CHALMERS

Sensor Fusion for Vehicular Networks

Master Thesis in Communication Engineering

Naga VishnuKanth Irukulapati

Department of Signals & Systems
Chalmers University of Technology
Gothenburg, Sweden 2011

Master's Thesis 2011:1

Abstract

Traffic congestion, CO2 emissions and road traffic fatalities are the three major concerns associated with transport sector. It is identified that human factor is the cause of road accidents in 80 percent of the cases. There are many driver assistance systems but they are mostly autonomous. With these current autonomous systems, if some disturbance is created in one of the vehicles on the road, then there is an amplification of the disturbances (called as string instability or shockwaves) in the follower vehicles. Cooperative driving systems provide a promising solution to the string instability problem and help in increasing throughput, reducing traffic congestion, improving safety, lowering CO2 emissions and reducing fuel consumption. However, for large scale deployment of these cooperative systems, ensuring safe and reliable operation is a big challenge. Grand cooperative driving challenge (GCDC) is a competition that aims to demonstrate that the cooperative systems work and achieve some specific objectives. The current master thesis is done as part of this GCDC competition.

In the master thesis, a real time sensor fusion system is developed for the application of vehicle platooning (road trains). The task of the sensor fusion algorithm is to provide filtered signals to the controller and make sure the system is robust to sensor failures. A careful balance is done between the information from the wireless communications and in-vehicle sensors by analysing the limitations of each sensor and complementing them with other sensors. To perform the task of fusing/combining the signals, an extended Kalman filter set up is used in the project. Algorithms that handle asynchronous sensor data, popularly called out-of-sequence measurement (OOSM), are studied and implemented. The sensor fusion system along with other blocks in the project were tested in real time for the GCDC competition and the results convey that the sensor fusion system is working.

Keywords: Sensor fusion; Extended Kalman filter; Out-of-sequence measurement (OOSM); Grand cooperative driving challenge (GCDC); Adaptive cruise control (ACC); Cooperative adaptive cruise control (CACC); Shock waves; Fail-safe design.

Acknowledgement

I would like to thank Lennart Svensson, Henk Wymeersch and Erik Ström for offering me this thesis. It was great working with them. They are very great mentors and were always open to questions. I would like to thank Fredrik Sandblom, Kirstoffer Lidström and Daniel Svensson for discussing through out whenever needed. I am very much indebted to the whole GCDC Chalmers team for making this journey a very remembered one: Bruno, Reza Tabar, AliReza, AliRaza, Usman, Josef, Cristoffer. We all worked together in the project and helped each other when we encountered a problem. Special thanks goes to my supervisor at DA-IICT Prof. Vijay Kumar. He was very active for discussion wheneever I call him or e-mail him.

Special gratitude goes to my family: my father Nageswara Rao, mother Adeswari, sister Manasa, brother Vijaya Bhaskar, uncle Vakuleswar Rao and many more people who have been giving me constant support throughout my life. They were all there through out with me with constant support and love.

I would like to thank all my friends especially Harsha, Ajay, Koushik, Shanmukha, Hashim and Dapeng, Mehrnaaz, Ahzaz, Farzaneh, Sara for their support through out masters. Special thanks to all my friends here and in India who made this two year journey very happy through out.

Naga VishnuKanth Irukulapati, Göteborg, June 2011.

Contents

Chapter 1

Introduction

Traffic congestion, CO2 emissions and road traffic fatalities are the three major issues that concerns the people involved in transport sector. According to a 2004 world health organization (WHO) report, nearly 1.3 million people die every year due to road traffic crashes and around 20-50 million people get injured. It was pointed out that if action is not taken by year 2020, the traffic fatalities may increase to 1.9 million annually. It is also disappointing to know that only 15% of the countries have comprehensive laws related to five key risks: drinking and driving, speeding, non-use of helmets, child restraints and seat-belts [11]. Inspite of these terrific numbers, there is a lot of hope on the other side of the coin.

Road traffic safety has been of prime importance these years. This interest can be seen in the announcement recently by Volvo cars that by the year 2020, no one shall be seriously injured or killed in a new Volvo. UN has called the decade 2011-2020 as the decade for road safety. In March 2010, UN General assembly has proclaimed that this decade should provide a framework to countries and communities to increase action to save lives on the world's roads[11]. UN secretary-general has requested the governments to release their national plans for the decade. One can see from these and recent research in this area that there is a growing importance on the road traffic safety systems worldwide.

Safety systems can be categorized into three main categories-passive, preventive and active safety systems. Passive systems try to reduce the post-crash damages. In EU and in many other countries, passive safety systems such as air-bags and seatbelts are a norm from quite some time. Preventive safety has recently taken a long stride with the arrival of cost efficient sensors. These systems warn the driver if a vehicle inadvertently leaves its lane; automatic braking systems also come in this category. In this front, there is a lot of research on Advanced Driver Assistance Systems (ADAS) that assist the driver in the driving process but currently they are mostly autonomous (non-cooperative) systems.

According to a report in 2003 [13], drivers are the reason for accidents with injured people in nearly 86% of the cases. The common mistakes include-driving too fast or too close; wrong lane or wrong overtaking; alcohol consumption etc. It is pointed out in [14] that around 50% improvement can be achieved by co-operative driver assistance. So, the next natural step in this journey towards safety systems seems to be the sharing of information using wireless communication between the entities (vehicles, infrastructures etc.). ADAS combined with cooperative driving systems, known as intelligent cooperative systems, is a promising solution towards better collective behavior which results in increased road throughput, reduced traffic congestions, improved safety, lower CO2 emissions and reduced fuel consumption due to smoother driving.

1.1 Cooperative systems & vehicle platooning

As discussed in above, the existing ADAS systems are fully autonomous systems and they use the information gathered by in-vehicle sensors which scan the surroundings of the vehicle. However, it is believed that intelligent sharing of information within vehicles and with infrastructure about their surroundings would make a significant progress in this area of ADAS. Cooperative driving is seen as a futuristic technology that brings us closer to the vision of congestion-free world where the road transport would be safe, clean and efficient [5, 17]. Vehicle platooning (road trains) is one such technique where cooperative system set up can be envisaged.

In the late 1990's and early 2000's, inter-vehicle communication played an important role in the vehicle control. An automated platoon was demonstrated by California's PATH (Vehicle Platooning and Automated Highways), Chauffeur in EU and cooperative driving systems in Japan [16]. In PATH, all vehicles have same vehicle dynamics and the controllers in each vehicle are designed based on the information from preceding vehicle and the lead vehicle [15]. Recently, Volvo has started with SARTRE project, where the concept of vehicle platooning is envisaged. In Sartre project, the convoy of vehicles is lead by a professional driver in a lead vehicle and all other vehicles in the platoon can follow the lead vehicle. Once the vehicles are in a platoon, the drivers can relax and need not worry about driving.

Platooning can enhance road capacity by allowing small head way time (distance between a follower and a leading vehicle divided by follower speed) [5]. Headway time can be also defined as the distance between the top of one vehicle to the rear end of the front/preceding vehicle giving a measure of the time it takes for the follower vehicle to cover that distance. Adaptive cruise control (ACC) systems were introduced several years ago to achieve and maintain specific head way time using the information from the in-vehicle sensors such as radar, lidar, vision-based systems. These sensors measure the relative distance, relative velocity etc for the preceding vehicle; based on this information, automatic throttle and brake are applied and the driver needs to just steer. The headway times of common ACC systems range from 1 sec - 3 sec. The main problem of the current ACC systems is the string instability problem that can cause shock waves. String stability, which is desired, can be described as the ability to prevent the amplifications of preceding vehicle's speed to the following vehicles in the chain. In traffic with lot of vehicles behind each other, a sudden brake action of one of the vehicles can introduce a disturbance in the platoon of the vehicles something like a shockwave and can lead to a full stand-still of follower vehicles or even a collision [5]. This situation can be tackled by cooperative systems. If each vehicle in the platoon sends out the information like the position, velocity and acceleration; controllers can be designed not just based on the preceding vehicle information but also based on the information from the other front vehicles in the platoon. For example, a controller can be designed based on the information from the preceding vehicle and also from the lead vehicle. These type of controllers that are ACC-like but also use information from wireless communication (vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I)) are generally called as cooperative adaptive cruise control (CACC) systems. These systems provide a promising solution to the string instability problem by having a chance to look further ahead in the traffic and react.

It has to be noted that in vehicle platooning projects like PATH, SARTRE project, all vehicles behind the lead vehicle have the same vehicle dynamics and also the lead vehicle is pre-assigned. In contrast, in GCDC, most of the participating vehicles are from different vendors. Hence, it is a challenge to have an idea of the implementation of the platooning systems of the other teams. Also, in GCDC, the platoon leader can be any vehicle from the vehicles participating in the competition. This seems to be more like a real world scenario where the platoon leader can be anyone and also any vehicle can be the follower. This is a challenge by itself especially during the controller design as uncertainty about the other vehicles behavior has to be addressed in the controller design.

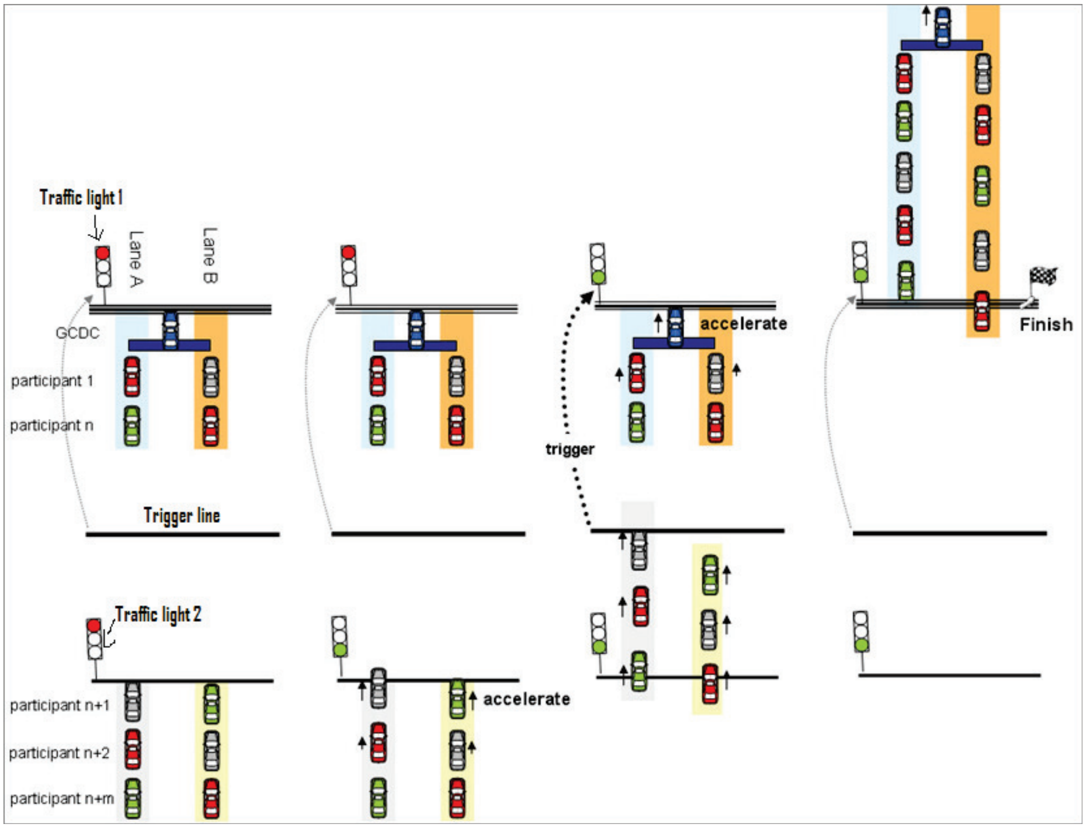


Figure 1.2.1: Urban scenario [5].

