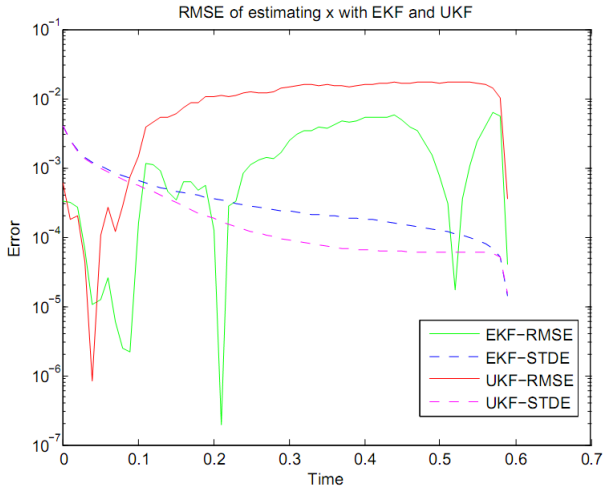


Figure 5.45: RMSE of x and y estimation of test 3

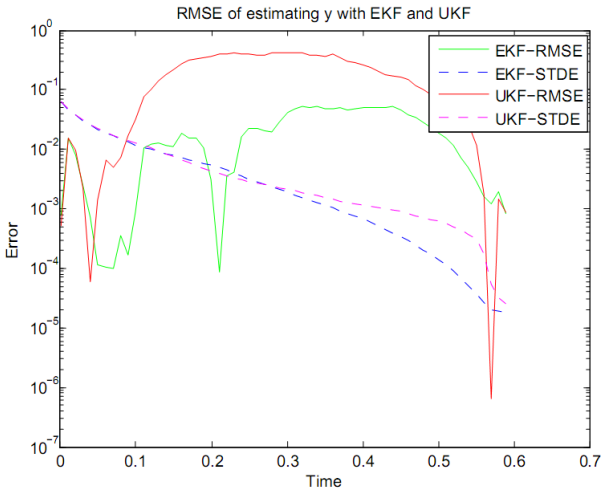


Figure 5.46: RMSE of y estimation of test 3

In Fig 5.47, TTC goes down to zero and in Fig 5.48, the point of first contact lies between -1.2 and 1, which all together indicates high probability of collision.

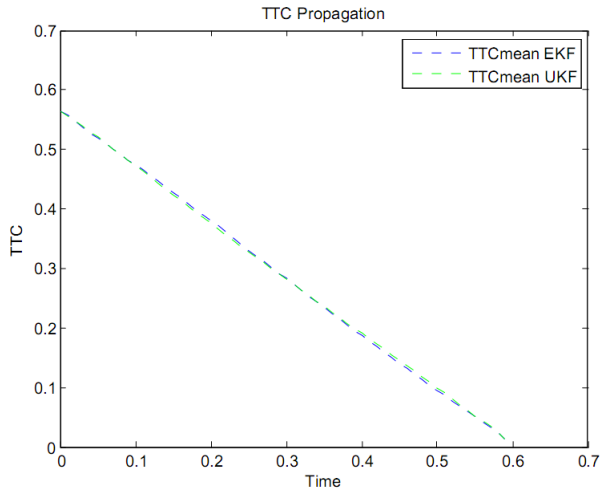


Figure 5.47: TTC of test 3

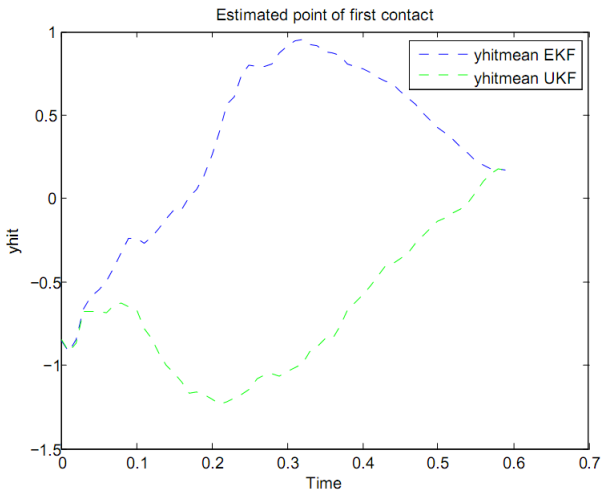


Figure 5.48: Point of First Contact estimation of test 3

Test Scenario 4 Fig 5.49 shows the vehicle collision orientation for Test 4.

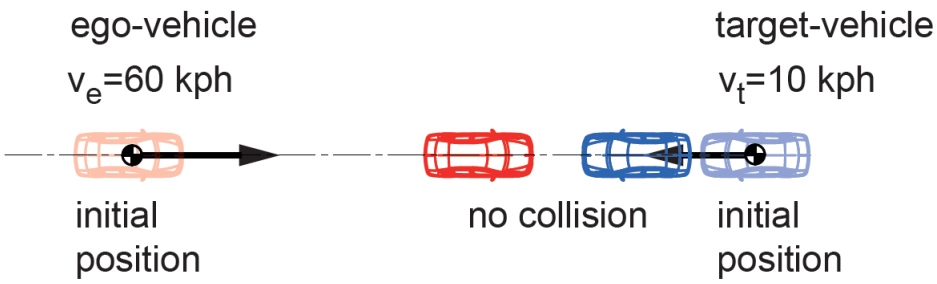


Figure 5.49: The final incidence orientation of test 4

Fig 5.50 shows the trajectory prediction of the last 0.6s before collision.

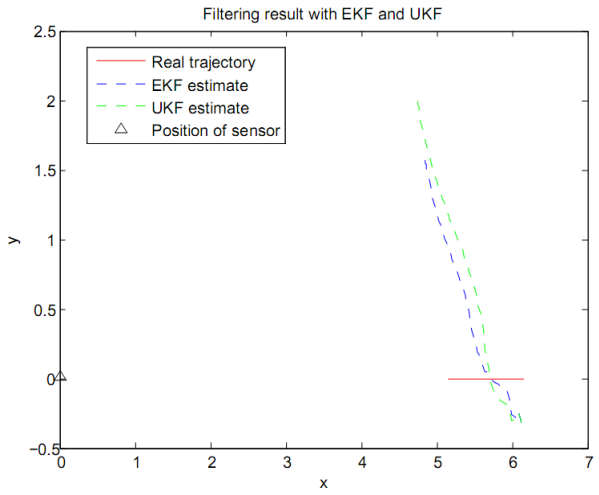


Figure 5.50: Trajectory of test 4

Fig 5.51 and Fig 5.52 show the error involved in the prediction of x and y coordinates respectively.

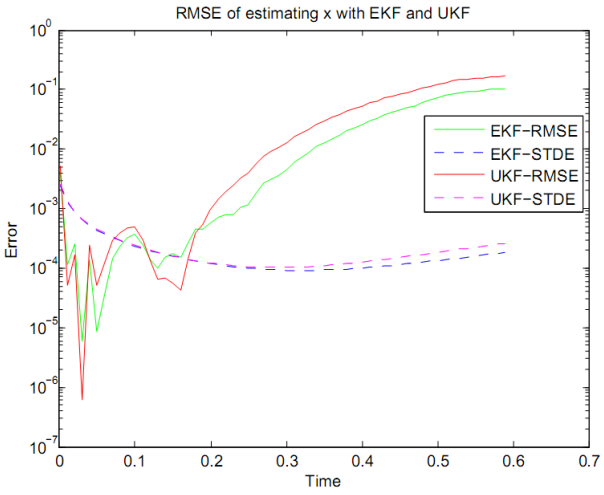


Figure 5.51: RMSE of x and y estimation of test 4

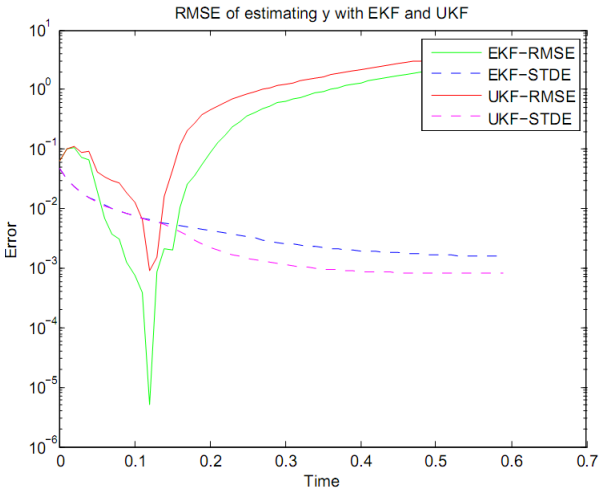


Figure 5.52: RMSE of y estimation of test 4

In Fig 5.53, TTC goes down to zero for some time and starts to go up again and in Fig 5.54, the point of first contact grows from zero to 18, which all together indicates that collision is unlikely.

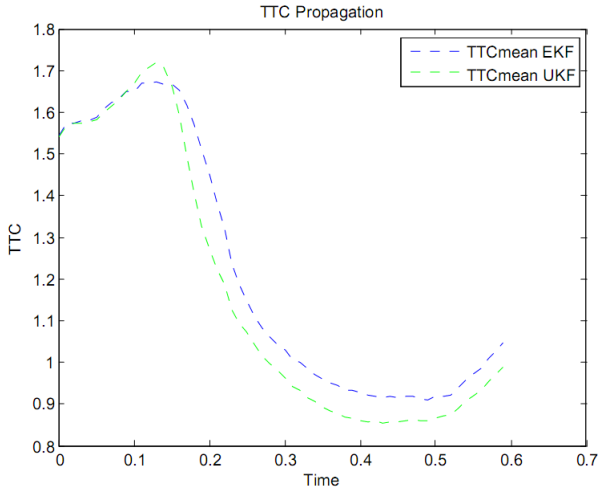


Figure 5.53: TTC of test 4

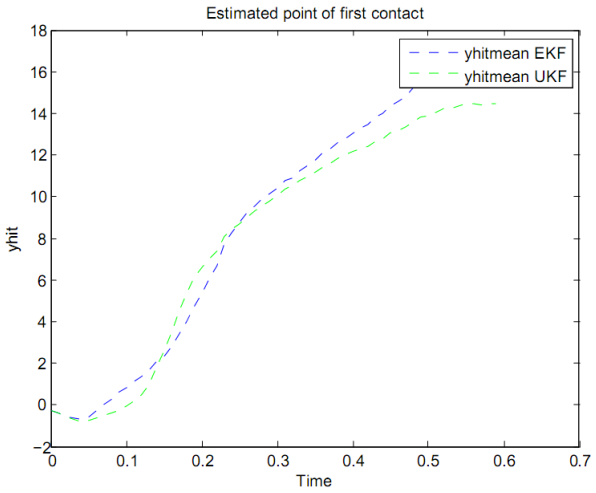


Figure 5.54: Point of First Contact estimation of test 4

Test Scenario 5 Fig 5.55 shows the vehicle collision orientation for Test 5.

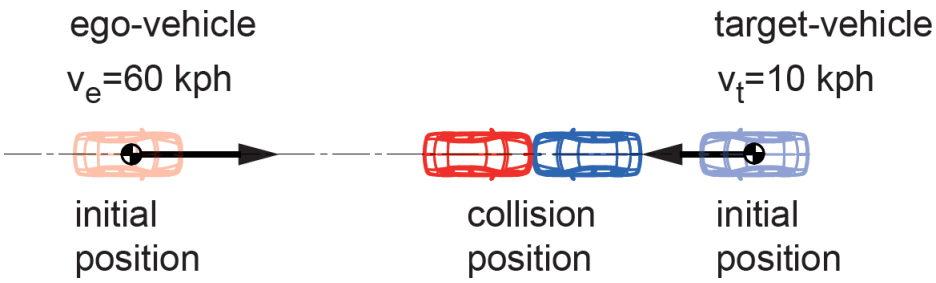


Figure 5.55: The final incidence orientation of test 5

Fig 5.56 shows the trajectory prediction of the last 0.6s before collision.

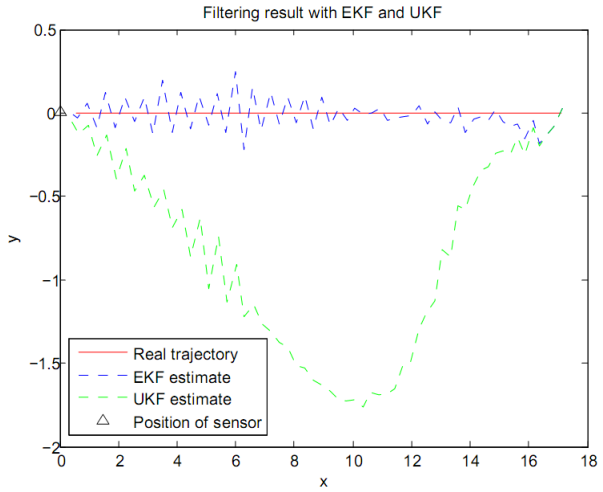


Figure 5.56: Trajectory of test 5

Fig 5.57 and Fig 5.58 show the error involved in the prediction of x and y coordinates respectively.

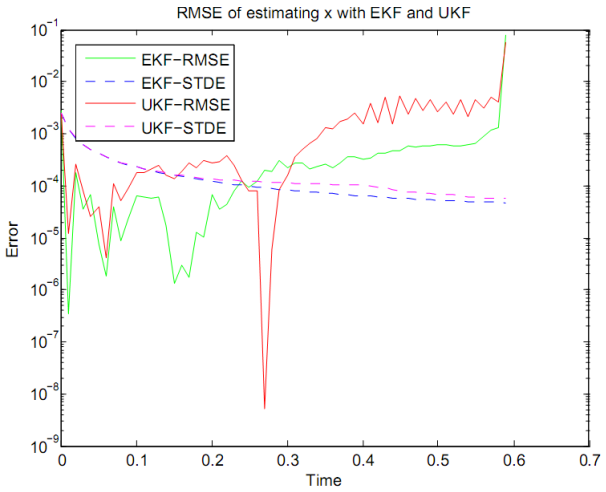


Figure 5.57: RMSE of x and y estimation of test 5

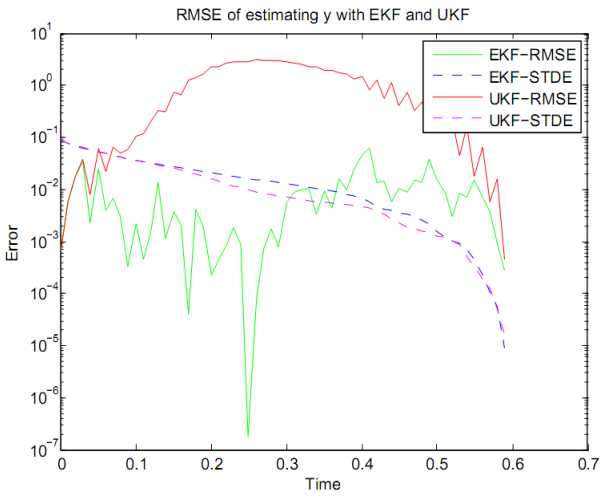


Figure 5.58: RMSE of y estimation of test 5

In Fig 5.59, TTC goes down to zero and in Fig 5.60, the point of first contact lies between -0.5 and 1.8, which all together indicates high probability of collision.

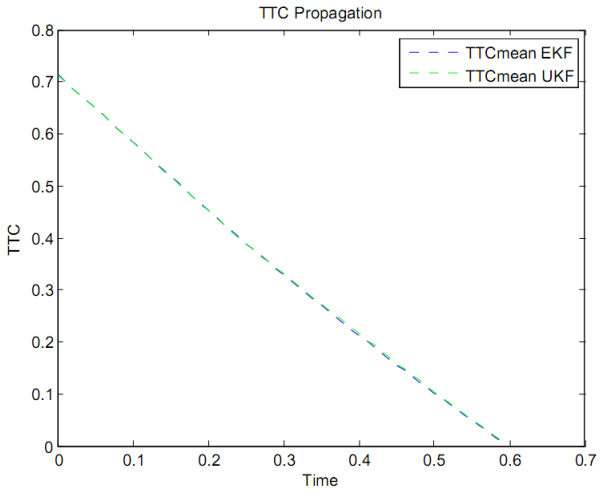


Figure 5.59: TTC of test 5

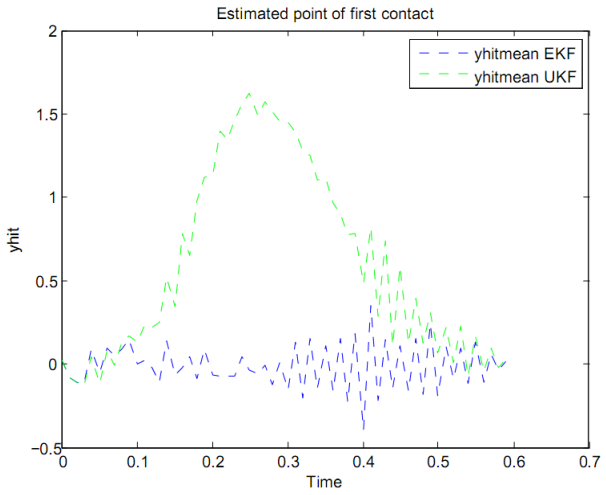


Figure 5.60: Point of First Contact estimation of test 5

Test Scenario 6 Fig 5.61 shows the vehicle collision orientation for Test 6.

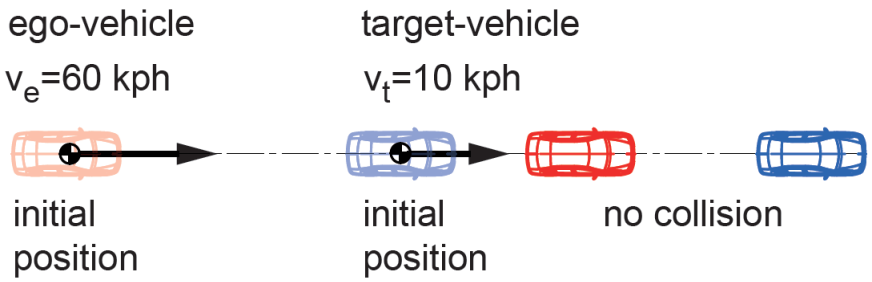


Figure 5.61: The final incidence orientation of test 6

Fig 5.62 shows the trajectory prediction of the last 0.6s before collision.

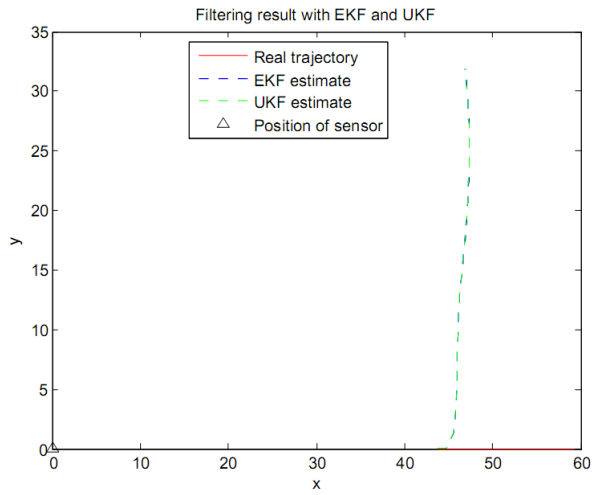


Figure 5.62: Trajectory of test 6

Fig 5.63 and Fig 5.64 show the error involved in the prediction of x and y coordinates respectively.

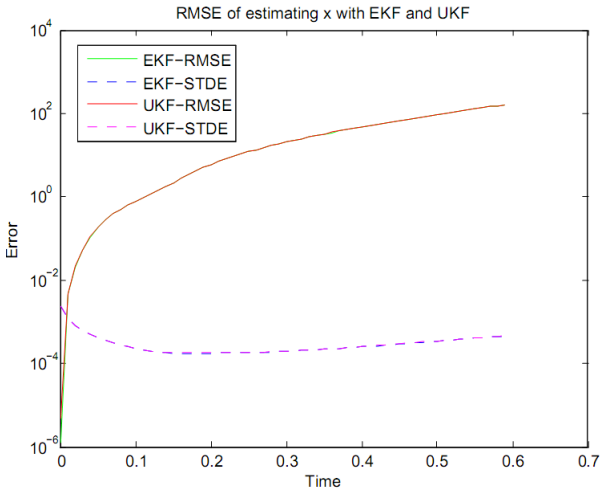


Figure 5.63: RMSE of x and y estimation of test 6

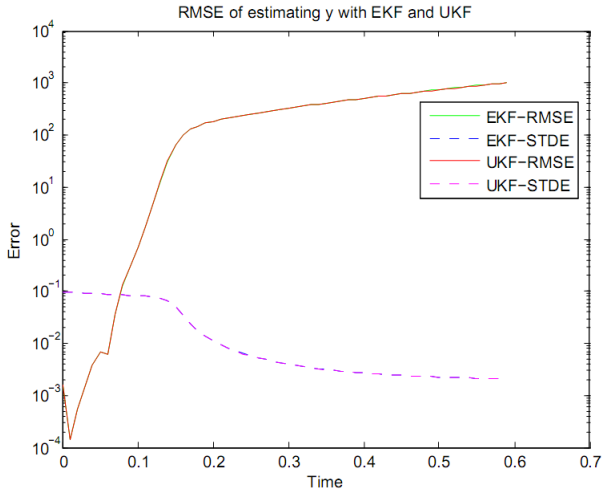


Figure 5.64: RMSE of y estimation of test 6

In Fig 5.65, TTC goes up for a while in the beginning and then starts to drop down and in Fig 5.66, yhit was approximately 0 in the beginning then it went down abruptly, which all together indicates low probability of collision.

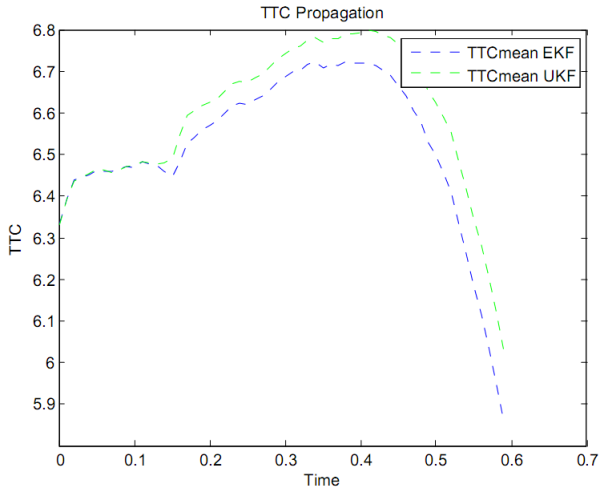


Figure 5.65: TTC of test 6

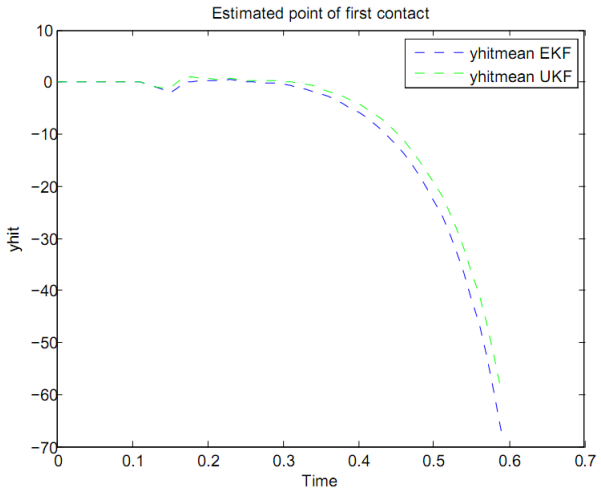


Figure 5.66: Point of First Contact estimation of test 6

Test Scenario 7 Fig 5.67 shows the vehicle collision orientation for Test 7.

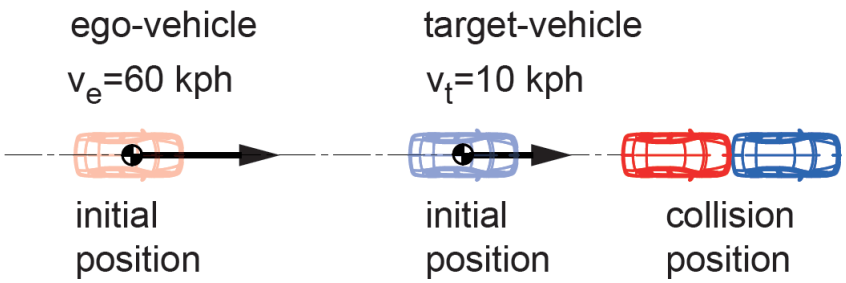


Figure 5.67: The final incidence orientation of test 7

Fig 5.68 shows the trajectory prediction of the last 0.6s before collision.

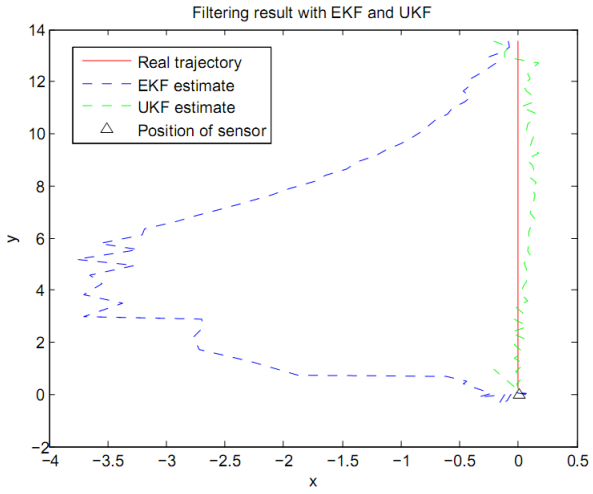


Figure 5.68: Trajectory of test 7

Fig 5.69 and Fig 5.70 show the error involved in the prediction of x and y coordinates respectively.

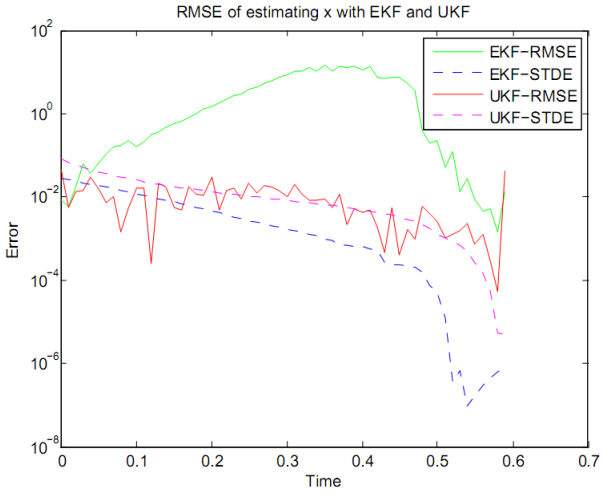


Figure 5.69: RMSE of x and y estimation of test 7

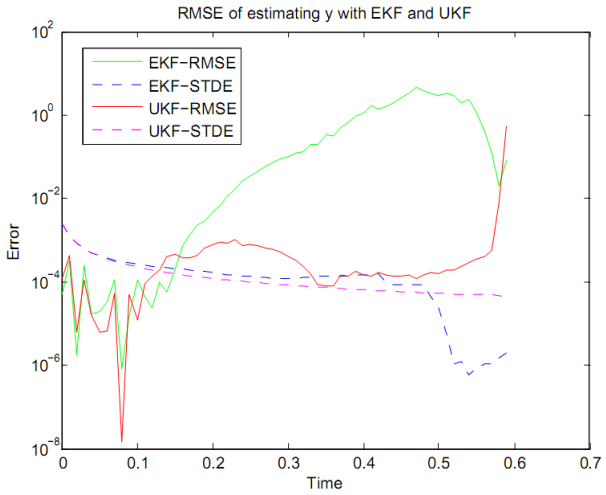


Figure 5.70: RMSE of y estimation of test 7

In Fig 5.71, TTC goes down to zero and in Fig 5.72, yhit has the mean value of 0 although some divergence problem is observed, which all together indicates high probability of collision.