1.2 GCDC

This section provides an overview of the Grand Cooperative Driving Challenge (GCDC) 2011 event, that is initiated and organised by The Netherlands Organization for Applied Scientific Research (TNO). GCDC aims to accelerate the integration, development, deployment of the cooperative driving systems and also demonstrate that the cooperative systems work and achieve some specific objectives.

The GCDC organization has defined some scenarios (briefly described in urban scenario-Section 1.2.1 and highway scenario-Section 1.2.2) which were chosen to demonstrate the possibilities that one achieve using cooperative systems. The participants compete in a cooperative driving scenario and the technical performance of the systems will be judged. String stability and smoothness are the two main criteria for platooning performance used in GCDC competition. Since smoothness directly relates to the acceleration levels, it lowers the fuel consumption and hence lower emissions. These criteria also help to lower the congestion problem and increase traffic flow. Since traffic on highways may contain more than one platoon separated by large interplatoon distances, this behavior is also tested during the GCDC by exchanging messages like platoon join and leave messages. In urban scenario, the acceleration behavior at the traffic lights can be improved by using platooning technology by maximizing the throughput at green lights for example. So, both interplatoon and intraplatoon behavior are important for smooth traffic flow and also to reduce traffic congestion. As part of the rules, the vehicles in the competition should operate automatically in longitudinal direction but the driver can steer and can control the lateral direction. Two scenarios are briefly described below.

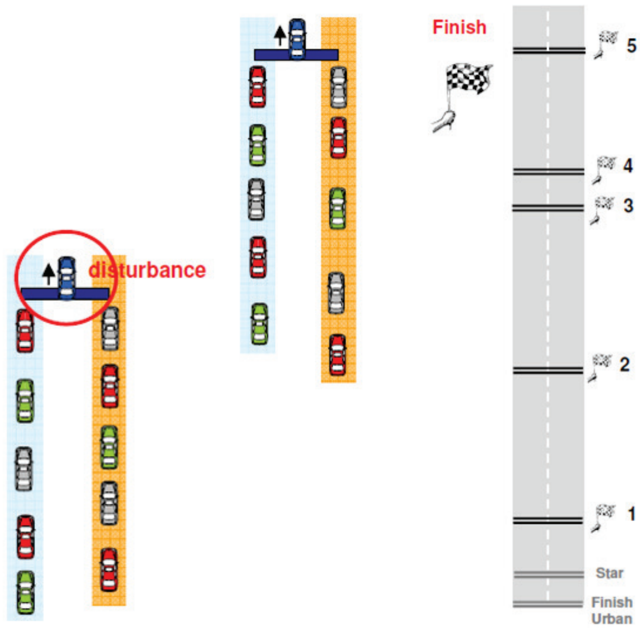


Figure 1.2.2: Highway scenario [5].

1.2.1 Urban-scenario

The main idea of this urban-scenario setting in two-fold: maximizing throughput at the traffic light and smoothly joining the platoon of vehicles away from traffic light (concerning to interplatoon behavior). At start of each round of the competition, all vehicles are divided in two lanes (A and B) and are subdivided further into two groups as can be seen in Fig. 1.2.1. So, it can be seen as two platoons in each lane-say vehicles near traffic light 1 as platoon 1 and vehicles near traffic light 2 as platoon 2. These group of vehicles wait near the traffic lights for the signal to turn green. The objective is first to cooperatively accelerate near the traffic light as the traffic light 2 turns green. At the start of each round, the traffic light 2 turns green and all the vehicles in platoon 2 starts driving by maintaining a smooth and stable platoon. As soon as the first vehicle crosses the trigger line, the traffic light 1 turns green and the gcdc lead vehicle starts accelerating with a predefined velocity profile. The vehicles of platoon 2 try to join platoon 1 as smoothly as possible, while maintaining platoon stability. The urban part of the scenario ends when the last vehicle of either Team A or Team B crosses the finish line first.

1.2.2 Highway scenario

The objective of the highway scenario is to see whether the participants can maintain a stable platoon by attenuating any disturbances created by one of the vehicles and also by avoiding the shock waves. The highway scenario begins as soon as the urban scenario ends and the GCDC vehicle remains at a constant speed for sometime till all vehicles are ready for the highway scenario. Fig. 1.2.2 shows the situation of the highway scenario. In highway scenario, the GCDC lead vehicle which is in between the two lanes creates disturbances in longitudinal accelerations similar to those that are expected in highway. In this scenario, the task of all the vehicles in the platoon is to make a smooth platoon by fully operating automatically in longitudinal direction. All the vehicles in the platoon that reaches the finish line first will receive a point in this scenario.

1.3 Goal of thesis

Ensuring safe and reliable operation of the cooperative systems (e.g., CACC) is one of the big challenges towards large scale deployment [5]. Fail safe design can be achieved by equipping the system with several (redundant) solutions that ensure safety during systems failure. One other aspect is that the if the system fails, the overall impact should not be more than if the system had not been use: this is called graceful degradation. From the point of view of signal processing, since there are many sensors that provide (redundant) information, a technique has to be built that ensured that the signals that are provided to the CACC system are robust. It is also very important to decide whether to give importance to the V2V, V2I information or to the in-vehicle sensors. A safe fail design has to be ensured by taking all sensors signals and also take care that signals are not degraded than in the original case.

For the design of a CACC, the following information may be needed:

- relative distance to the preceding vehicle
- speed, acceleration of the ego vehicle, preceding vehicle and leader vehicle.

Few CACC systems may also use in the information from all the vehicles in the platoon and not just the preceding and platoon leader information. It has to be assured that the signals that the controller needs are accurate and are noisy free. The goal of the thesis is to develop a sensor fusion system for vehicular networks. With the help of sensor fusion, information from different sensors is fused to get a robust estimate of the signals that the controller needs. For designing a good fusion system, advantages and disadvantages of the different sensors has to be studied so that complementary nature of the sensor signals can be explored. Next step is to design a filter that can do the fusion of different signals and also filters the noisy signals. An extended Kalman filter set up is used for this fusion by deriving the equations from continuous time to discrete time. Synchrnonising information from different sensors before fusing them is quite essential and hence algorithms to deal with out-of-sequence measurements is also studied. In the next chapters, concepts necessary for understanding the implemented sensor fusion system will be presented.

Chapter 2

System architecture & Problem formulation

2.1 System architecture

Fig. 2.1.1 shows the overall system architecture that has been used in the project. The system consists of different clearly defined modules, each with its own task. All of these modules are associated with one or more hardware components. In this section, brief description of the different components used in the project will be discussed.

- GPS: Global Positioning System(GPS) is a satellite-based global navigation system that works in all weather conditions providing position and time information. Each satellite continuously broadcasts its timing information from internal clock(s) available on the satellite. A GPS receiver picks up those signals from different satellites and also logs the received time. The receivers then calculate the distance between the satellites and itself from the time it took for each of the signals to reach the receiver. In principle, once the receiver calculates this distance information from 3 or more satellites, exact location can be calculated by a method called trilateration. However since the clock signals of the satellites and the receivers are not synchronized, there may be a need of 4 or more satellites for knowing the position of the receiver. Some of the sources that may affect the GPS signal and thereby the accuracy of the position information are: Atmorsphereic delays (ionosphere and troposphere delays); signal multipath; receiver clock errors; orbital errors; number of satellites visible; satellite geometry etc., [20]. The accuracy that we can expect out of these receivers can be in meters (around 10m). To improve the accuracy of the received signal, techniques such as differential and real-time kinematic (RTK) GPS techniques are used that improve accuracy to within two meters and centimeter level accuracy respectively. RTK is a highly precise technique that provides centimeter level accuracy. It needs a RTK-base station within a range of 10Km radius of the area that the actual RTK-GPS receiver is working on. Since the base station knows its exact position, it can help to improve the accuracy of the GPS receiver mesaurements by sending the correction information. Given the high position accuracy demands laid down in the rules of the gcdc competition, Trimble SPS852, an RTK-GPS is used in the project.
- Communication: In order to test the cooperative setting in vehicular networks, all vehicles need to communicate their vehicle related information and to achieve this, one needs a transceiver unit to do the task. In order to cope with the gcdc competition rules, the communication unit should be able to use an 802.11p based transceiver. The set of hardware performing communication (vehicle to vehicle (V2V) and vehicle to infrastructure (V2I)) was bought as a package from the GCDC organization. It consists of an Alix board 3D2 with an Atheros based WLAN card,

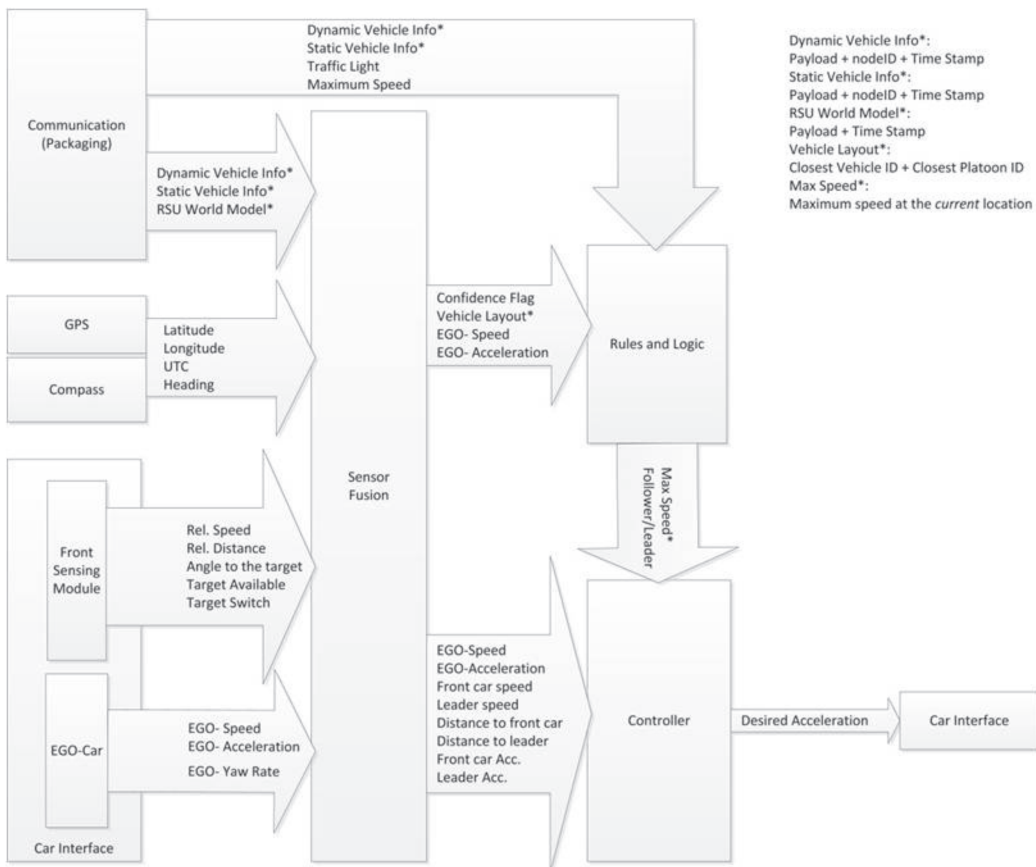


Figure 2.1.1: System Architecture

Mikrotik R52H. In addition to the competition package, an extra set of 6 dBi radio antennas was bought to ensure the desired communication range.

- Rules and Logics: Part of the objective of the GCDC competition revolves around the establishment of a platoon. Such an operation must respect a pre-defined set of rules which is based in not only the ego vehicle, but also in the states of all other vehicles. These rules were implemented in the form of a state machine using Matlab Simulink.