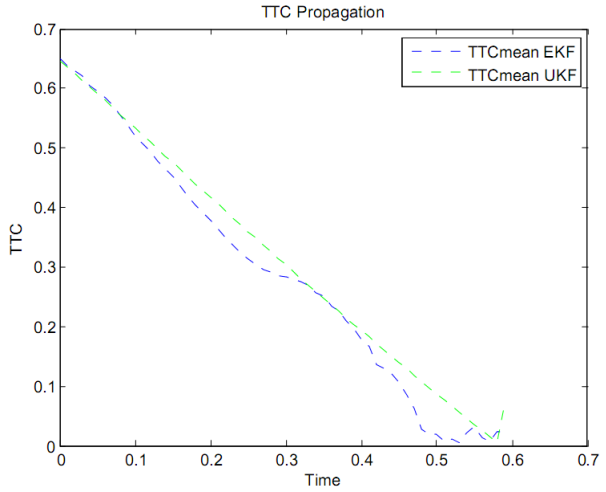


Figure 5.71: TTC of test 7

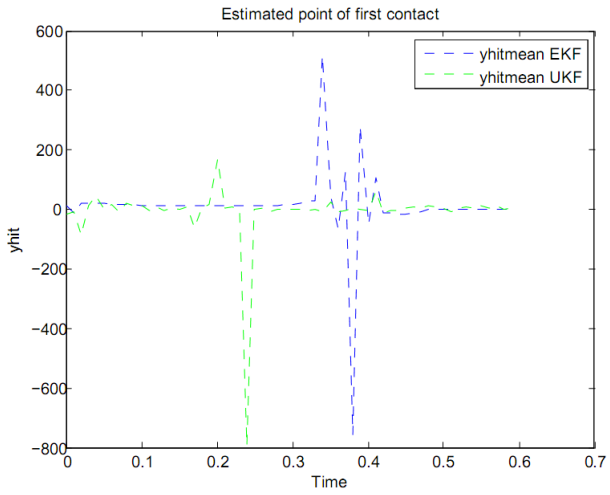


Figure 5.72: Point of First Contact estimation of test 7

Test Scenario 8 Fig 5.73 shows the vehicle collision orientation for Test 8.

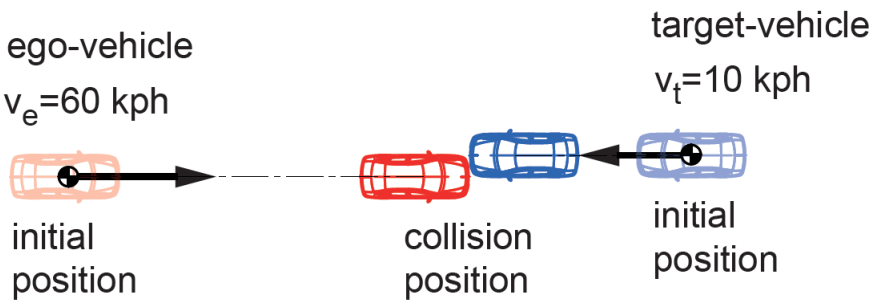


Figure 5.73: The final incidence orientation of test 8

Fig 5.74 shows the trajectory prediction of the last 0.6s before collision.

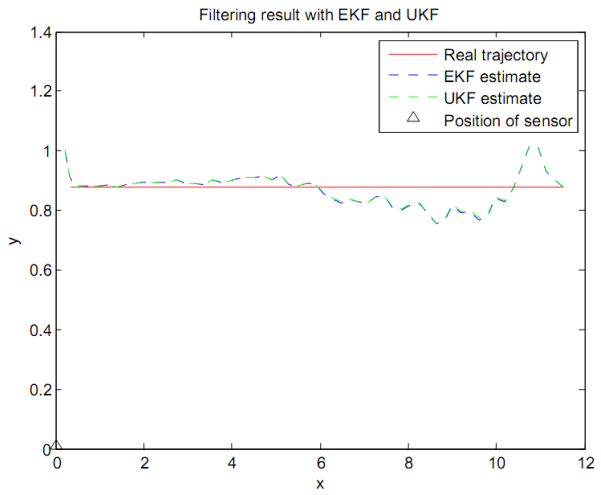


Figure 5.74: Trajectory of test 8

Fig 5.75 and Fig 5.76 show the error involved in the prediction of x and y coordinates respectively.

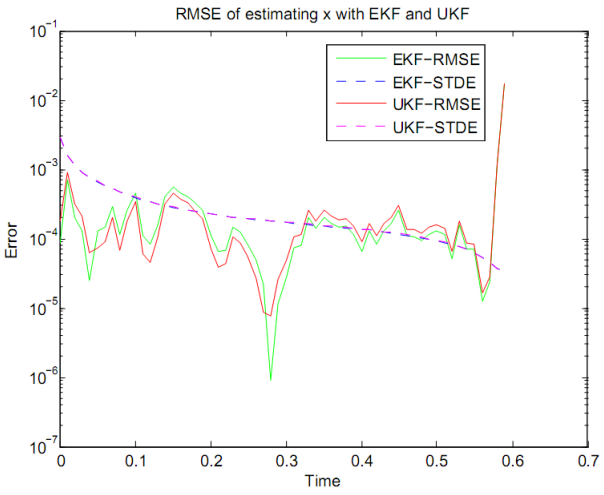


Figure 5.75: RMSE of x and y estimation of test 8

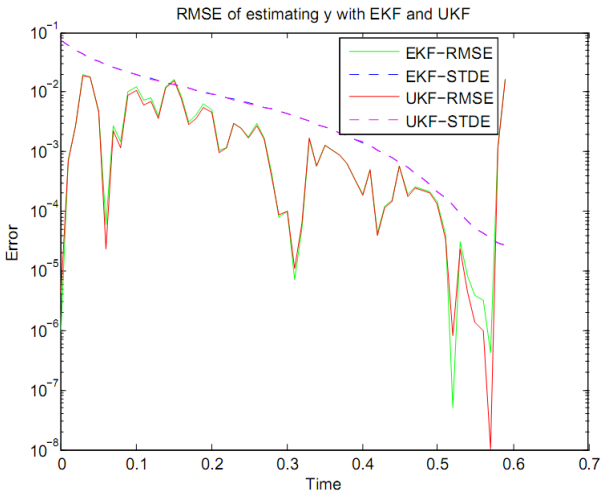


Figure 5.76: RMSE of y estimation of test 8

In Fig 5.77, TTC goes down below 0.1 and in Fig 5.78, yhit lies in between 0.75 and 1.05, which all together indicates high probability of collision.

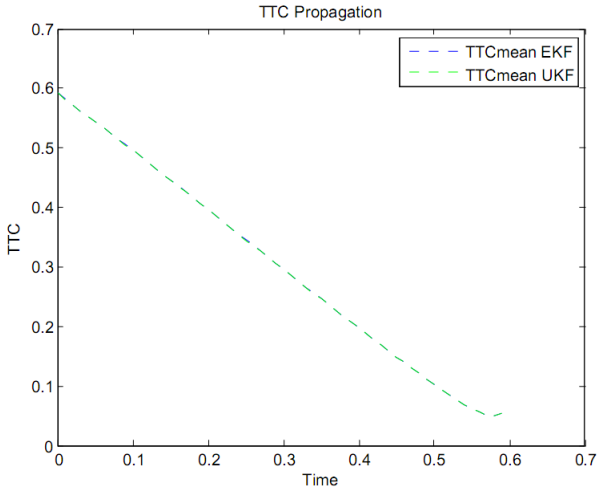


Figure 5.77: TTC of test 8

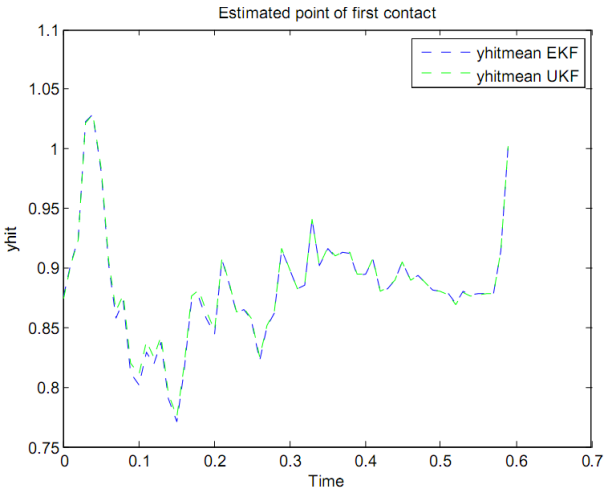


Figure 5.78: Point of First Contact estimation of test 8

Test Scenario 9 Fig 5.79 shows the vehicle collision orientation for Test 9.

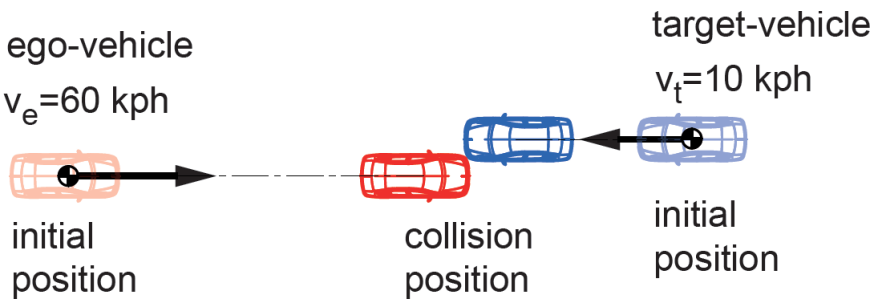


Figure 5.79: The final incidence orientation of test 9

Fig 5.80 shows the trajectory prediction of the last 0.6s before collision.

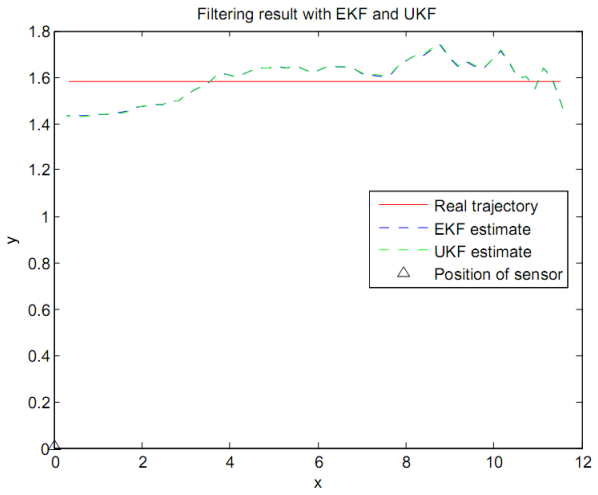


Figure 5.80: Trajectory of test 9

Fig 5.81 and Fig 5.82 show the error involved in the prediction of x and y coordinates respectively.

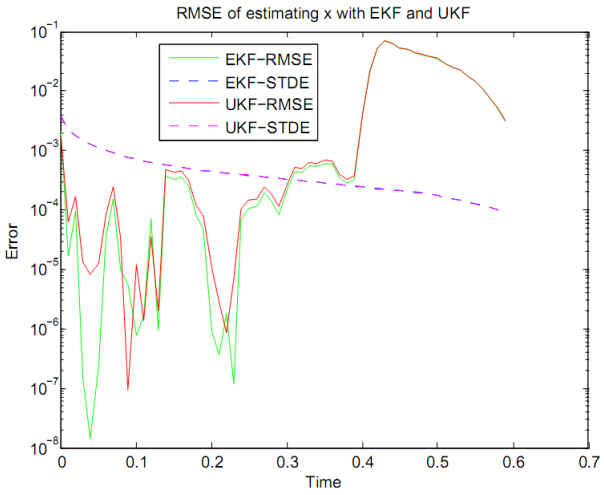


Figure 5.81: RMSE of x and y estimation of test 9

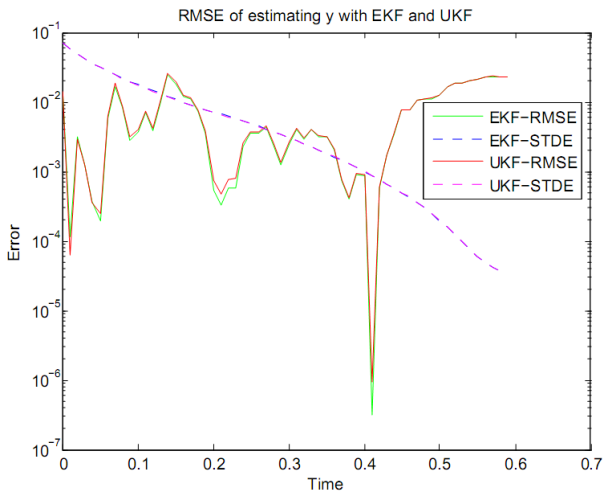


Figure 5.82: RMSE of y estimation of test 9

In Fig 5.83, TTC goes down below 0.1 and in Fig 5.84, yhit lies in between 1.4 and 1.85, which all together indicates moderate probability of collision.