- Vehicle Interface: The vehicle interface can be considered as a gateway that handles all the interactions to the vehicle's CAN bus. This interface is responsible for extracting the signals that represent the current state of the vehicle. The vehicle interface also sends the control input to the vehicle in terms of a requested acceleration. In this project we bypass the requested acceleration signal originating from the standard ACC in the S60. The computed requested acceleration will still be constrained by maximum and minimum bounds, defined in the engine node of the vehicle

- *Front Sensing Module*: This module takes the information from Radar, camera and lidar and fuses the information. It then sends out the information of the centre in path vehicle regarding the relative distance, range rate,acceleration of the preceding vehicle and also ego vehicle speed.

- With the exception of the communication protocol, all software developed for the project runs on a single real-time hardware (RTH), a dSPACE MicroAutoBox 2. Ranging from interaction with the car CAN bus, serial communication with the GPS and the Ethernet interface with the communication unit, this real time environment (RTE) is also responsible for algorithms related with platooning logic, sensor fusion and vehicle control, as described earlier.

- The RTH can be regarded as a central hub gathering information from all the hardware and sensors in the vehicle. A Simulink model was built to incorporate all necessary algorithms aimed at data treatment and signal processing.

- Sensor Fusion: To get the benefits of the cooperative set up, one can use the concept of sensor fusion to improve the accuracy and the reliability of the estimation of the data. Vehicles get information from the communication, internal and external sensors. By using data fusion, data from different (complementary) sensors can be combined for enhanced reliability and accuracy. The current thesis work is focused on this part of the project.

The actual information used from the these sensors will be discussed in Section 2.2.2.

2.2 Problem formulation

2.2.1 Motivation

Figures 2.2.1 - 2.2.3 show the velocity of the preceding vehicle. This velocity is measured from two different sensors: one is the Radar measured velocity (written as CAN measurement) and the second measurement comes from V2V itself. In the bottom plot in all these figures, the sensor fusion output is also shown for clarity. As can be seen from the plots, the behavior of the signals from these two different sources is quite different. For example, in Fig. 2.2.1, the radar reports bad velocity measurement whereas in Fig. 2.2.3, we can see that the V2V information is almost zero meaning that we didn't receive any information from the preceding vehicle for that scenario. Fig. 2.2.2 shows a scenario where both the sources of signal provide good measurements of the preceding vehicle. Given these two information sources, the task of the sensor fusion is to combine these sources of information and get a better estimate of the parameter of interest, in this case the velocity of preceding vehicle.

One naive way of combining information from these information sources is to take an average of the two signals but as can be seen in those plots, when the V2V information is zero, then averaging

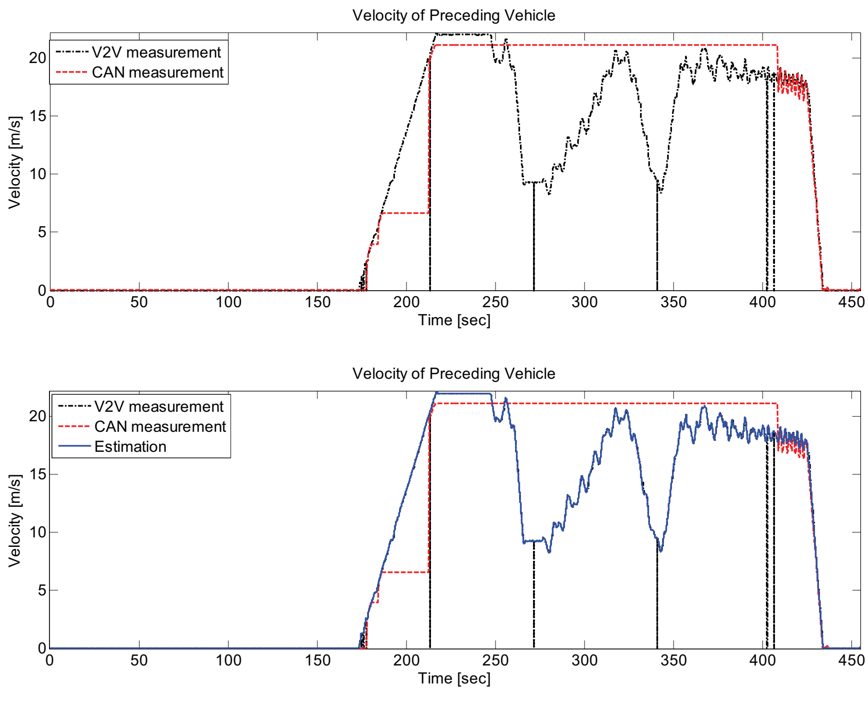


Figure 2.2.1: Speed of preceding vehicle: case1

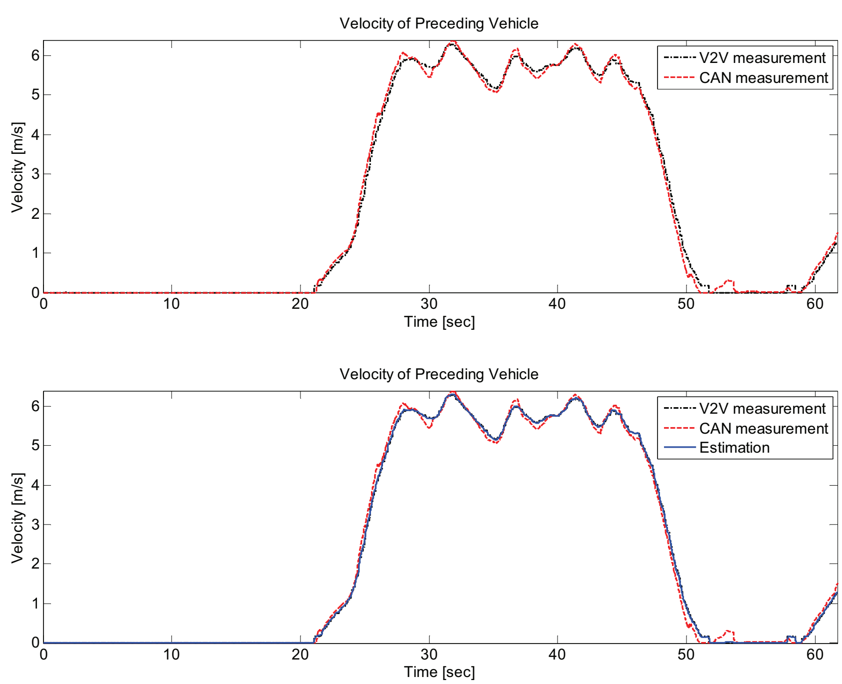


Figure 2.2.2: Speed of preceding vehicle: case2

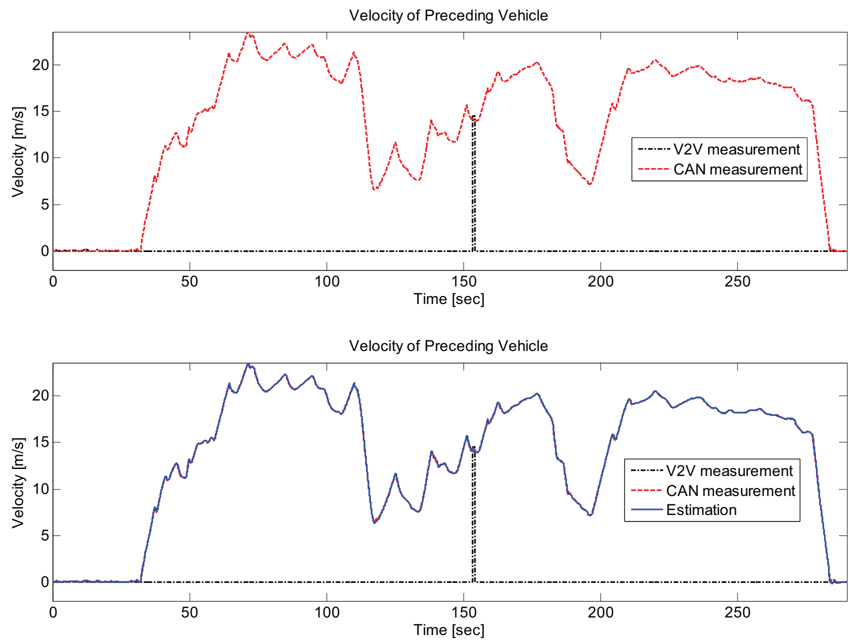


Figure 2.2.3: Speed of preceding vehicle: case3

may deteriorate the performance of the filtering system. A clever filtering technique has to be adopted that can take into account the prior knowledge that we have about the signals so that good estimates can be extracted.

2.2.2 Sensor fusion

There are many sensors in the system that we use and each one has its own strengths and weaknesses. Following are the sources of information:

- Ego RTK – GPS: The information from the ego vehicle GPS unit used in the thesis includes position, speed and heading. Information about the reliability of the received GPS data is also used from the checksum and status messages. The GPS that we are using in the project is real-time kinematic GPS which gives position information with centimeter level accuracy. However the heading that we get from GPS fluctuates very rapidly when the vehicle is at rest or at low speeds. The GPS is also prone to multipath errors and loses the connection under the bridges and tunnels. The GPS system that we can update information either 5, 10 or 20 Hz. We used the system to take inputs from GPS at 20Hz.

- Front sensing module(FSM): There is information from radar, camera about the range (relative distance), range rate (relative velocity), relative acceleration from the ego vehicle to the preceding vehicle. The relative acceleration signal that comes out of this module is very bad but the range measurements are quite accurate. The radar system that is used in the project is made by Delphi. The detection range is 150 m with a field of view of 16^0. In raw form, the radar system can detect upto 20 targets [18]. But the information that we used is the fused information from radar, camera and this FSM module gives the information about the centre in path vehicle's information and range, range rate, relative acceleration, whether the vehicle is present or not, whether the target has changed are used in the project. The FSM module was not reporting stationary objects. For example, when there is a vehicle which is standing still at some distance and we are approaching that vehicle, the FSM module was reporting that there is no vehicle in front which is not the case. Also the radar system is limited by the field of view. If there is a vehicle which is not in the field of view of the radar, then the FSM module doesn't report any vehicle.

- CAN: Ego vehicle's speed and acceleration are taken out from the CAN bus.
- V2V communication: One more information source is the vehicle-to-vehicle communication. Some of the information that is communicated through wireless include the vehicle's position, position time stamp, position accuracy, velocity, heading, acceleration, yaw rate. This information is sent at a rate of 10 Hz with a header that contains the sequence number, message time stamp etc. These messages are received from all the vehicles in the range. Apart from these messages, information from road side units such as traffic light information, speed limits on different roads were also communicated. For fusion in the project, we use the position, velocity, acceleration, heading of the preceding/front vehicle. Position and message time stamp and message sequence number are also used during fusion to synchronize data from different sources and also to avoid fusing same information more than once.

The preceding vehicle information is fused with the ego vehicle's information such that overall accuracy of the estimation and fusion algorithm is improved. By using many information sources, we can achieve robustness to temporary sensor failures.

As a result of sensor fusion, we are expecting

- filtered velocity, acceleration of the preceding vehicle (and the ego vehicle)
- accurate distance between the ego and the preceding vehicle
- better estimate of ego vehicle's position and preceding vehicle's position

Besides filtering and fusing information from preceding vehicle, information of platoon leader is also filtered. An extended Kalman filter set up is used for fusing the information from preceding vehicle with the ego vehicle's sensor information. A simple Kalman filter us used to filter the information from leader vehicle. Details of the implementation can be seen in Chapter 5.