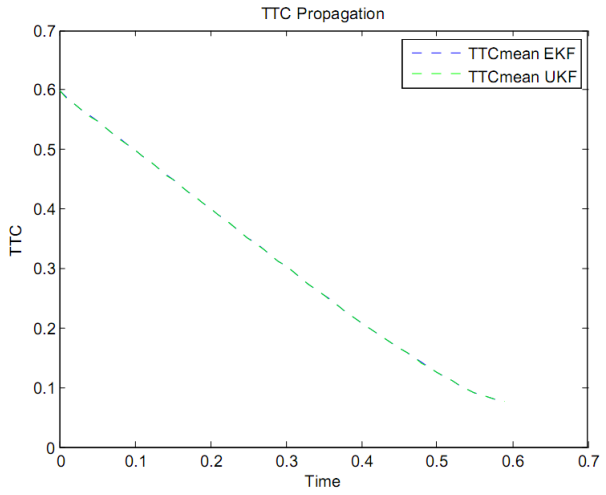


Figure 5.83: TTC of test 9

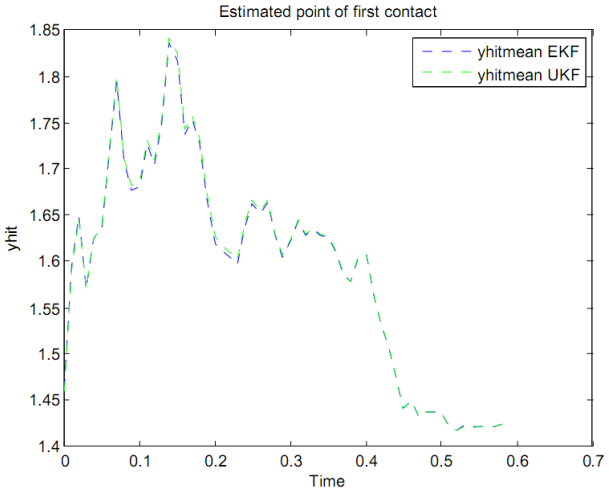


Figure 5.84: Point of First Contact estimation of test 9

Test Scenario10 Fig 5.85 shows the vehicle collision orientation for Test 10.

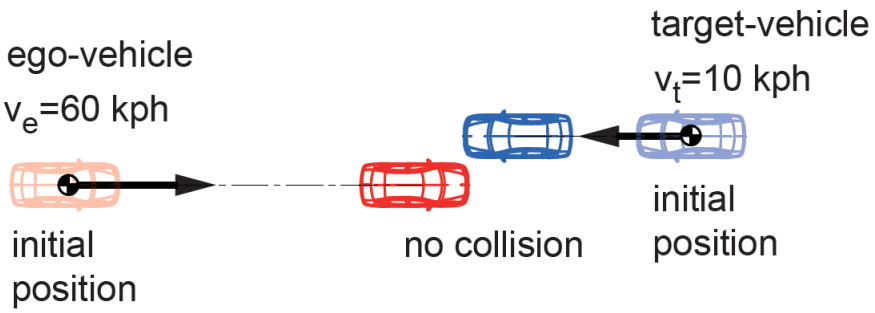


Figure 5.85: The final incidence orientation of test 10

Fig 5.86 shows the trajectory prediction of the last 0.6s before collision.

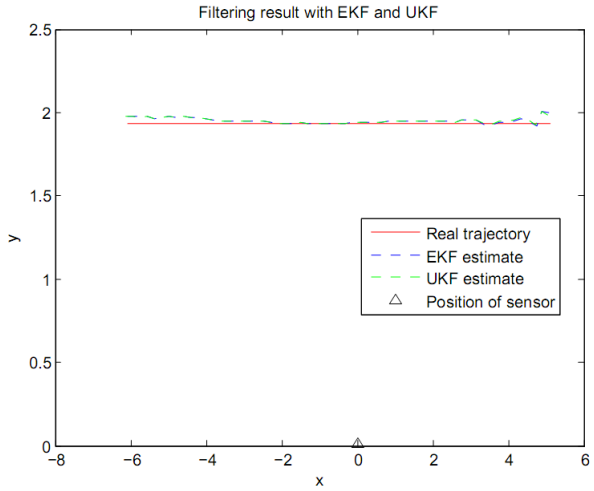


Figure 5.86: Trajectory of test 10

Fig 5.87 and Fig 5.88 show the error involved in the prediction of x and y coordinates respectively.

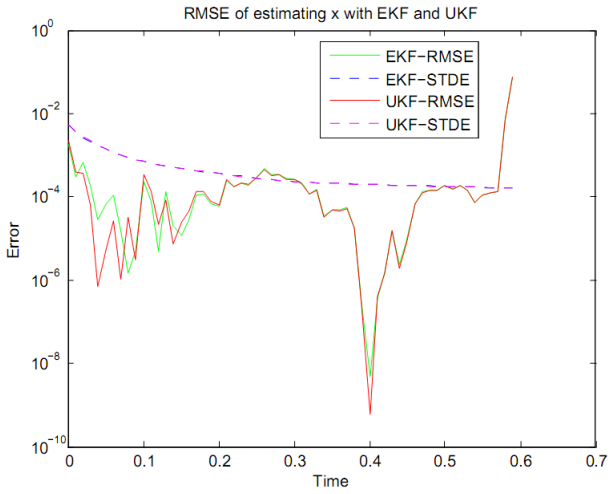


Figure 5.87: RMSE of x and y estimation of test 10

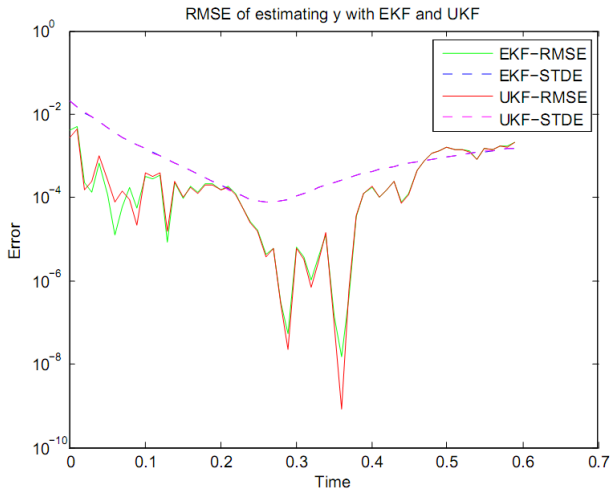


Figure 5.88: RMSE of y estimation of test 10

In Fig 5.89, TTC goes down in the beginning and then it starts to go up. and in Fig 5.90, yhit lies in between 1.88 and 2, which all together indicates that collision is unlikely.

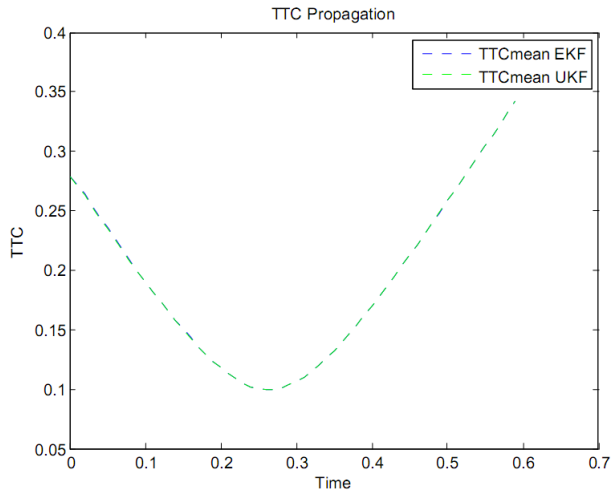


Figure 5.89: TTC of test 10

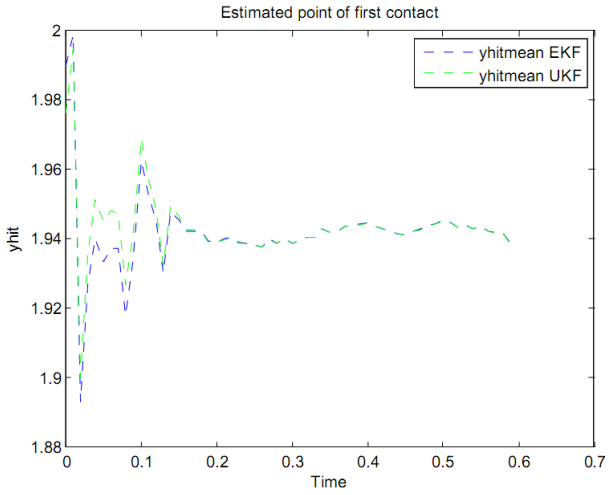


Figure 5.90: Point of First Contact estimation of test 10

Test Scenario11 Fig 5.91 shows the vehicle collision orientation for Test 11.

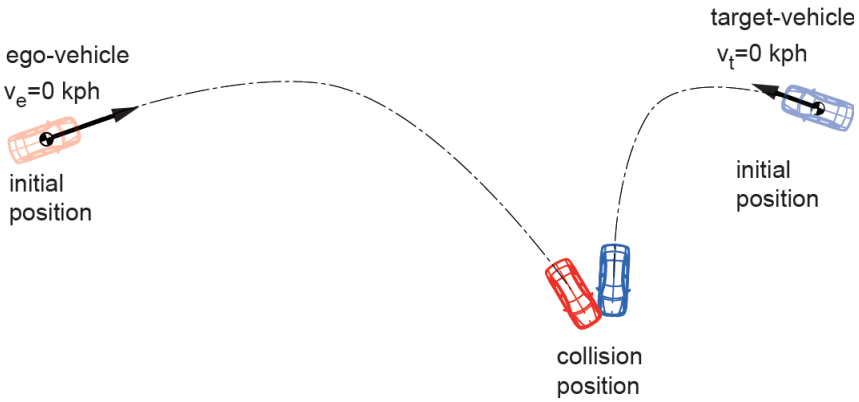


Figure 5.91: The final incidence orientation of test 11

Fig 5.92 shows the trajectory prediction of the last 0.6s before collision.

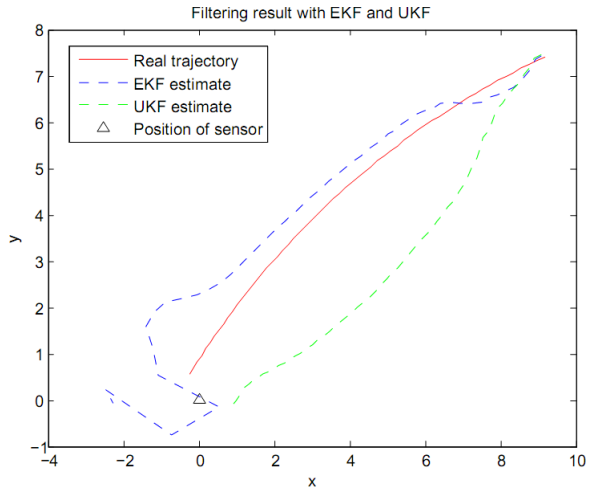


Figure 5.92: Trajectory of test 11

Fig 5.93 and Fig 5.94 show the error involved in the prediction of x and y coordinates respectively.

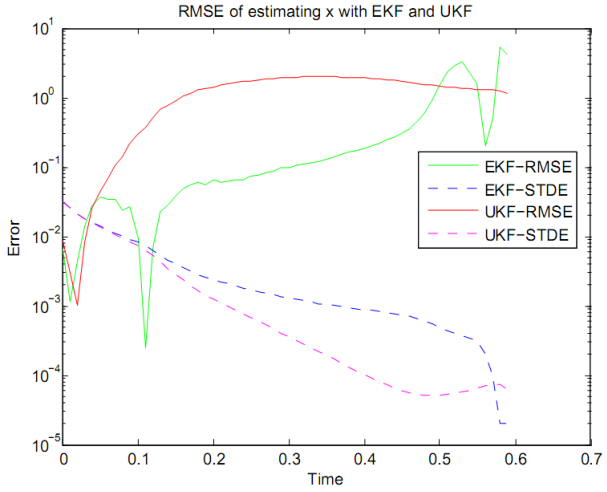


Figure 5.93: RMSE of x and y estimation of test 11

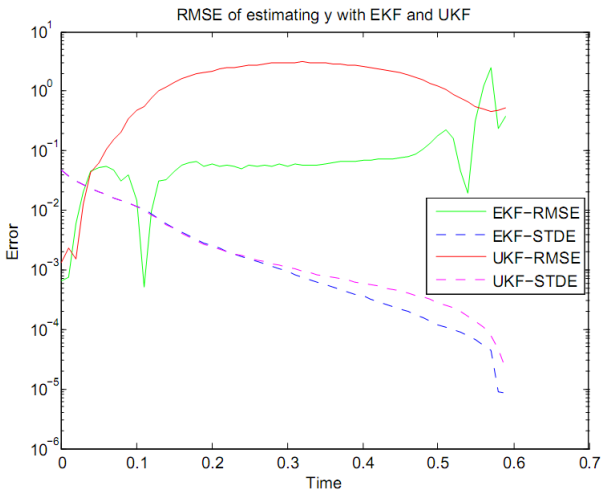


Figure 5.94: RMSE of y estimation of test 11

In Fig 5.95, TTC goes down to zero and in Fig 5.96, yhit has the mean value of 0 although some divergence problem is observed, which all together indicates high probability of collision.

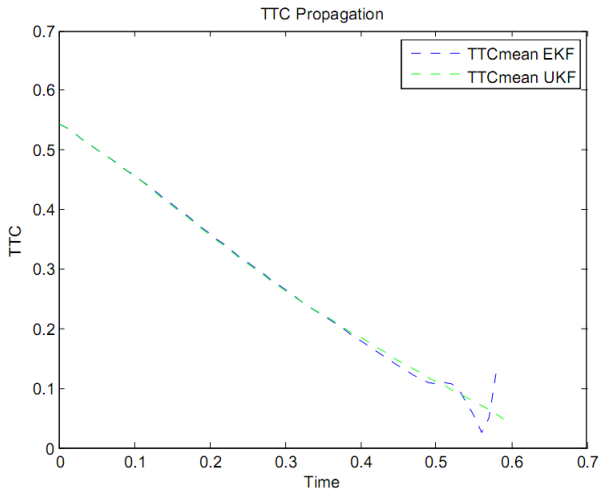


Figure 5.95: TTC of test 11

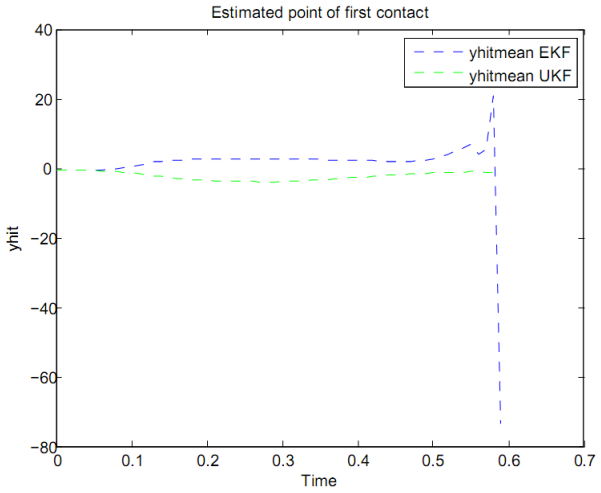


Figure 5.96: Point of First Contact estimation of test 11

Test Scenario12 Fig 5.97 shows the vehicle collision orientation for Test 12.

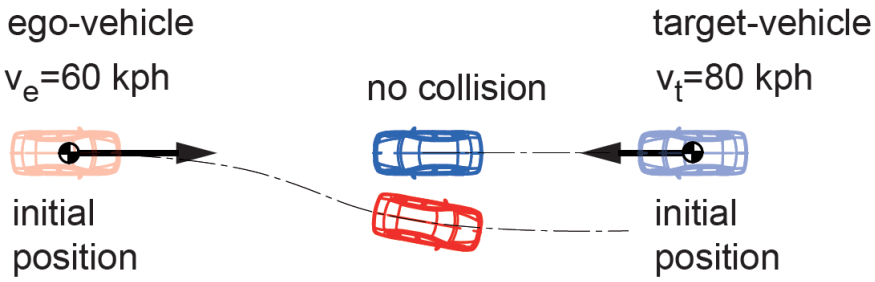


Figure 5.97: The final incidence orientation of test 12

Fig 5.98, shows the trajectory prediction of the last 0.6s before collision.

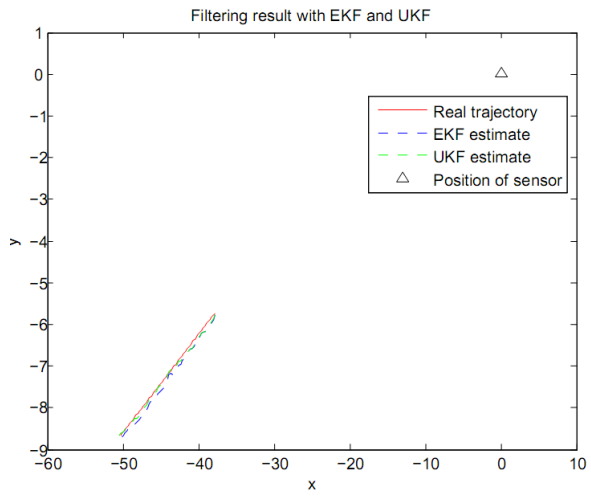


Figure 5.98: Trajectory of test 12

Fig 5.99 and Fig 5.100 show the error involved in the prediction of x and y coordinates respectively.

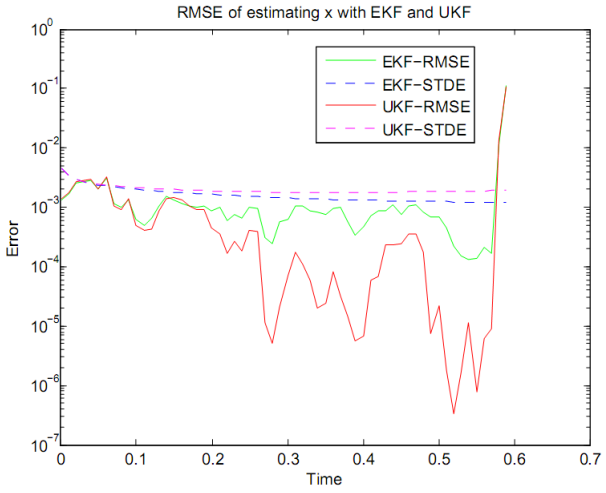


Figure 5.99: RMSE of x and y estimation of test 12

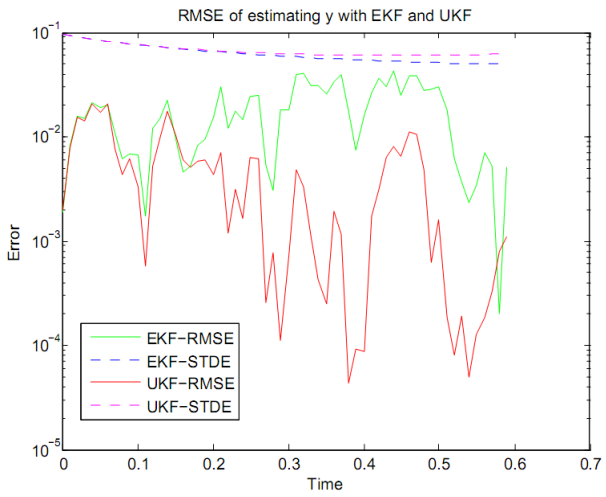


Figure 5.100: RMSE of y estimation of test 12

In Fig 5.101, TTC goes up steadily and in Fig 5.102, yhit lies in the range of 0 and 3.2 in. This tells us that there is a very low probability of collision.

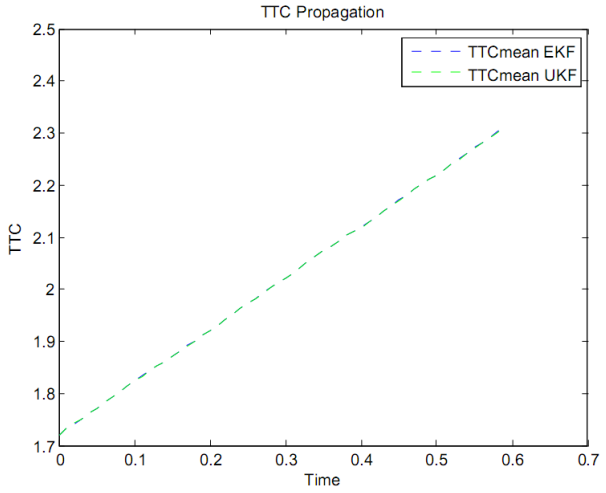


Figure 5.101: TTC of test 12

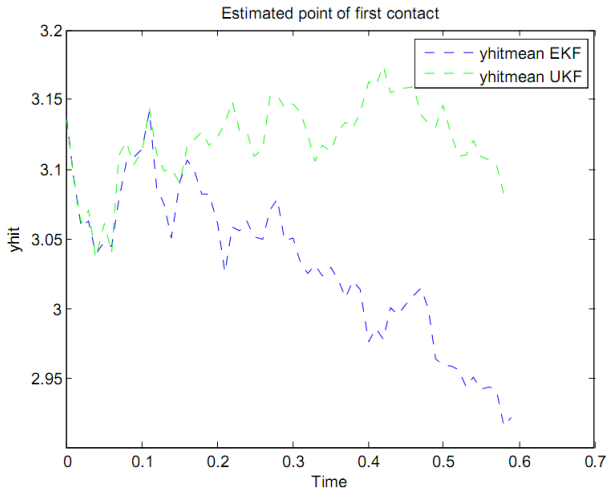


Figure 5.102: Point of First Contact estimation of test 12

Test Scenario13 Fig 5.103 shows the vehicle collision orientation for Test 13.

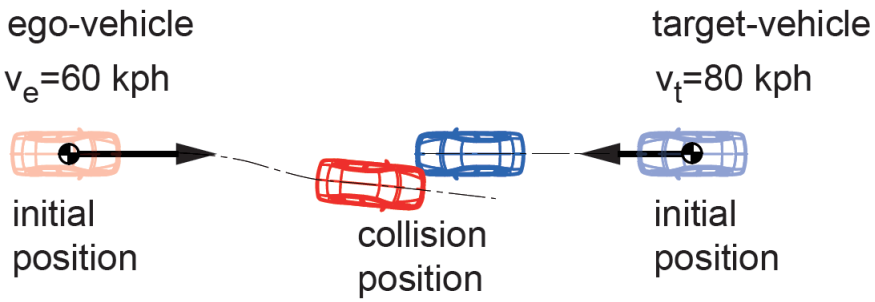


Figure 5.103: The final incidence orientation of test 13

Fig 5.104 shows the trajectory prediction of the last 0.6s before collision.

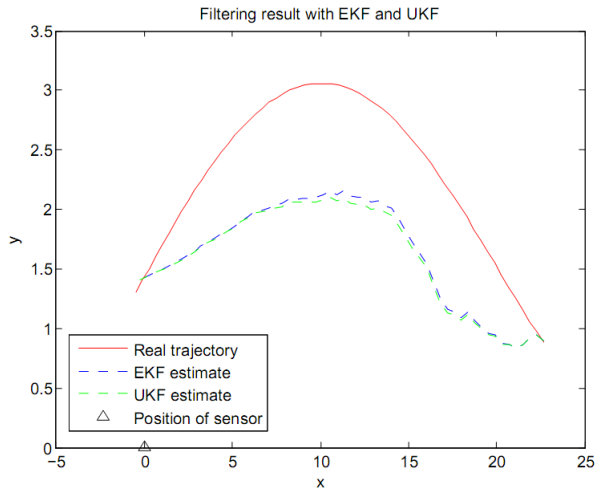


Figure 5.104: Trajectory of test 13

Fig 5.105 and Fig 5.106 show the error involved in the prediction of x and y coordinates respectively.

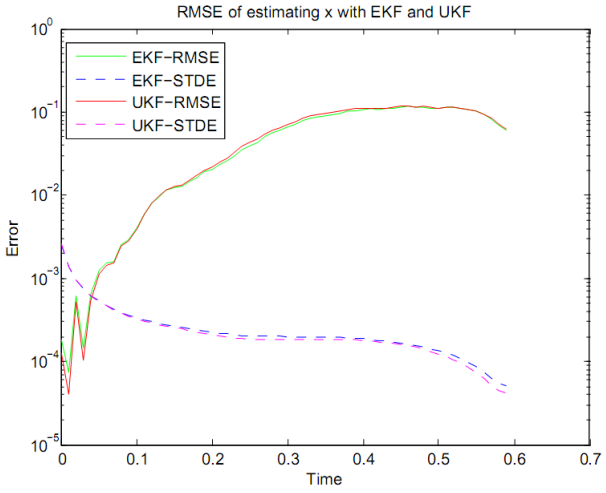


Figure 5.105: RMSE of x and y estimation of test 13

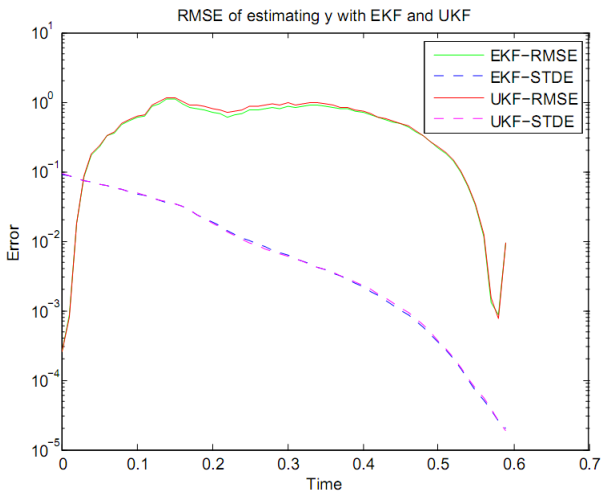


Figure 5.106: RMSE of y estimation of test 13

In Fig 5.107, TTC goes down to a value of 0 steadily and in Fig 5.108, yhit lies in the range of 0.4 and 2.2. This tells us that there is some probability of collision.

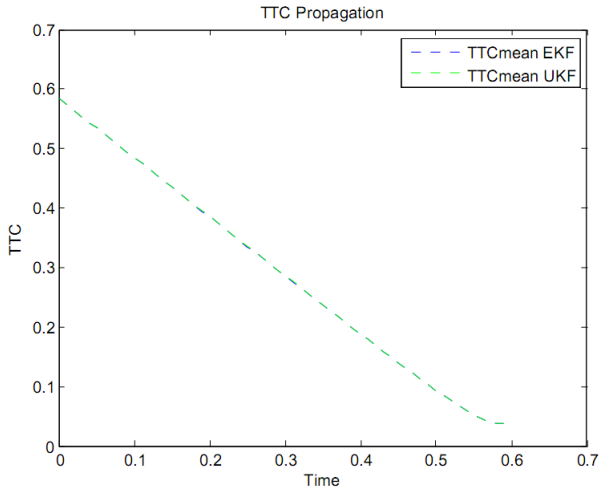


Figure 5.107: TTC of test 13

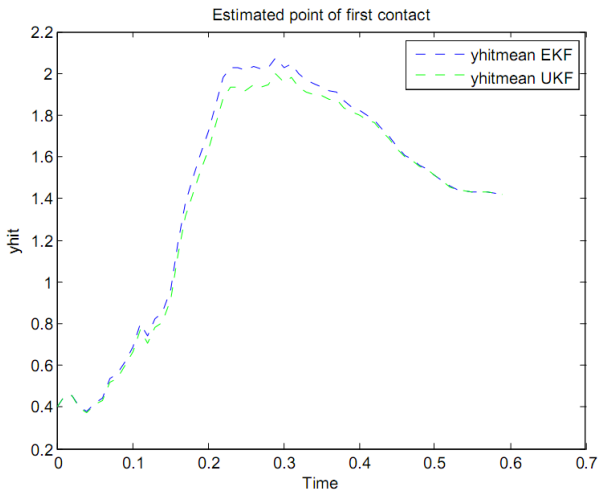


Figure 5.108: Point of First Contact estimation of test 13

Test Scenario14 Fig 5.109 shows the vehicle collision orientation for Test 14.