In Chapter 3, basic concepts related to the filtering theory will be introduced. Kalman filter and extended Kalman filter concepts are introduced with some simple examples. Sometimes, the linear differential equations concerning the system model are written in continuous time and later discretized. This discretization and related concepts are also discussed in later part of Chapter 3. Fusing the information from different sensors operating at different frequency leads to out-of-sequence problem which is discussed in Chapter 4. Implementation details are given in Chapter 5 and test results are discussed in Chapter 6.

Chapter 3

Filtering theory

3.1 Introduction

Estimation is the process of inferring the quantity of interest based on the noisy, indirect measurements of the sensors. An estimator is a mapping from the set of data to the parameters of interest. There are mainly two different approaches for estimating the parameter: non-Bayesian approach or Fisher approach and the other one is Bayesian approach. In Bayesian approach, the parameter that we wish to estimate is viewed as random with some prior probability density function (pdf) whereas in non-Bayesian approach, the parameter is viewed as a non-random/ unknown constant. In this thesis, we focus on Bayesian approach. A brief introduction to various estimation technqiues is provided in Appendix A.

3.2 Tracking system overview

Tracking can be seen as estimation of the state of a moving object based on measurements with the state defined as smallest vector that summarizes past history of the system. Usually tracking systems are designed to work in real-time, but can of course also be used off-line. Fig. 3.2.1 gives an overview of the tracking system. Essentially a tracking system can be divided into the following blocks: [2]

- **Sensor(s)**. Generate measurements to form input (observations) to the tracking system, generally position(s).
- **Gating and data association**. Form connections between tracks and obsevations by solving the so called assignment problem. The assignment is generally one-to-one, meaning that each track can only be assigned to one observation. In multiple-object tracking situations, this may seem as the central part of the tracking system, however, it does not include models for motion or measurement noise.
- **Track maintainance**. As object dissappear out-of-view and new object appear, a certain amount of administration is required.
- **Filtering and prediction**. This is the main part of the tracking. It deals with the fact that the measured observations contains noise. This is contained in the measurement model. Moreover, it enables prediction of new measurement using a motion model.
  - *Measurement model*. In estimation theory we estimate a (true) parameter that is hidden in a noisy measurement. In tracking, the parameter(s) are put in a state vector, usually denoted by x_k (in discrete case and $x(t)$ in continuous case), that can change over time. The measurement z_k (or $z(t)$ in continuous case) is modeled as a function of the state x_k ,

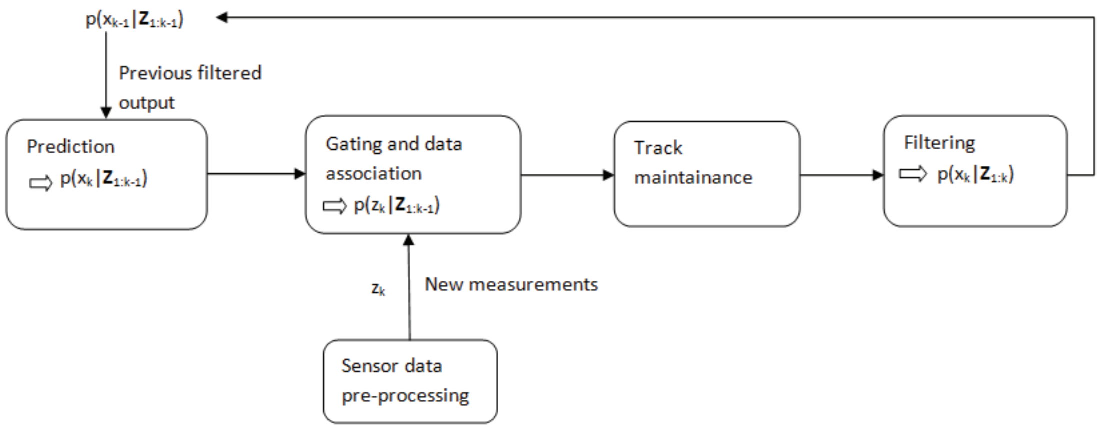


Figure 3.2.1: General description of the tracking system

and the measurement noise w_k. If w_k is Gaussian, then Kalman filtering will produce the optimal estimate of x_k.

– *Motion(process) model.* Describes how the state at time $k+1$: x_{k+1} can by computed using the previous state x_k and the process noise v_k.

In this thesis, track maintainence and association are not needed as the FSM module ouputs the centre in path vehicle's information. The competition scenarios are also pre-defined:there will be vehicles in two different lanes and the vehicles will not overtake each other during the competition round. So, this problem of track maintaince is very simple for the thesis. Same is the case with association. There is need for gating because there were many outliers in many measurements and there was a need to eliminate these outliers before using them in the filter. The main focus of the thesis will be on filtering and prediction. In the next section, basic concepts of the Bayesian way of estimation will be presented with some mathematics that is necessary. Section 3.4, Kalman filter will be introduced building on the concepts from Section 3.3.

3.3 Bayesian approach to estimation

In tracking, the parameters we wish to estimate vary over time and can be modelled using a dynamical system (time-varying parameters). Since target tracking applications are real-time, we need techniques which are not computationally expensive and preferably recursive. The background that is needed to understand the filtering using the Bayesian set up will be explored below.

3.3.1 Bayes' theorem

In Bayesian approach, the parameter of interest is a random variable with some prior pdf and we would like to obtain the posterior pdf using Bayes formula:

$$p(x|z) = \frac{p(x\ ,z)}{p(z)} = \frac{p(z|x)p(x)}{p(z)} = \frac{1}{k}p(z|x)p(x) \tag{3.3.1}$$

where x is the parameter we would like to estimate modeled stochastically and z is the measurement. $p(x)$ is the prior pdf of x and it reflects the uncertainities of x before (prior to) getting measurements. $p(z|x)$ is the likelihood function and $p(x|z)$ is the posterior distribution, that describes our knowledge about x after (post) observing the measurement z. k is a normlization constant since it doesn't depend on the parameter we wish to estimate x.

Non-Bayesian approach can be seen as a special case of Bayesian approach with prior ignorance i.e, if $p(x)$ is assumed to be uniform, then both these methods will be same. The following is an example of the three different estimation techniques. The posterior density function $p(x|z)$ describes everything that one would like to know about parameter x and the task of different Bayesian filtering algorithms is to find this posterior density function.

3.3.2 Models

To recursively estimate the state of the dynamic system, state space models is one approach to go with. In engineering, state-space representations is seen as a mathematical model of the physical system with a set of inputs, outputs and state variables (often expressed as vectors) governed by a first-order differential equation. The most general non-linear state space model in which the non-stationary parameter x_k is modeled as a dynamic model as follows:

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, v_{k-1}) \tag{3.3.2}$$
$$z_k = h_k(x_k, u_k, w_k) \tag{3.3.3}$$

where x_k is the state vector that is the quantity of interest. u_k is a (known) control input to the system. v_k is a stochastic process called process noise. z_k is the measurement at time k. w_k is the measurement noise. *Process model* or the *motion model* is given in (3.3.2). It is used to describe the distribution of the state from $x_{k-1}$to x_k using $p(x_k|x_{k-1})$. Because of the process model, all the previous measurements $\mathbf{Z}_{1:k-1}$will be utilized where $\mathbf{Z}_{1:k-1} = \{z_1, z_2, ...z_{k-1}\}$. The *measurement model* or *sensor model* is given in (3.3.3). It relates the measurements z_k to the state vector x_k using likelihood function $p(z_k|x_k)$. f_{k-1}, h_k are functions realting to the process model and measurement model and depending on the problem, these has to be defined.

3.3.3 Recursive Bayesian estimation

In recursive Bayesian estimation, the unknown pdf is obtained recursively over time using a mathematical model that describes the underlying process and using the measurements [23]. Recursive Bayesian estimation and thereby Kalman filter and so on relies on Markov process and markovian property assumption. Markov process can be desribed as a stochastic (time-varying random phenomenon) process with a Markov property (or memorylessness), in which the probability distribution of the future states given the present state of the system, depends only on the present state and independent of the the future and past states. Fig. 3.3.1 shows a simple diagram of the hidden markov model where x_{k-1} *and* x_k.. are the true (unobserved) states of the system and z_{k-1} *and* z_k.. are the measurements observed suring time instants k-1 and k respectively.

Due to the above Markovian assumption on the states, the probability of the current true state conditioned in the immediately previous one doesn't depend conditionally on the other previous states. In other words:

$$p(x_k|x_{k-1}, x_{k-2}, x_{k-3}, ...x_1, x_0) = p(x_k|x_{k-1}) \tag{3.3.4}$$

Similarly, the current measurement depends only on the current state and is conditionally independent on the previous states given current state:

$$p(z_k|x_k, x_{k-1}, x_{k-2}, x_{k-3}, ...x_1, x_0) = p(z_k|x_k) \tag{3.3.5}$$

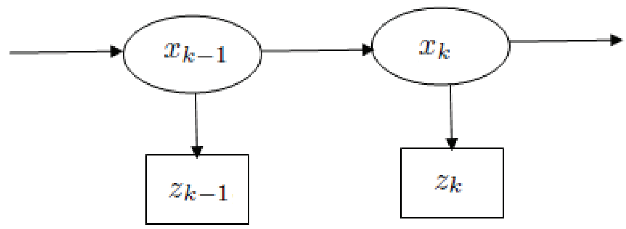


Figure 3.3.1: Hidden Markov model

By using these two equations, the joint probability distribution over all true states and measurements of a HMM can be given by:

$$p(x_0, x_1, ...x_{k-1}, x_k \ , z_1....., , z_{k-1}, z_k) = p(x_0) \prod_{i=1}^{k} p(z_i|x_i) \ p(x_i|x_{i-1}) \tag{3.3.6}$$

For deriving (3.3.6), first the joint pdf has to be written recursively in terms of conditional probabilities and later the Markov properties in (3.3.4) and (3.3.5) can be applied to get the final form. As stated earlier, the quantity of interest is the posterior distribution and this can be obtained conceptually by marginalizing the previous states and then dividing with probability of the measurement set as follows:

$$p(x_k|\mathbf{Z}_{1:k}) = \frac{p(z_k|x_k, \mathbf{Z}_{1:k-1}) \ p(x_k|\mathbf{Z}_{1:k-1})}{p(z_k|\mathbf{Z}_{1:k-1})} = \frac{p(z_k|x_k) \ p(x_k|\mathbf{Z}_{1:k-1})}{p(z_k|\mathbf{Z}_{1:k-1})} \tag{3.3.7}$$

where $\mathbf{Z}_{1:k} = \{z_1, z_2, ...z_k\}$ is the measurement vector. $p(z_k|x_k)$ relates the measurement at time instant k to the state at time k and is the likelihood function. $p(x_k|\mathbf{Z}_{1:k-1})$ gives a connection between the current state based on the past measurements and can be computed using:

$$p(x_k|\mathbf{Z}_{1:k-1}) = \int p(x_k|x_{k-1}, \mathbf{Z}_{1:k-1}) \ p(x_{k-1}|\mathbf{Z}_{1:k-1}) \ dx_{k-1} = \int p(x_k|x_{k-1}) \ p(x_{k-1}|\mathbf{Z}_{1:k-1}) \ dx_{k-1} \tag{3.3.8}$$

This equation is similar to (3.3.1), just that we have to apply Markov properties and joint pdf of HMM as stated in above equations. Discrete-time state space models as shown in Section 3.3.2 gives a way of finding the $p(z_k|x_k)$ and $p(x_k|\mathbf{Z}_{1:k-1})$.

Summary of recursive Bayesian estimation:

- Need a process model and a measurement model as stated in Section 3.3.2to start with. Then using these models, we can do the following.
- *Prediction* : In this step, we use the knowledge of previous stage and all previous measurements to predict the future state x_k. *Prediction stage* can be given as in (3.3.8).
- *Update* : The equation in (3.3.7) is called the *measurement update* and as we can see that the likelihood function is multiplied to the output of the prediction and then normalized. So, this likelihood function can be help to adjust the psoetrior distribution with the new data.
- We do this prediction and update steps recursively and find the posterior distribution that can be used to find the estimate of the state vector using some criteria like maximum a posterior (MAP), minimum mean square estimation (MMSE) etc.