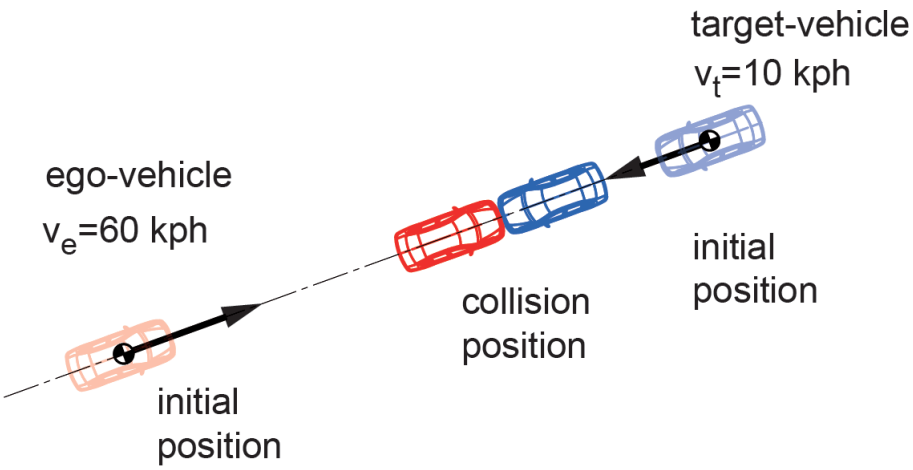


Figure 5.109: The final incidence orientation of test 14

Fig 5.110 shows the trajectory prediction of the last 0.6s before collision.

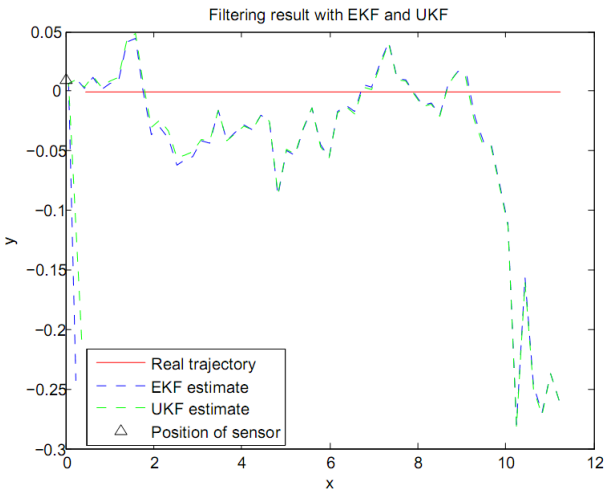


Figure 5.110: Trajectory of test 14

Fig 5.111 and Fig 5.112 show the error involved in the prediction of x and y coordinates respectively.

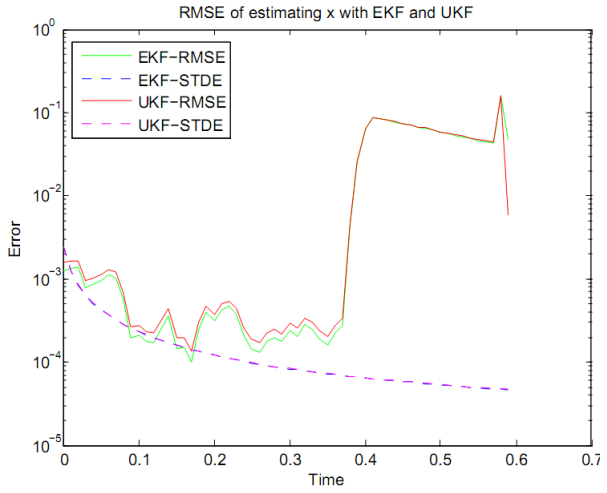


Figure 5.111: RMSE of x and y estimation of test 14

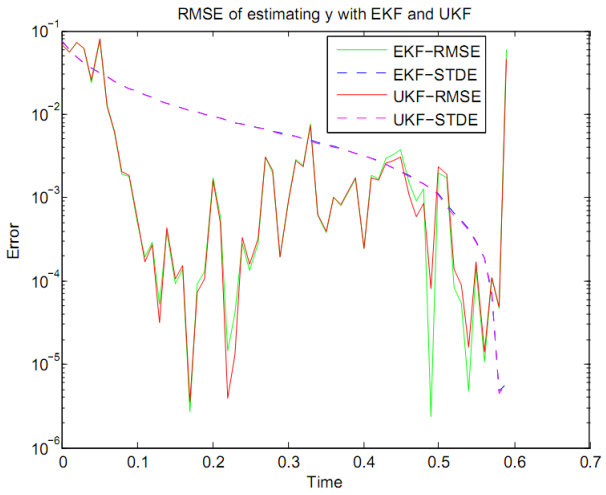


Figure 5.112: RMSE of y estimation of test 14

In Fig 5.113, TTC goes down to a value of 0 steadily and in Fig 5.114, yhit lies in the range of -0.35 and 0.05. This tells us that there is a high probability of collision.

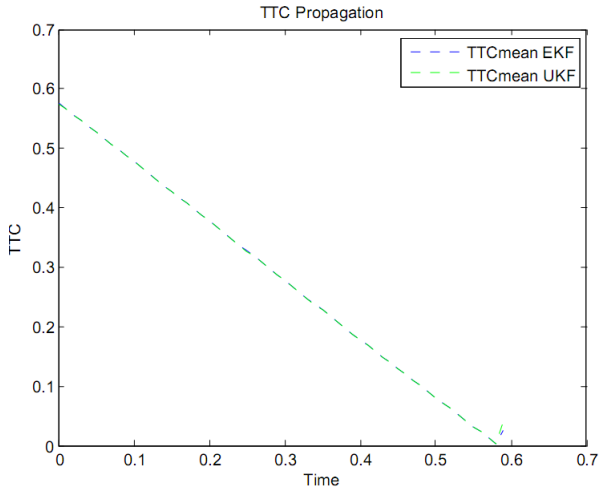


Figure 5.113: TTC of test 14

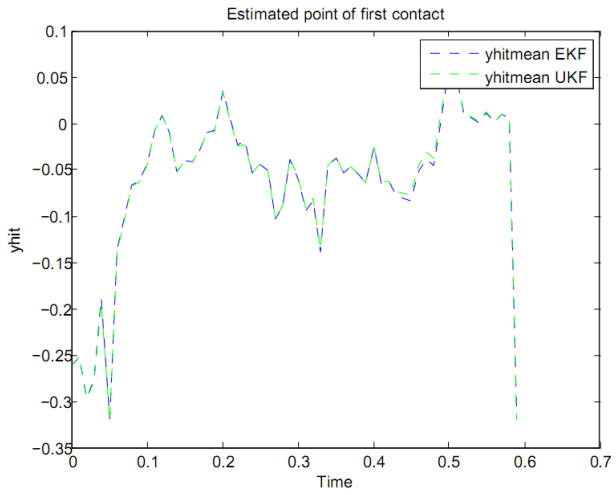


Figure 5.114: Point of First Contact estimation of test 14

Test Scenario15 Fig 5.115 shows the vehicle collision orientation for Test 15.

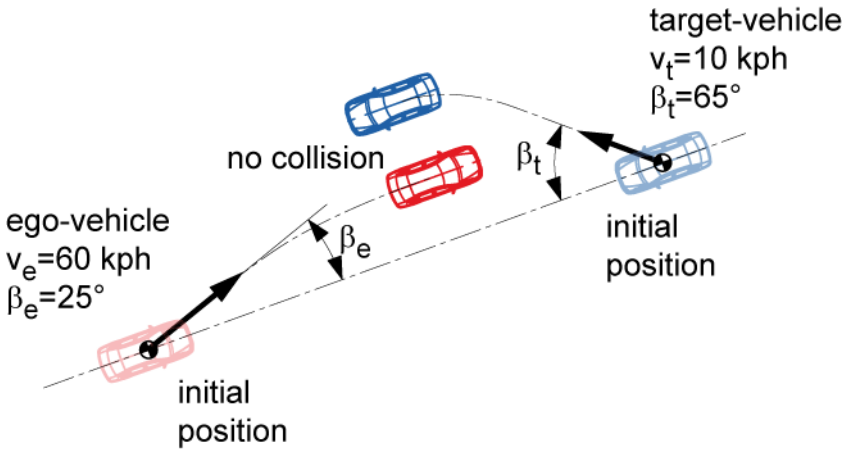


Figure 5.115: The initial and final incidence orientation of test 15

Fig 5.116 shows the trajectory prediction of the last 0.6s before collision.

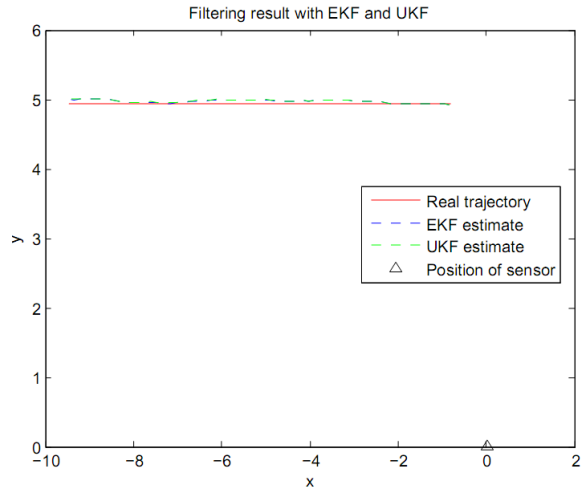


Figure 5.116: Trajectory of test 15

Fig 5.117 and Fig 5.118 show the error involved in the prediction of x and y coordinates respectively.

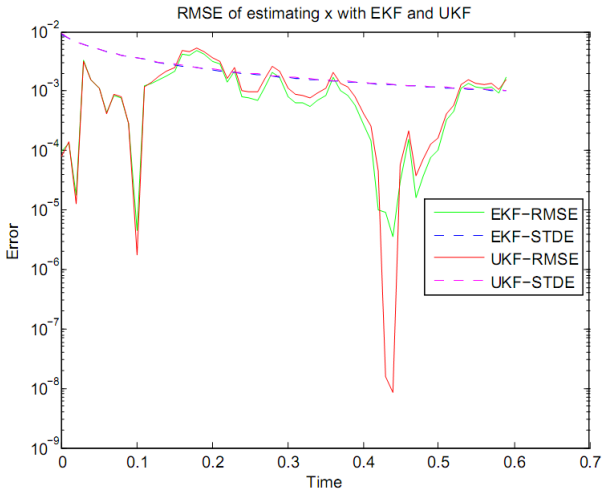


Figure 5.117: RMSE of x and y estimation of test 15

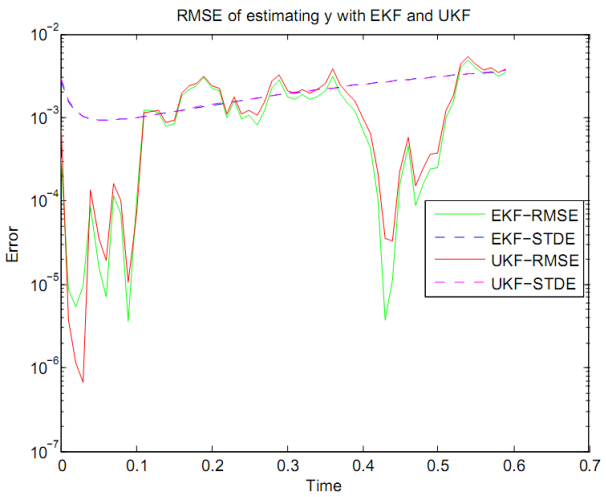


Figure 5.118: RMSE of y estimation of test 15

In Fig 5.119, TTC goes up steadily and in Fig 5.120, yhit lies in the range of 4.915 and 4.96. This tells us that collision will not occur.

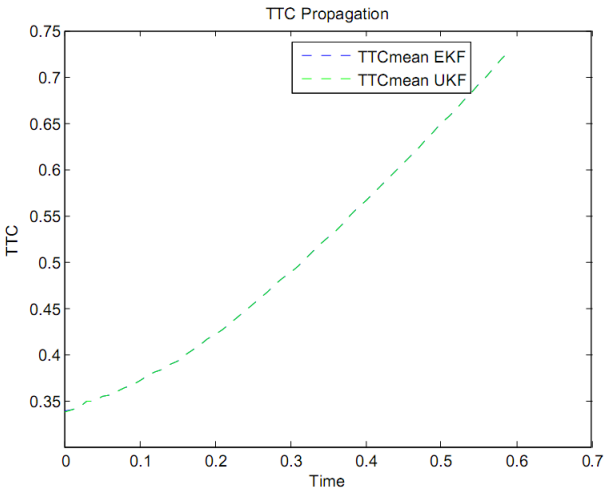


Figure 5.119: TTC of test 15

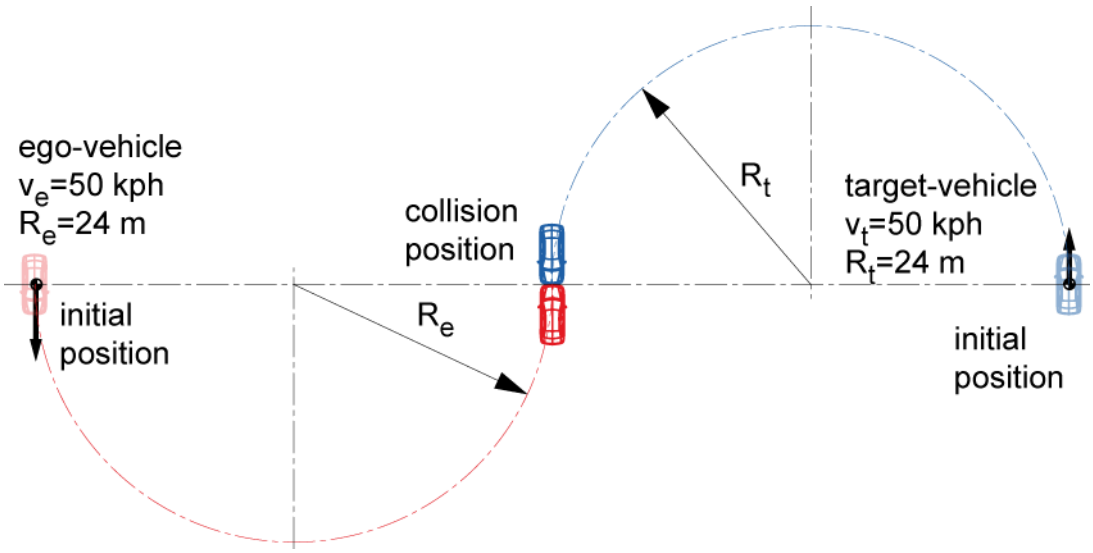


Figure 5.121: The initial and final incidence orientation of test 16

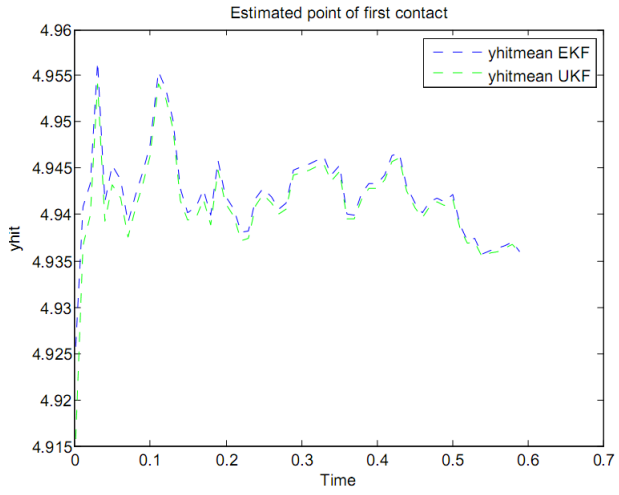


Figure 5.120: Point of First Contact estimation of test 15

Test Scenario16 Fig 5.121 shows the vehicle collision orientation for Test 16. Fig 5.122 shows the trajectory prediction of the last 0.6s before collision.