Finding the analytical form of the posterior distribution as in (3.3.7) is quite difficult except for few cases. Assumptions in the Kalman filter set up make it possible to find the analytical expressions for (3.3.8) and also posterior distribution in (3.3.7). Let's explore Kalman filter in the subsequent sections.

3.4 Kalman Filter

In Kalman filter, both the process model, measurement model are linearly dependent on x_k , u_k , v_k *and* w_k. The noises v_k and w_k are assumed to be white Gaussian noise and also independent of each other. In this setup, $p(x_l|\mathbf{Z}_{1:m})$ is Gaussian for all l and m. Using these assumptions, the process model and the measurement model can be expressed as follows [1, 3]

$$x_k = F_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}v_{k-1} \tag{3.4.1}$$

$$z_k = H_k x_k + C_k u_k + w_k \tag{3.4.2}$$

where $v_{k-1} \sim \mathcal{N}(0,\ Q_{k-1})$ and $w_k \sim \mathcal{N}(0,\ R_k)$. Solving these two equations recursively will result in Kalman filter [6].

After using properties of Gaussian random variables, one can get the following equations for the Kalman filter:

- *Prediction step* : The state prediction and the covariance prediction can be given by:

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} \tag{3.4.3}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^{'} + Q_{k-1} \tag{3.4.4}$$

  Update step: The state prediction and the covariance prediction can be given by:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - h_k\hat{x}_{k|k-1}) \tag{3.4.5}$$

$$P_{k|k} = P_{k|k-1} - K_k(S_k)^{-1}K_k^{'} \tag{3.4.6}$$

  where S_k is the innovation $(z_k - h_k\hat{x}_{k|k-1}$ is the new information in the measurement) residual:

$$S_k = H_k P_{k|k-1} H_k^{'} + R_k \tag{3.4.7}$$

  and K_k is the Kalman gain and has an important role in making a tradeoff between the measurements and the prior information.

$$K_k = P_{xy}P_{yy}^{-1} = P_{k|k-1}H_k^{'}S_k^{-1} \tag{3.4.8}$$

If the measurement variance R_k is more than the process noise covariance Q_k, then K_k will be less value which implies that we trust more on the prediction than the measurements. Similarly if the process noise is much larger than the measurement variance, then our trust will be more on the measurements than on the prediction, which seems to be intuitively correct and also we can see the same mathematically from (3.4.8) and (3.4.5).

Kalman filter is the MMSE filter for the linear process and measurement models with Gaussian noise and a linear MMSE estimate for linear models with non-Gaussian case [21, 22].

3.4.1 Process model

The general non-linear process model is given by (3.3.2). This model helps to make a prediction to the current time instant based on the previous state and previous measurements. Most of these motion models are mainly driven by the laws of physics. Following are few examples of motion models.

Example. A most frequently used model to describe the vehicle motion us the constant velocity (CV) model, which is based on the assumption that acceleration between two time samples is constant and taken from a zero-mean white noise process. Sometimes this model is called discrete white acceleration model [1]. Consider a simple case where the vehicle is moving along a straight line. Let us assume that the position and velocity of the vehicle at time-step k are given by p_k and $\dot{p}_k$ in some coordinate system. The motion model of the vehicle can be described as:

$$x_k = F\ x_{k-1} + G\ v_{k-1} \tag{3.4.9}$$

where T is the sampling time, $F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ is the state transition matrix and $G = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}$. The effect of noise term $v_k$on the velocity during the time period T is $v_k T$ and on the position is $\frac{T^2}{2} v_k$. So, we can write the equation as $p_k = p_{k-1} + T\ \dot{p_k} + \frac{T^2}{2}\ v_k$ i.e, the position at time instant k depends on the past position plus T times the velocity. Since we can't model the vehicel motion accurately with this model, we add noise term. The process covariance can be computed as

$$Q = E\{G v_k v_k^{'} G^{'}\} = GE\{v_k v_k^{'}\}G^{'} = G\sigma_v^2 G^{'} = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} \sigma_v^2$$

A number of different process models can be seen in [4, 1, 2].

3.4.2 Measurement model

General non-linear measurement model is given by (3.3.3). This model helps to make a link between the state and the measurement at current time. This model is mainly driven by the sensors that one uses for the system. Following are few examples of motion models.

Example. Radar sensor model: Suppose the state vector which we would like to estimate is given in cartesian coordinates as:

$$x_k = \begin{bmatrix} p_{x_k} \\ p_{y_k} \\ \dot{p}_{x_k} \\ \dot{p}_{y_k} \end{bmatrix} \tag{3.4.10}$$

In the Radar system, generally the Radar provides measurements such as range, azimuth angle in polar coordinates:

$$z_k = \begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{p_{x_k}^2 + p_{y_k}^2} \\ arctan(\frac{p_{y_k}}{p_{x_k}}) \end{bmatrix} + \begin{bmatrix} w_r \\ w_\phi \end{bmatrix} \tag{3.4.11}$$

with w_r, w_ϕdefined as the measurement noise in range and azimuth angle respectively. We can see that in this measurement model, z_k is a non-linear function of the state. In Section 3.5, we can see one method of dealing with non-linear measurement model and process model.

3.4.3 Evolution of pdf's

Fig. 3.4.1 gives evolution of the probability density function for a 2D position state vector. The process covariance was 0.01 whereas the measurement variance is 0.5. Hence according to this model, we belive more on the process model. . We can see that the posterior distribution is narrower than the prior and predicted density. Prior information can be seen as the information that we get in the previous recursion when the Kalman filter runs. Once we have the prior information, we use prediction stage of Kalman filter and use the equations of prediction step to get this density function. Once we get the measurements, we can compute the Kalman gain and thereby compute the mean and covariance of the state estimates given the measurements. Note here that prediction density is much flatter than

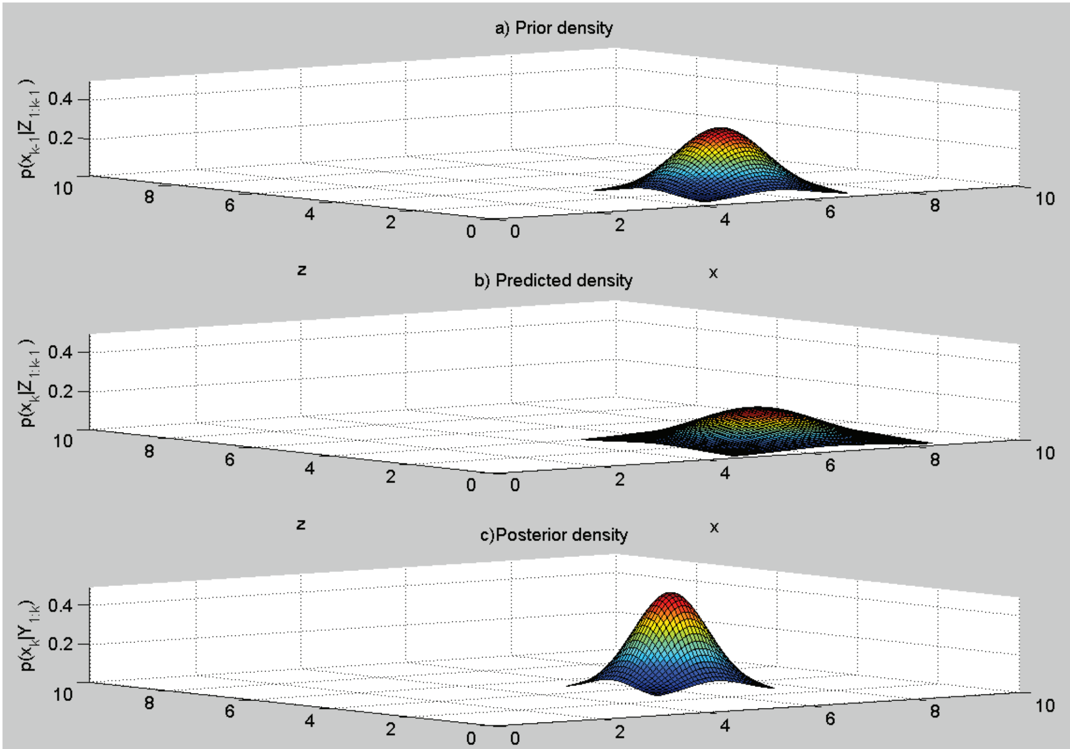


Figure 3.4.1: Evolution of probability density function

the prior density because of the fact that there is some uncertainty that is introduced in the prediction stage.

3.4.4 Trade-off

In FIR filter, there is a tradeoff between the delay and the smoothness in the filtered signal. Similar to these techniques, in Kalman filter, the parameters that have the capability of this tradeoff are process variance and measurement variance. If the process noise covariance is very less, then we believe more in the process model. Similarly if the measurement noise variance is less, then we trust more on the measurements. This means that, for example, if the process noise covariance is less, then the estimated signal will take some time to catch up the measured signal thereby having some delay. These can be seen in the following plots:

3.5 Extended Kalman filter

In some practical problems, the assumptions made by the Kalman filter seems to be very restrictive. For example, in Section 3.4.2, the Radar measurement model came out to be non-linear with respect to the state vector that we have defined. In cases like that, the models are no more linear. Extended Kalman filter (EKF) is one sub-optimal way to extend the Kalman filter setting to handle the non-linear models. Other techniques that are not covered in the thesis include unscented Kalman filter, particle filters etc.

The basic idea in EKF is to linearlise the non linear models using the first order Taylor expansion. Assuming independent white Gaussian noise and using the first order Taylor approximation to the non linear models ensures that the linear Gaussian assumption hold for the linearlised system [1, 3, 18].

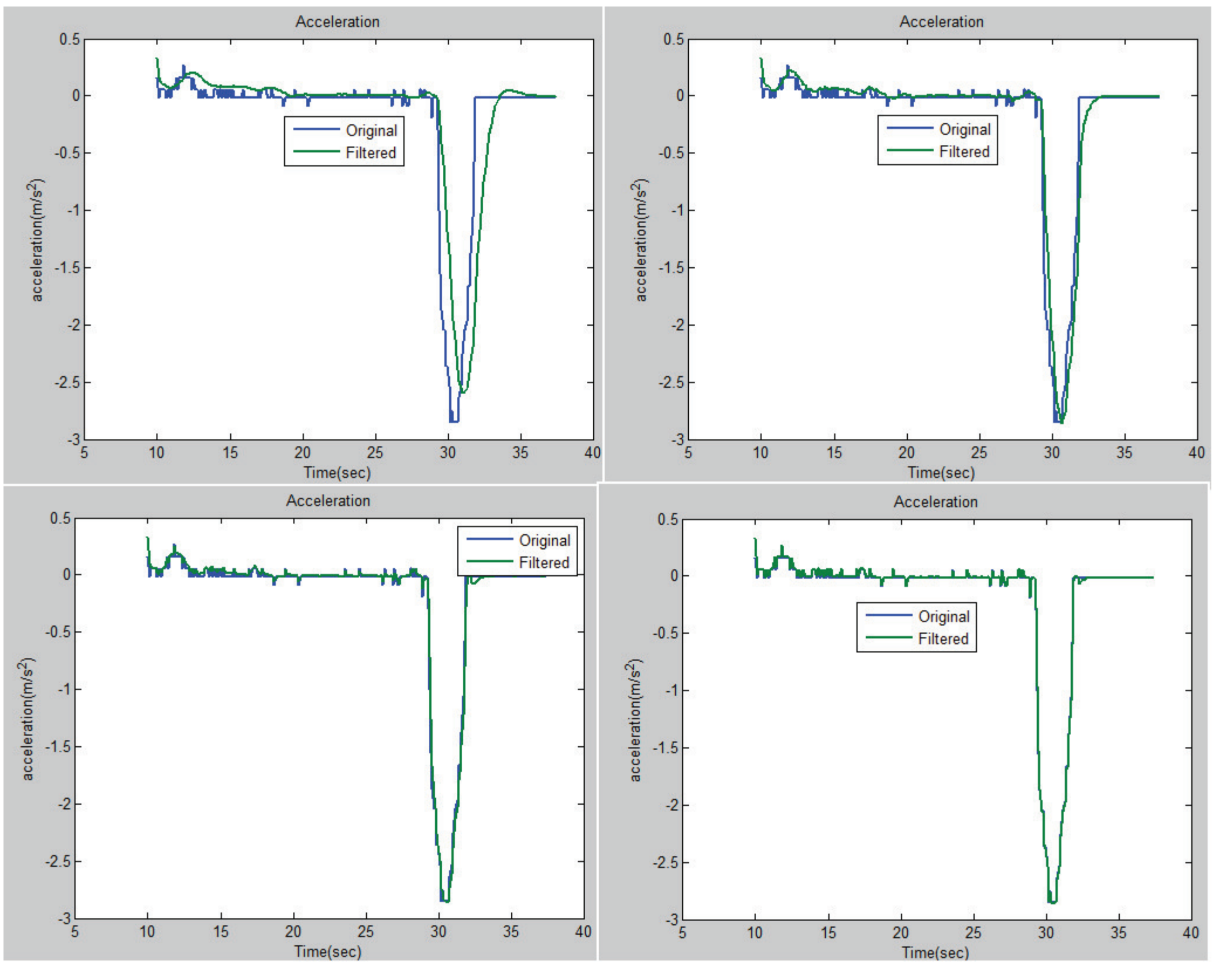


Figure 3.4.2: Varying process noise covariance. top left q=0.01; top right q=0.1; bottom left q=1; bottom right q=10

Let is assume that the following are the process and measurement non-linear models:

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1} \tag{3.5.1}$$
$$z_k = g_k(x_k) + v_k \tag{3.5.2}$$

First order Taylor approximation applied to (3.5.1), (3.5.2) can be computed using:

$$f_{k-1}(x_{k-1}) \approx f_{k-1}(\hat{x}_{k-1|k-1}) + A_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1}) \tag{3.5.3}$$
$$g_k(x_k) \approx g_k(\hat{x}_{k|k-1}) + D_k(x_k - \hat{x}_{k|k-1}) \tag{3.5.4}$$

where $A_{k-1} = \frac{\partial f_{k-1}(x)}{\partial x} |_{x=\hat{x}_{k-1|k-1}}$ and $D_k = \frac{\partial g_k(x)}{\partial x} |_{x=\hat{x}_{k|k-1}}$. Note that $f_{k-1}$is linearlised around the previous estimate $\hat{x}_{k-1|k-1}$and $g_k$is linearised around the predicted value $\hat{x}_{k|k-1}$.

Based on the above equations, the process model and measurement model can be rewritten as:

$$x_k = (A_{k-1}x_{k-1} + d_{k-1}) + w_{k-1} \tag{3.5.5}$$
$$z_k = (D_k x_k + e_k) + v_k \tag{3.5.6}$$

with

$$d_{k-1} = f_{k-1}(\hat{x}_{k-1|k-1}) - A_{k-1}\hat{x}_{k-1|k-1} \tag{3.5.7}$$
$$e_k = g_k(\hat{x}_{k|k-1}) - D_k\hat{x}_{k|k-1} \tag{3.5.8}$$

The goal is to determine the posterior distribution as in the case of any Bayesian filtering problem. The derivation is not presented here but the final equations of the EKF are as below:

Prediction step :

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + d_{k-1} = f_{k-1}(\hat{x}_{k-1|k-1}) \tag{3.5.9}$$
$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A^{'}_{k-1} + Q_{k-1} \tag{3.5.10}$$

Update step :

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - g_k(\hat{x}_{k|k-1})) \tag{3.5.11}$$
$$P_{k|k} = P_{k|k-1} - K_k D_k P_{k|k-1} \tag{3.5.12}$$

where $K_k = P_{k|k-1}D^{'}_k(R_k + D_k P_{k|k-1}D^{'}_k)^{-1}$is the Kalman gain. In the thesis, we have used EKF set u to deal with the non-linear process model and measurement models.

3.6 Sensor fusion

When there are many (more than one) information sources that provide the information about the parameters that we would like to estimate, it would be good to combinee those information sources cleverly. This is what sensor fusion does. It tries to combine information from various sources giving weightings to belief of different information sources. The concept of sensor fusion is not a new concept to the humans. For example, data that comes from the human auditory (ears) and the visual capabilities (eyes) are very different by nature. But the human brain fuses this information to assign a certain sound to a target that one sees visually. The brain seems to create an abstarct representation for each target like pedestrians, vehicles, etc. It is natural to think that while the brain does this fusion, it has to associate objects in temporal sense i.e, relate to the sound and visuals (synchronization) and do a position matching. This is more or less how a fusion system works in general [29].

Example. Fusion of GPS and IMU

A very common fusion that is much prevelant is the fusion of information from global positioning system (GPS) and interial measurement unit (IMU). GPS gives precise absolute location and in differential mode, it can give centimeter level accuracy. However, the GPS information is not reliable in cases when the signal is prone to multipath, if we are under a tunnel etc. It is during this time, the dead-reckoning technique works. But if we just go on predicting the position based on some laws of physics about the movement of the vehicle, the error that will be introduced in the position will drift over time. At this juncture, information from IMU can be of great help. The information from IMU such as velocity, acceleration, steering angle etc can be used to give a good position information in cases not just when the GPS doesn't work but also using IMU will increase the robustness of the system. This is becasuse as we ge more measurements about the same parameter we wish to estimate, the effect of measurements errors can be reduced by taking more and more measurements.

One other common fusion is from GPS and Radar. GPS gives a very low update rate max 20 Hz, the IMU sensors work at much higher rate and Radar works at around 50 Hz. The information from these sources are also fused and can estimate the parameters of interest more accurately.

In the thesis. to get the benefits of the cooperative set up, sensor fusion to improve the accuracy and the reliability of the estimation of the data. Vehicles get information from the communication, internal and external sensors.The information received wirelessely from other vehicles is combined with the information from the on-board sensors.

3.7 Discretization

In the previous chapter, we have seen the discrete-time state space model and derived the Kalman filter equations them. In this chapter, we will see more nonlinear and linear time-continuous state-space models and disretization of these continuous-time models will be presented. Such non-linear state space models are common when the system matrices depend on the current state. For example, if heading angle and velocity are part of the state vector, position changes in Cartesian coordinates have a nonlinear dependence on the heading and velocity, and we end up with a nonlinear dynamic model to describe the target's motion.

3.7.1 Continuous time linear stochastic system

The continuous-time linear stochastic system model is given by:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + D(t)v(t) \tag{3.7.1}$$

$$z(t) = Cx(t) + Eu(t) + w(t) \tag{3.7.2}$$

where $x(t)$ is the state vector of dimension n_x and $u(t)$ is the control input of dimension n_u. The continuous time input disturbance or the process noise$v(t)$ is of dimension n_v. The system matrix is given by A and B, D are the continuous time input gain and noise gain.

Solving first order differential equation like (3.7.1) will result in following [1]

$$x(t) = F(t, t_0)x(t_0) + \int_{t_0}^{t} F(t, \tau)[B(\tau)u(\tau) + D(\tau)v(\tau)]d\tau \tag{3.7.3}$$

where $F(t, t_0) = e^{\int_{t_0}^{t} A(\tau)d\tau}$ and $E\{v(t)v(\tau)^{'}\} = V(t)\delta(t - \tau)$; $x(t_0)$is the initial state. Discretization of (3.7.3) gives [1]:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + D_{k-1}v_{k-1} \tag{3.7.4}$$

where $v_k = \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau)A}Dv(\tau)d\tau$ with zero-mean and white assumption on $v(t)$, it follows that $E[v_k] = 0$ and $E[v_k v_j'] = Q_k\delta_{kj}$

$$Q_k = \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau)A}DV(\tau)D'e^{(t_{k+1}-\tau)A'}d\tau \tag{3.7.5}$$

3.7.2 Continuous time non-linear stochastic system

A more general continuous-time process model is given by (3.7.6).

$$\dot{x}(t) = \alpha(x(t), u(t)) + D(t)v(t) \tag{3.7.6}$$

We would like to disretize this non-linear function in the same way as we have done for the discrete case in Section 3.5 and get a form such as (3.7.1).

Chapter 4

Out of sequence measurement

4.1 Introduction

In a multi-sensor target tracking system, different sensors may operate at different frequencies (sending information at different rates). Due to some internal processing within each sensor system and due to the delays in the communication network, the processing/fusion centre may receive information from different sensors with different delays. When a discrete-time filter say Kalman filter is used for target tracking, care has to be taken while processing the delayed data that is entering the filter in order to obtain optimal estimate of a target's state [7]. If the proper processing of OOSM is not given enough consideration, the quality of the state estimates can be degraded when compared to the quality of the state estimate provided by a single sensor [7].

An example of l-step lag OOSM measurement is shown in Fig. 4.1.1 where the filter has processed information till $z_k$and later a measurement that has time stamp $z_d$arrives to the processing center. There has been a lot of interest in dealing with OOSM and there are many research papers in this area. The algorithms can be broadly classified as 1-step lag (single-lag) and $l-$step lag (multiple-lag) algorithms. The single-lag OOSM algorithm is applicable when the OOSM z_d observed at time t_d lies between the latest two measurements. The $l-$step lag OOSM algorithm is applicable for the general case when the OOSM z_d can have an arbitrary time delay and thus lies between measurements z_{k-l} and $z_{k-l+1}$and observed at time after $z_k$and before z_{k+1}. Some of the papers addresses issues like data association, clutter environment, hypothesis management which are very common issues in any multi-target multi-sensor target tracking system [8].

In order to process this out of sequence measurement, there are many techniques existing in the literature. Some of them will be described below. Note that in this chapter, a small change is made in the notations to make the concepts of OOSM clear. For example the state-tranition matrix at time k was written as F_k in previous chapters. But in this chapter, to make things clear, we include the time of transition explicitly that is $F_{k,k-1}$ indicating that this transition matrix is for the time interval from $k-1$ to k.

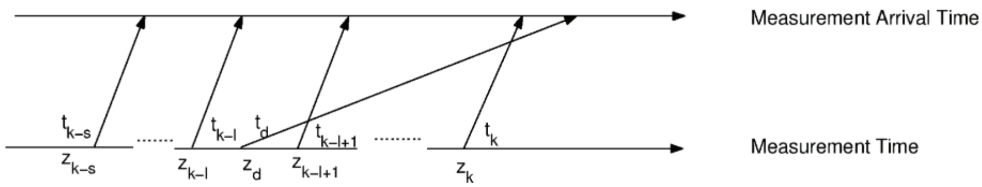


Figure 4.1.1: OOSM z_d arrives after a later measurement $z_k$is processed

4.2 Basic approaches to handle OOSM

4.2.1 Neglect

A simple naive way of handling OOSM is to just neglect the OOSM itself. Though it is a very simple idea, in some applications, the delayed information is much more precise and accurate than the information that comes on time . So, neglecting is desired when the delayed data is more accurate and contains different set of information that updates part of the state vector. If information from each sensor is useful for updating different part of the state vector, it will be not at all desired to neglect the delayed data.