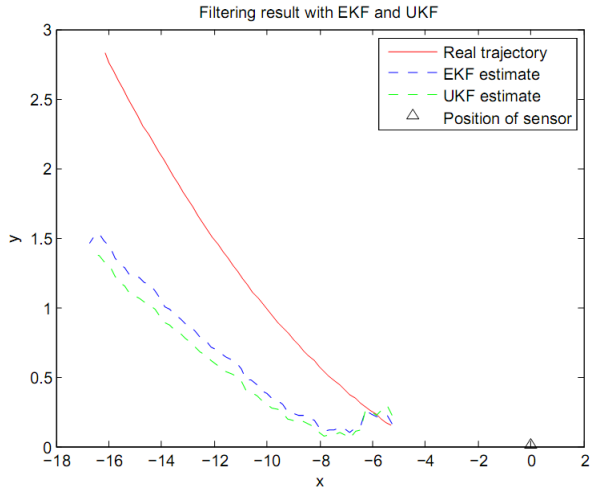


Figure 5.122: Trajectory of test 16

Fig 5.123 and Fig 5.124 show the error involved in the prediction of x and y coordinates respectively.

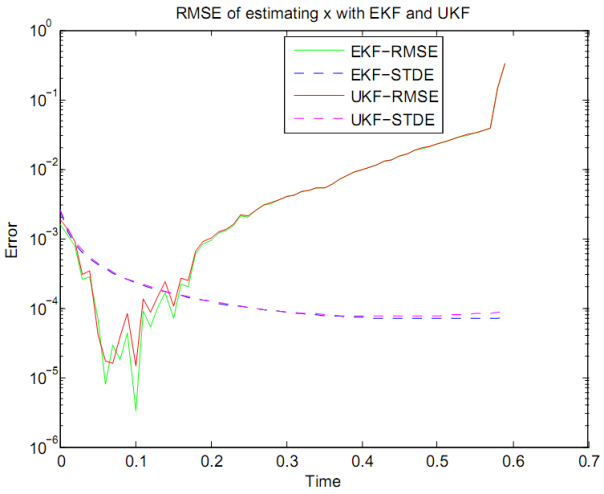


Figure 5.123: RMSE of x and y estimation of test 16

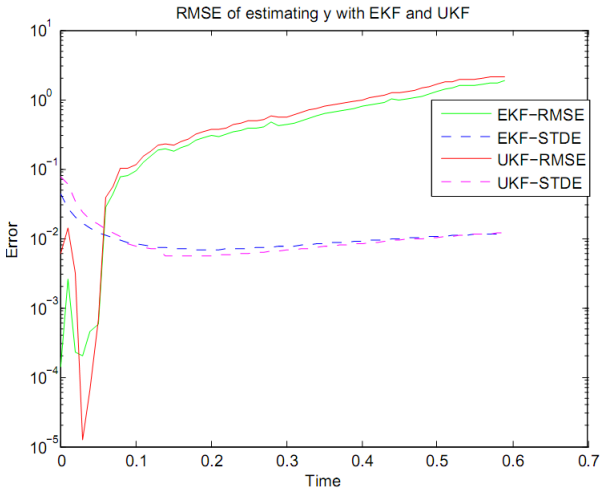


Figure 5.124: RMSE of y estimation of test 16

In Fig 5.125, TTC goes up steadily and in Fig 5.125, yhit lies in the range of -0.25 and 0.2. This tells us that collision will not occur.

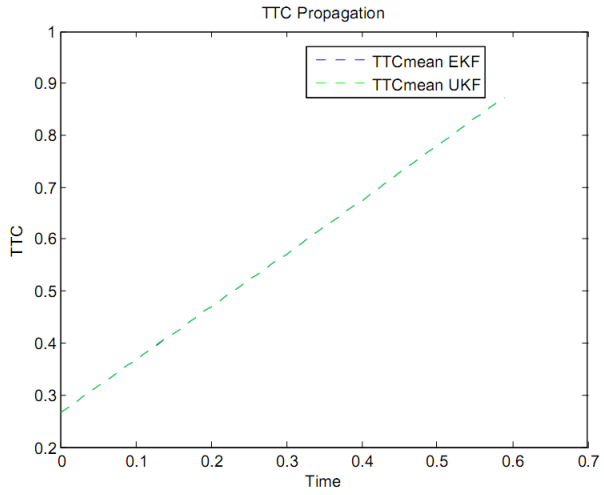


Figure 5.125: TTC of test 16

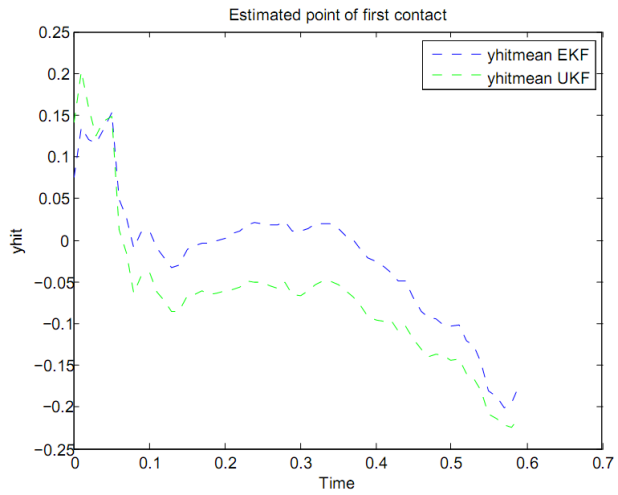


Figure 5.126: Point of First Contact estimation of test 16

Test Scenario17 Fig 5.126 shows the vehicle collision orientation for Test 126.

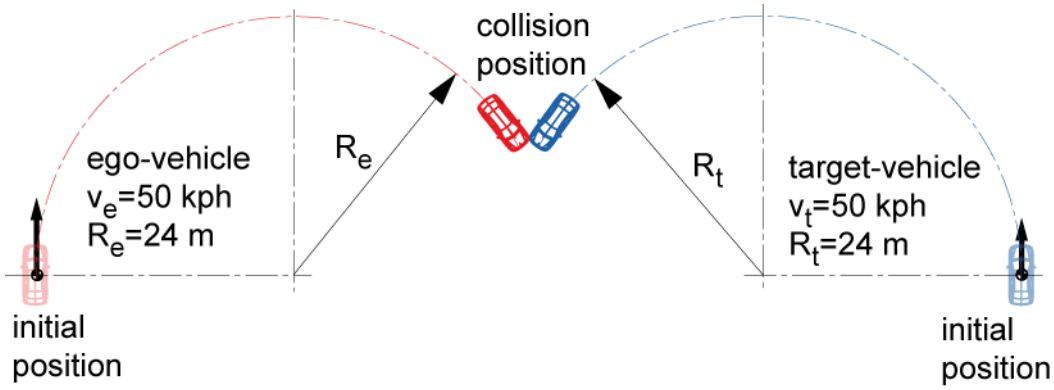


Figure 5.127: The final incidence orientation of test 17

Fig 5.127 shows the trajectory prediction of the last 0.6s before collision.

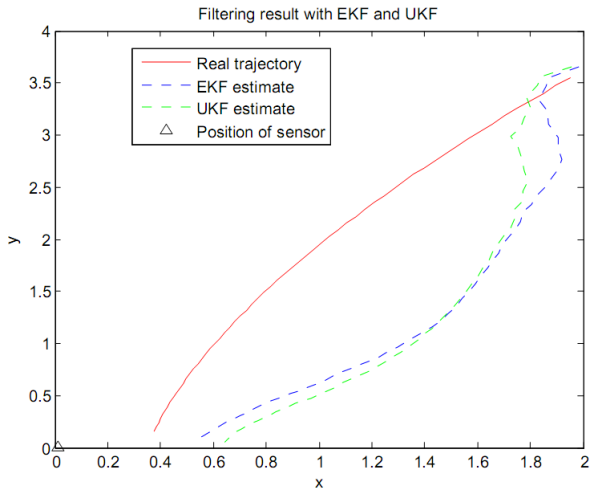


Figure 5.128: Trajectory of test 17

Fig 5.128 and Fig 5.129 show the error involved in the prediction of x and y coordinates respectively.

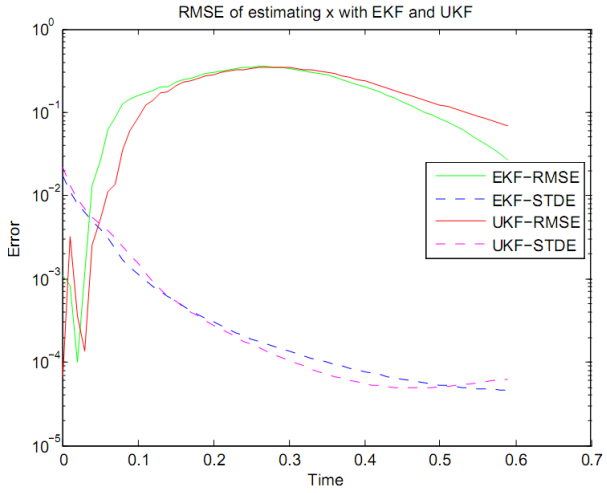


Figure 5.129: RMSE of x and y estimation of test 17

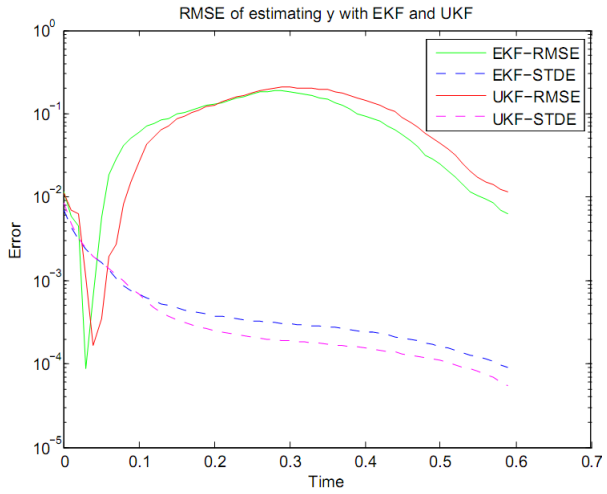


Figure 5.130: RMSE of y estimation of test 17

In Fig 5.131, we see the value of TTC going down below 0.15s and in Fig 5.131, the y coordinate of the first point of impact is getting closer to 0 at approximately 0.25 after sharp divergence, which means collision is inevitable.

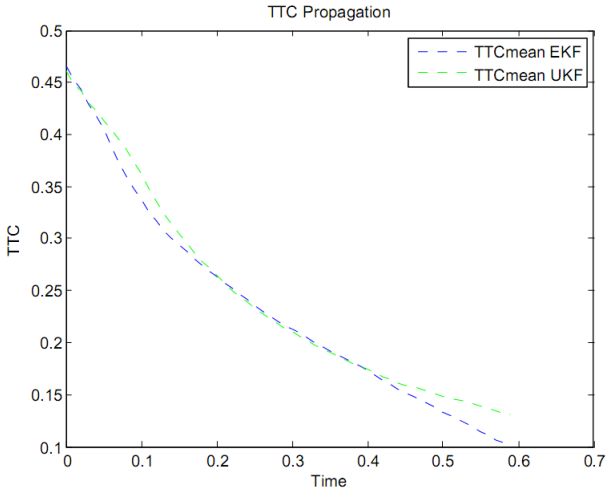


Figure 5.131: TTC of test 17

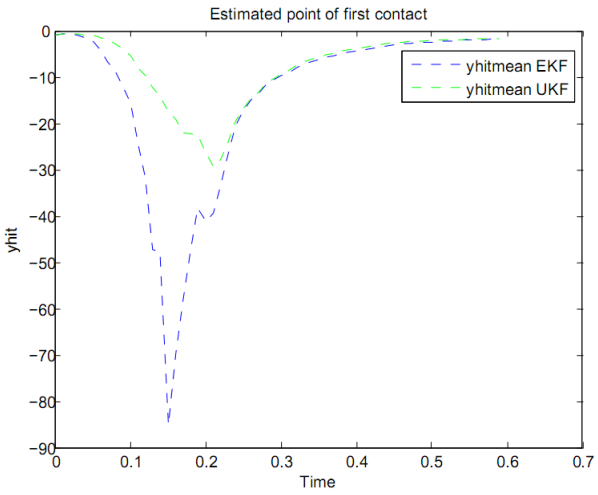


Figure 5.132: Point of First Contact estimation of test 17

Test Scenario18 Fig 5.133 shows the vehicle collision orientation for Test 18.

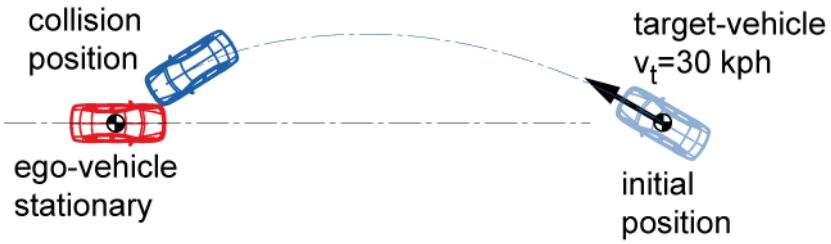


Figure 5.133: The final incidence orientation of test 18

Fig 5.134 shows the trajectory prediction of the last 0.6s before collision is given.

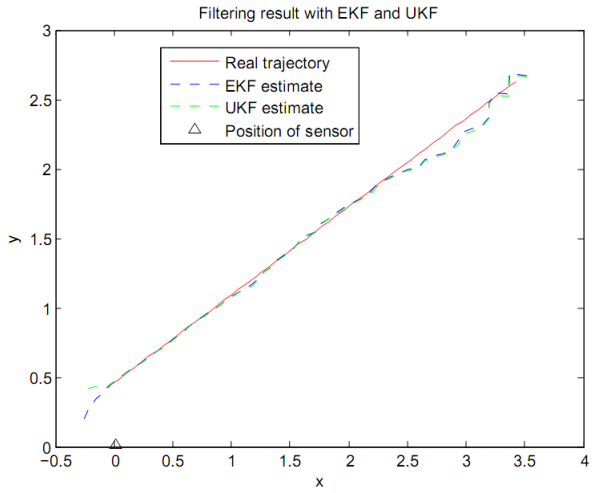


Figure 5.134: Trajectory of test 18

Fig 5.135 and Fig 5.136 show the error involved in the prediction of x and y coordinates respectively..

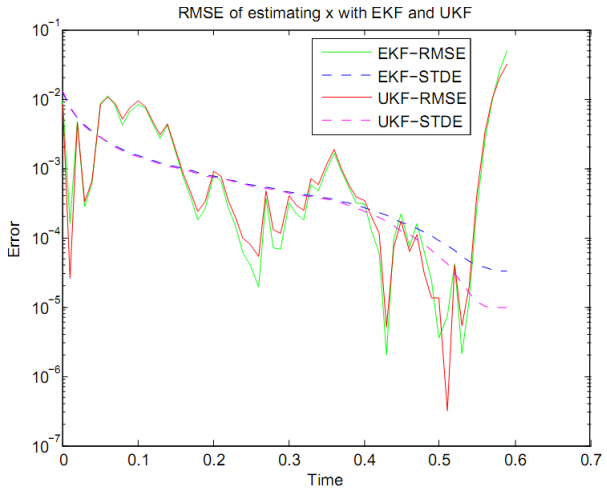


Figure 5.135: RMSE of x estimation of test 18

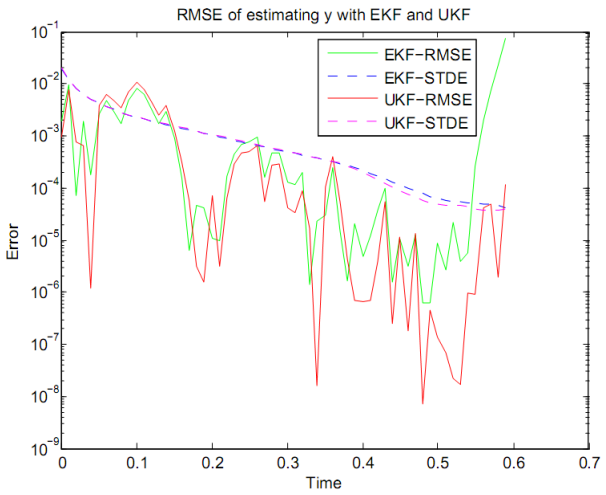


Figure 5.136: RMSE of y estimation of test 18

In Fig 5.137, we see the value of TTC going down from 0.6 below 0.1s and in Fig 5.138, the y coordinate of the first point of impact is approximately within the range of 0.2m and 0.6m, which means collision is inevitable.

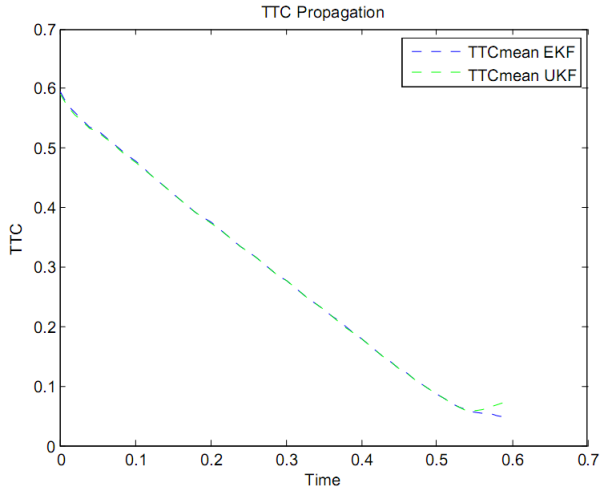


Figure 5.137: TTC of test 18

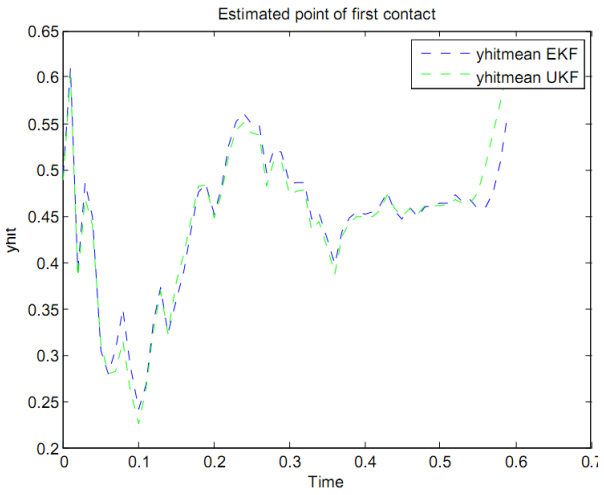


Figure 5.138: Point of First Contact estimation of test 18

6 Discussion of the results

From the simulation results, it is observed that different models can be implemented for assessing the traffic situation on the road.

Collision prediction using overlap and TTC calculation is very fast and robust, but to make the numbers we get meaningful, a supportive accident scenario database is needed. This model is very fast and robust and can be applied for any general case.

Collision prediction using Monte Carlo simulation is very fast, effective and can be used in conjunction with vehicle tracking. It always demands the covariance and the mean of the tracking result. It can be used for any accident scenario.

Collision prediction using stochastic drivers' input is able to quantify, the risk involved in the driving strategy combinations of the two drivers. This is even very useful for problems of collision avoidance. It can be used for general accident scenarios. The only disadvantage is its slowness, as it involves so many simulations. It is hoped that the model can be made faster in the future with some improvements.

The last model of collision prediction, which uses TTC and space propagation, is very fast and is used with radar tracking. Moreover it can be used for any accident scenario investigation. The advantages of this model are, prediction of collisison even in times, when there is much error involved in tracking and assessing the risk both along the x and y axes of the vehicle.

Finally, we can conclude that any one of these models can be implemented depending on the specific problems in hand.

7 Summary and conclusions

In this thesis an effort is made to do a preliminary study of different state of the art models available for tracking and collision estimation. The literature review is coupled with implementation of some of these models and in addition some suggestions for improving the stochastic driver's input based and the overlap based collision prediction models is given. For the stochastic driver's input based collision prediction model, a new way of assesing the traffic situation is shown and a single track vehicle model, which considers the basic driving limits of a vehicle is implemented. For the overlap based collision prediction model, it is shown in this work, how to relate all the important geometrical variables with the vehicle dynamics of both vehicles. The simulation results are promising and further refining would hopefully produce more satisfactory results.

The two Kalman filters, EKF and UKF are implemented for tracking and it seems that both have similar performances, but cases of big differences in performance are also exhibited. The advantage of using UKF over EKF is in avoidance of effort needed to get the Jacobian matrices.

The coordinated turn model is shown to be suitable for tracking along a straight road or a curved one. It basically mixes the tracking results of two or more state space models proportionally. This might be very interesting for future work, as its adaptive nature makes it to have a better performance.

The collision prediction model, which is based on stochastic input of drivers is shown to be suitable and is able to provide an assessment of different possible scenarios, including the minimum possible collision time estimation. It makes it possible to measure the risk involved in the current strategy that is being followed or considered to be followed. This might be even interesting for collision avoidance application. Moreover this model can be used for any kind of traffic accident orientation. To make it suitable for real-time application, some improvements should be made and moreover programming it in C can make it even faster.

The collision estimation model, which is based on TTC calculation and projection overlap on a hypothetical line, which is perpendicular to the relative velocity has proven to be effective and fast for real time application. It can be used together with a database built using several test drives. This can be used, if visual tracking and own car state estimation is available or if the vehicles communicate each other and exchange their own car state estimations.

The Monte Carlo simulation based collision prediction model, which needs the mean and covariance matrices of Kalman filtering state estimation, is also very fast and effective. Similarly like in the Case of the projection based model, this model can be used, if visual tracking and own car state estimation is available or if the vehicles communicate each other and exchange their own car state estimations.

The last model, which is based on TTC and space covariance, is also very effective, even it is able to predict the y coordinate of the first hit point. It is shown that eventhough the radar tracking itself might not sometimes give sufficient results, since ego mtotion compensation is not considered and the time is so short for filter settling (giving stable values), the algorithm is still able to predict collision.

Here in this work, it is also shown that, collision estimation can be performed, without the use of the Kalman Filter, just with the use of models that make use of very simplified vehicle dynamics models.

As a final conclusion, it can be said that this work is still far from perfection, but it can be considered as a step towards fruitful collision prediction model implementation.

Bibliography

[BY07] M. Buhren and B. Yang. A global motion model for target tracking in automotive applications. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 2, 2007.

[Che08] X. Chen. *Requirements and Concepts for Future Automotive Electronic Architectures from the View of Integrated Safety.* Universitätsverlag karlsruhe, 2008.

[DD07] Peter Schuster Dana Desrosiers, Charles Birdsong. A pre-crash simulator to evaluate vehicle collision prediction algorithms. 2007.

[Eic10] Arno Eichberger. Contributions to primary, secondary and integrated traffic safety. *Habilitation Thesis, Graz University of Technology*, 2010.

[Eid08] A. Eidehall. *Tracking and threat assessment for automotive collision avoidance.* PhD thesis, 2008.

[EUC01] Eu,"white paper- european transport policy for 2010, time to decide",european commission, office for official publications of the european communities, luxemburg, 2001. 2001.

[FDL03] K. C. Fuerstenberg, K. C. J. Dietmayer, and U. Lages. Laserscanner innovations for detection of obstacles and road. In *Proceedings of AMAA*. Citeseer, 2003.

[Fel07] Timothy Felty. Multivariate gaussian distribution. *http://mathworks.com/matlabcentral/fileexchange/5984-multivariate-gaussian-distribution*, 2007.

[GCT08] Y. Goyat, T. Chateau, and L. Trassoudaine. Tracking of vehicle trajectory by combining a camera and a laser rangefinder. *Machine Vision and Applications*, pages 1–12, 2008.

[Gia09] Luigi Giaccari. In polygon test for convex polygon, 2009.

[GL02] D. Hefferman G. Leen. Expanding automotive electronic systems",. *IEEE Computer*, pages 5–6, 2002.

[Hal05] Dr. Yngve Haland. Integrated safety by pre-crash triggering. Technical report, Autoliv Inc, 2005.

[HS07] J. Hartikainen and S. Särkkä. Optimal filtering with kalman filters and smoothers-a manual for matlab toolbox ekf/ukf. *Journal of the American Statistical Association*, 2007.

[JAC08] Guillaume JACQUENOT. Minimum distance between two polygons. *http://mathworks.com/matlabcentral/fileexchange/22444-minimum-distance-between-two-polygons*, 2008.

[Jan05] J. Jansson. *Collision avoidance theory with application to automotive collision mitigation*. PhD thesis, Linköping University, 2005.

[Köh04] M. Köhler. Accurate precrash detection. *IBEO Automobile Sensor GmbH*, 2004.

[KSD09] N. Kaempchen, B. Schiele, and K. Dietmayer. Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 10(4):678–687, 2009.

[LGPN08] A. Lambert, D. Gruyer, G. S. Pierre, and A. N. Ndjeng. Collision probability assessment for speed control. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 1043–1048, 2008.

[MRR06] S. Müller, H. Ritter, and H. Rohling. Pre-crash application for multiple target situations. In *IRS 2006 International, Krakow, Poland*, 2006.

[MTO04] M. M. Meinecke, T. B. To, and J. Obermann. A 24 ghz radar-based automotive pre-crash system. In *International Workshop on Intelligent Transportation, Hamburg/Germany*, 2004.

[MZD09] M. M. Muntzinger, S. Zuther, and K. Dietmayer. Probability estimation for an automotive pre-crash application with short filter settling times. In *Intelligent Vehicles Symposium, 2009 IEEE*, 2009.

[Ord05] F. Orderud. Comparison of kalman filter estimation approaches for state space models with nonlinear measurements. In *Proceedings of Scandinavian Conference on Simulation and Modeling, Trondheim*, 2005.

[Pre] Prevent. Time to collision.

[PTAA07] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone. Sensor fusion for predicting vehicles' path for collision avoidance systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):549–562, 2007. Cited By (since 1996): 7.

[RvdM97] Eric A. Wan Rudolph van der Merwe. The square-root unscented kalman filter for state and parameter-estimation. *Oregon Graduate Institute of Science and Technology*, 1997.

[SA09] Fawzi Nashashibi Samer Ammoun. Real time trajectory prediction for collision risk estimation between vehicles. *IEEE International Conference on Computational Photography, San Francisco*, 2009.

[TPA05] M. Tsogas, A. Polychronopoulos, and A. Amditis. Unscented kalman filter design for curvilinear motion models suitable for automotive safety applications. In *8th International Conference on Information Fusion*, volume 2, 2005.

[WB95] G. Welch and G. Bishop. An introduction to the kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 1995.