4.2.2 Reprocessing

In Fig. 1, the once the filter has computed the estimates of the state for time instant k, and later when the measurement which is l-time step lag is received like z_d, the filter can go back to $z_{k-l}$and reprocess information from there on. By this reprocessing, one gets the exact solution but for this process to work, we need to store all the measurements from time instant just before OOSM till the latest update time of the filter. In a normal tracking systems, where the system has to do tasks such as data association, track management., doing these steps again for each measurement may be a computationally demanding task. .

4.2.3 Buffering

Depending on the application and the timing constraints, if the processing center is aware of the OOSM and measurement time, then it can buffer all th einputs from the OOSM measurement time till the time measurement has arrived. (in Fig. 4.1.1, all measurements from z_{k-l+1} till $z_k$will be buffered). As and when the OOSM measurement has arrived, the processing center can now go ahead and process the OOSM and buffered data in sequence. This is a good way to do if the processing center doesn't have any space considerations to store all these measurements and also if the application which needs these processed inputs can wait that long.

4.3 Algorithms that don't need reordering and reprocessing the measurement

This is the main area where most of the work regarding OOSM has been done both for 1-step/single -lag problems and l-step lag problems.

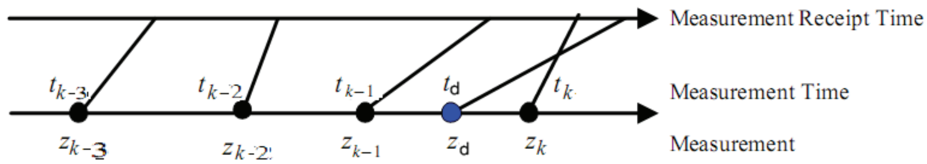


Figure 4.3.1: The out-of-sequence measurement z_d observed at time t_d arrives after the last processed measurement z_k observed at time t_k . Since z_d lies between the measurements z_{k-1} and z_k , z_d is a single-lag out-of- sequence measurement by convention [28].

4.3.1 Update with One-step-lag OOSM algorithms

The time-discrete linear version of process model and the measurement model can be written as:

$$x_k = F_{k,k-1}\, x_{k-1} + v_{k,k-1} \tag{4.3.1}$$
$$z_k = H_k\, x_k + w_k \tag{4.3.2}$$

where x_i is the state vector of the system at time t_i, $F_{k,k-1}$ is the state transition matrix from time $t_{k-1}$to time t_k. The process noise for this interval is denoted by $v_{k,k-1}$ and w_k is the measurement noise. The measurement matrix is given by $H_k$and z_k is the measurement vector at time k (k=0, 1, ..). The process noise and the measurement noise are assumed zero-mean, white and mutually independent.

The time t_d, at which the OOSM was made, is assumed to be such that $t_{k-1} < t_d < t_k$as shown in Fig. 4.3.1 and one can write

$$x_k = F_{k,d}\, x_d + v_{k,d} \tag{4.3.3}$$

At time $t = t_k$, one has

$$\hat{x}_{k|k} = E[x_k|\mathbf{Z}_{1:k}] \quad and \quad P_{k|k} = cov[x_k|\mathbf{Z}_{1:k}] \tag{4.3.4}$$

where $\mathbf{Z}_{1:k} = \{z_1,\ z_2 \ldots\ldots\ldots\ldots z_{k-1}\ z_k\}$. Given this definition, the OOSM

$$z_d = H_d\, x_d + w_d \tag{4.3.5}$$

which arrives after the state estimate (ref above equation) has been calculated at time k. Our goal is to update this state estimate obtained at $t_k$with an earlier measurement z_k i.e, we want to calculate:

$$\hat{x}_{k|d} = E[x_k|\mathbf{Z}_{1:k}, z_d] \quad and \quad P_{k|d} = cov[x_k|\mathbf{Z}_{1:k}, z_d] \tag{4.3.6}$$

We want to do this update without any reordering and reprocessing of the measurements.

The exact solution for the 1-step lag is proposed in [9] and is described below. The algorithm is similar to the normal predict-update cycle of the Kalman filter except that in this case, since we have to update with a measurement which is older than the current state update, we retrodict back to that state estimate first and then update with that delayed measurement. Obtain the retrodicted state $\hat{x}_{k|d}$ from $\hat{x}_{k|k}$ using $x_k = F_{k,d}\, s_d + v_{k,d}$ as

$$\hat{x}_{d|k} = E[x_k|\mathbf{Z}_{1:k}] = F_{d,k}\, E[x_k - v_{k,d}|\mathbf{Z}_{1:k}] \tag{4.3.7}$$
$$= F_{d,k}\, [\hat{x}_{k|k} - \hat{v}_{k,d|k}] \tag{4.3.8}$$

1. Evaluate the corresponding covariance
2. Calculate the filter gain for updating the state $x_k$with measurement z_d using $z_d = H_d\ x_d + w_d$
3. Update the state estimate $\hat{x}_{k|k}$ to $\hat{x}_{k|d}$ and calculate the corresponding covariance.

The only difference between [9] and the papers that existed before [7, 2] is that the retrodicted noise (second term in $F_{d,k}\ [\hat{x}_{k|k} - \hat{v}_{k,d|k}]$). Let's call algorithm for 1-step lag defined by Hilton [7], which is similar to algorithm B of [9] as "B1" algorithm and optimal 1-step lag algorithm defined by Bar-Shalom[9] as "A1" (). It is shown in [9] that although B1 is sub-optimal, it is a very simple algorithm as it need not calculate the retrodicted noise term which includes finding the cross-covariances.

4.3.2 Update with l-step-lag OOSM algorithms

In $l-$step lag OOSM problem, the time t_d, at which the OOSM was made, is assumed to be such that $t_{k-l} \leq t_d < t_{k-l+1}$as shown in fig 1. In [26], this general l-steplag problem was solved in framework of algorithm B1, i.e, using the linear retrodiction

$$\hat{s}(d|k) = F_{d,k}\ \hat{s}(k|k) \tag{4.3.9}$$

and this algorithm is called "Bl" as an extension to algorithm B1.

Algorithm	Author & Paper	Calculation Steps	Exact	Storage
Bl	Mallick[26]	l	No	High
Al1, Bl1	Bar-Shalom[10]	1	No	Low

In [27], the authors have defined 3 different scenarios depending on the knowledge of the delayed sample time $t_d$and proposed algorithms for solving l-step lag problem accordingly.

Initially during the start of thesis, it was predicted that the communication delays would be around 500 msec. But when we did tests, it turned out that the average communication delay from V2V is around 10 msec and without any OOSM algorithms, the system is giving good performance. However, with the logged data, we have tested using reprocessing when the packets are delayed more and it doesn't improve performance much.

Chapter 5

Implementation

5.1 Introduction

We have information from

- GPS of ego vehicle about position, velocity, heading, acceleration of ego vehicle
- GPS of target vehicle: target vehicle's position, velocity, heading, acceleration
- Radar of ego vehicle: range (distance between ego vehicle and target vehicle), range rate

As said in previous chapters , the task of the sensor fusion is to fuse the information from these sensors and estimate the

- relative distance between the ego vehicle and preceding vehicle
- ego vehicle heading, speed and acceleration
- preceding vehicle speed and acceleration.

$Data\ pre-processing$:The latitude, longitude information from ego vehicle's GPS and preceding vehicle's position information are converted to global coordinates system using Universal Transverse Mercator (utm) model. When the euclidean distance is taken between these two GPS positions, then we can get the estimate of the distance between the two vehicles. However, since the mounting position of the GPS's on both vehicles is also included, the distance that we get by directly taking GPS's euclidean distance is not the same as reproted by the Radar. So, this mounting position is deducted from the GPS positions so that now this value can be compared with the radar measurements.

As a first step, we need to find out the state vector. This state vector typically contains the elements to be estimated or elements from which we can get the information that we are interested in. After defining the state vector, we need to write down the process model for the application of interest. For this, we need to define the elements of state vector, state transition matrix and the process noise.

5.2 Discrete-time state space model

5.2.1 State vector

As we are interested in filtering the position, velocity, acceleration, heading of ego vehicle, we have added those elements in the state vector. The state vector at time instant k is defined as follows:

$$s_k = [x_k^1 \, y_k^1 \, \theta_k \, v_k^1 \, a_k^1 \, d_k^2 \, v_k^2 \, a_k^2]^T,$$

where the state vector at time instant k is defined by s_k. The position of ego vehicle in global coordinate system is given by $(x_k^1 \, y_k^1)$. The inclination angle (angle from North pole) is defined by θ_k. The speed and acceleration of the ego vehicle is denoted by $(v_k^1 \, a_k^1)$; speed and acceleration of the target vehicle by $(v_k^2 \, a_k^2)$. The distance between ego vehicle and target vehicle is denoted by d_k^2 .

5.2.2 Process model

Once we define the state vector, we are ready to define the process model. Process model gives the evolution of the state vector over time. Given current state vector, we can predict what will be the values of the state vector in future based on motion model of the vehicle.

$$\begin{aligned}
\begin{bmatrix} x_{k+1}^1 \\ y_{k+1}^1 \\ \theta_{k+1} \\ v_{k+1}^1 \\ a_{k+1}^1 \\ d_{k+1}^2 \\ v_{k+1}^2 \\ a_{k+1}^2 \end{bmatrix} &= \begin{bmatrix} x_k^1 + v_k^1 * T * cos(\theta_k) + a_k^1 * \frac{T^2}{2} * cos(\theta_k) \\ y_k^1 + v_k^1 * T * sin(\theta_k) + a_k^1 * \frac{T^2}{2} * sin(\theta_k) \\ \theta_k \\ v_k^1 + a_k^1 * T \\ a_k^1 \\ d_k^2 + (v_k^2 - v_k^1) * T + (a_k^2 - a_k^1) * \frac{T^2}{2} \\ v_k^2 + a_k^2 * T \\ a_k^2 \end{bmatrix} + noise \qquad (5.2.1) \\
\implies s_{k+1} &= f_k(s_k) + n_k \qquad (5.2.2)
\end{aligned}$$

5.2.3 Linearization

Since the process model is non-linear function of the state vector, we can use algorithms like particle filter or extended Kalman filter or unscented Kalman filter. We choose to work with extended Kalman filter....

So, for using the extended kalman filter, first we need to linearlize (5.2.1) to get it of the following form:

$$s_{k+1} = A_k * s_k + d_k + w_k$$

where
$d_k = f_k(s_k^k) - A_k * s_k^k$ acts as a (control) input to the filter and

$$\begin{aligned}
A_k &= \frac{\partial f_k(s)}{\partial s}|_{s=s_k^k} \\
&= [\frac{\partial}{\partial x^1} \frac{\partial}{\partial y^1} \frac{\partial}{\partial \theta} \frac{\partial}{\partial v^1} \frac{\partial}{\partial a^1} \frac{\partial}{\partial d^2} \frac{\partial}{\partial v^2} \frac{\partial}{\partial a^2}]s
\end{aligned}$$

A_k is the state transition matrix in the discrete domain.

$$M_k = \begin{bmatrix} 1 & 0 & -v_k^1 * T * sin(\theta_k) - a_k^1 * \frac{T^2}{2} * sin(\theta_k) & T * cos(\theta_k) & \frac{T^2}{2} * cos(\theta_k) & 0 & 0 & 0 \\ 0 & 1 & v_k^1 * T * cos(\theta_k) + a_k^1 * \frac{T^2}{2} * cos(\theta_k) & T * sin(\theta_k) & \frac{T^2}{2} * sin(\theta_k) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -T & -\frac{T^2}{2} & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and A_k can be found using $A_k = M_k|_{s=s_k^k}$

5.3 Continuous-time state space model

Alternatively, as described in Section 3.7, the state-space can be modelled in continuous time. The idea is to first write the state equation in continuous time differential form then linearize the equation since it is of non-linear form. Once we get a nice linear continuous time state equation, we can discretize this equation to get discrete time state equation.

$$d\begin{bmatrix} x^1(t) \\ y^1(t) \\ \theta(t) \\ v^1(t) \\ a^1(t) \\ d^2(t) \\ v^2(t) \\ a^2(t) \end{bmatrix} = \begin{bmatrix} cos(\theta(t)) * v^1(t) \\ sin(\theta(t)) * v^1(t) \\ 0 \\ a^1(t) \\ 0 \\ v^2(t) - v^1(t) \\ a^2(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dB_{\theta(t)} \\ dB_{a^1(t)} \\ dB_{a^2(t)} \end{bmatrix} \tag{5.3.1}$$

$$ds(t) = f(s(t))dt + G * dB(t) \tag{5.3.2}$$

$$= f(\hat{s}) + (\nabla f(\hat{s}(t)))^T (s(t) - \hat{s}(t)) + G * dB(t) \tag{5.3.3}$$

$$A(t) = \nabla f(\hat{s}(t)))^T = [\frac{\partial}{\partial x^1} \frac{\partial}{\partial y^1} \frac{\partial}{\partial \theta} \frac{\partial}{\partial v^1} \frac{\partial}{\partial a^1} \frac{\partial}{\partial d^2} \frac{\partial}{\partial v^2} \frac{\partial}{\partial a^2}]s$$

$$A(t) = \begin{bmatrix} 0 & 0 & -sin(\theta(t)) * v^1(t) & cos(\theta(t)) & 0 & 0 & 0 & 0 \\ 0 & 0 & cos(\theta(t)) * v^1(t) & sin(\theta(t)) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.3.4}$$

$$A_k = A(t)|_{s=s_k^k} \tag{5.3.5}$$

5.3.1 Discrete time state transition matrix

Once we have found the state traansition matrix in continuous-time domain, we can find the discrete-time state transition matrix as:

$$F_k = e^{AT} = I + A_k * T + A_k^2 * \frac{T^2}{2} + ...$$

A_k^3 onwards is zero in our case.
So

$$F_k = \begin{bmatrix} 1 & 0 & -sin(\theta_k) * v_k^1 * T & cos(\theta_k) * T & \frac{T^2}{2} * cos(\theta_k) & 0 & 0 & 0 \\ 0 & 1 & cos(\theta_k) * v_k^1 * T & sin(\theta_k) * T & \frac{T^2}{2} * sin(\theta_k) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -T & -\frac{T^2}{2} & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3.6}$$

Where T is the difference between the samples at time $t_{k+1}$and t_k.
The process noise is evaluated using (3.7.5).
Since the state vector contains the position information from GPS in cartesian coordinate system and also connects it to the polar coordinate system of the Radar, the measurement models are all linear because of these elements in state vector.

5.3.2 Process noise

To have the process model to be complete, we need to define the noise term n_k and corresponding covariance. If the process model is well-known like constant-velocity (CV), then we can get the process models from the literature and the corresponding covariances. However, if the process model is not known or if we want to derive it ourselves, then it is a good idea to start with continuous time domain so that we can get the different entities of the process noise covariance.The process noise is derived as given in Section 3.7 and specifically in (3.7.5).

Chapter 6

Test results

The current sensor fusion system is developed on an incremental basis. First GPS information of the ego vehicle is fused with the in-vehicle sensors information. Once the simulation results were satisfactory, the system was run in real-time in the car and the data is logged. Parameters were fine-tuned so that the system gives good performance in the particular area of application. Later preceding vehicle information from V2V was also fused to the existing system and tested in real-time. Lastly, the leader vehicle information from V2V was also filtered by using a separate Kalman filter set up. Some results which explain the capability of the sensor fusion system are discussed below.

6.1 Preceeding vehicle

Fig. 6.1.1 shows the velocity of the preceding vehicle with top figure that shows the V2V measurement and the measurement from the radar unit of the ego vehicle. Eventhough both the sensors should convey the same information i.e, about the preceding vehicle speed, we can see from the figure that V2V information was always zero which doesn't match with the velocity that radar/CAN unit is providing. In the bottom figure, the result of fusing the information from these two different sources is given. It seems that the sensor fusion system has relied mostly on the CAN measurement in this case ignoring the V2V information.

Contrary to Fig. 6.1.1, Fig. 6.1.2 tells a different story. In Fig. 6.1.2, CAN measurement was almost constant during the period 220-420 seconds whereas the V2V was providing with some measurement. The bottom plot of Fig. 6.1.2 also has the estimation result provided by the sensor fusion system. In this case, we can see that the sensor fusion system seems to have followed V2V information as opposed to radar information in Fig. 6.1.1. If one closely observes the estimated and the V2V information in Fig. 6.1.2, especially around times 210., 270, 240, 410 sec, the estimation didn't follow the dips. These dips are due to the packet loss in V2V information. So, we can see that even when we have packet loss, due to the process model in the sensor fusion system, the estimation doesn't follow the dips and neglect those measurements as outliers.

Fig. 6.1.3 shows the velocity of the preceding vehicle when both sources of information are good. In this case, as one expects the sensor fusion system gives the best fused information.

In Fig. 6.1.4, relative distance between the ego vehicle and the preceding vehicle is shown. The two main direct sources of information for the relative distance are radar and the position information from V2V. As explained in the properties of radar in Section 2.2.2, the radar has a limited field of view and was also not reporting stationary objects. This plot is a case when there was a GCDC vehicle in between two lanes and Chalmers vehicle was just behind the GCDC lead vehicle. The bottom plot of Fig. 6.1.4 shows the center in path vehicle information and that value is 1 when the radar detects a vehicle in front and zero when the radar thinks that there is no vehicle in front. So, sometimes the radar was not able to detect any object because the field of view was not sufficient as can be seen in

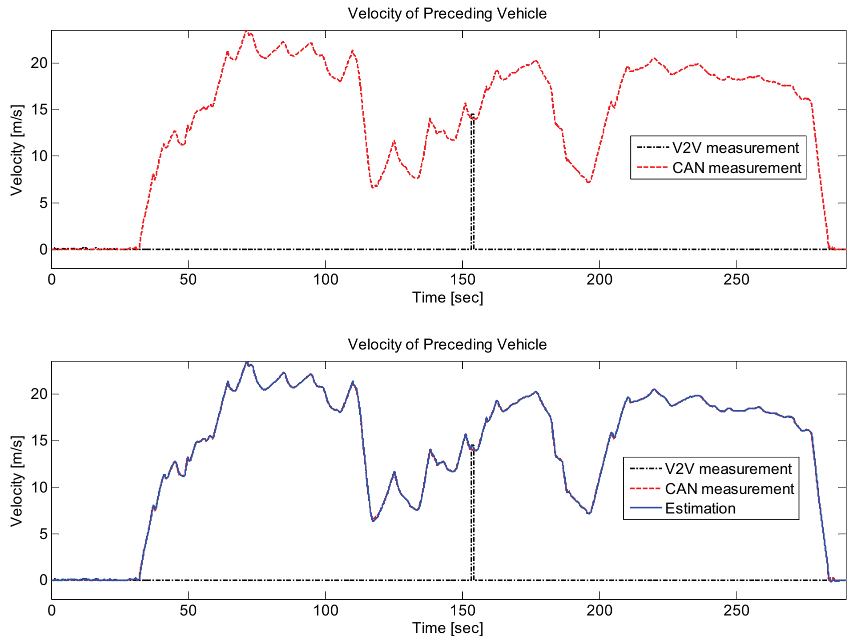


Figure 6.1.1: Velocity of preceding vehicle: bad V2V information

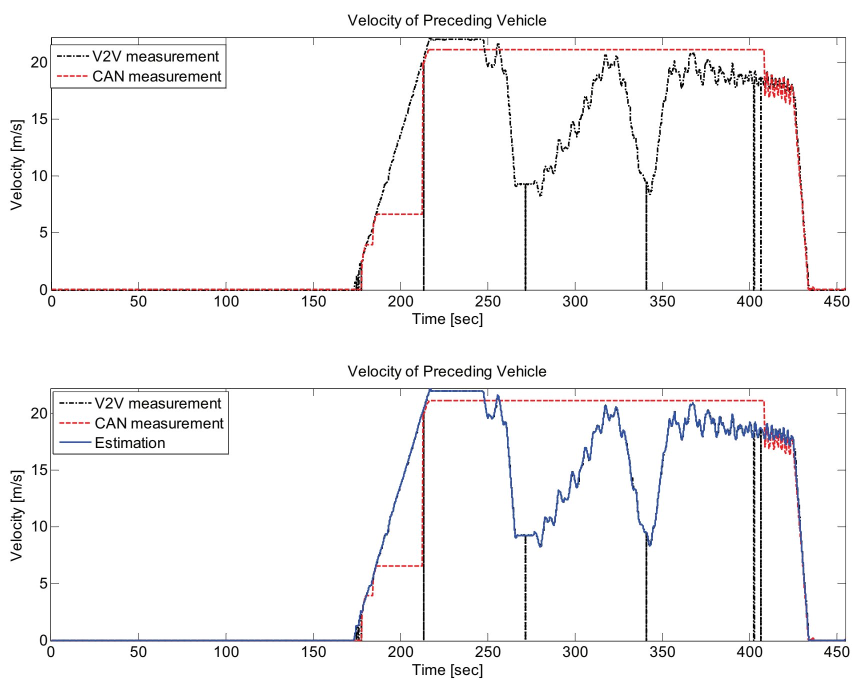


Figure 6.1.2: Velocity of the preceding vehicle: bad radar

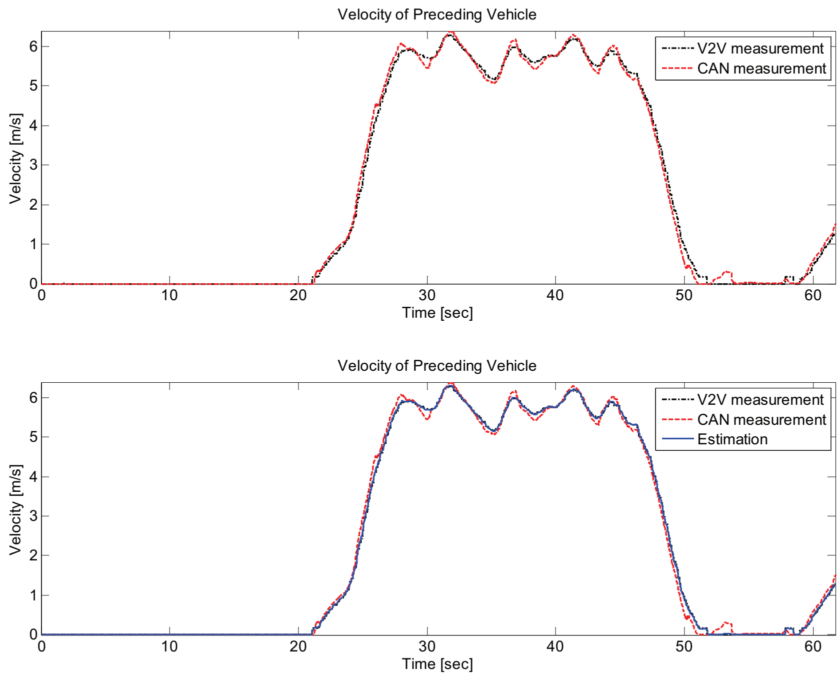


Figure 6.1.3: Velocity of the preceding vehicle: both sources of information good

the bottom plot of Fig. 6.1.4. However since the V2V information need not have any field of view for the information to get communicated, the information from V2V was very helpful in determining the relative distance.

6.2 Ego Vehicle

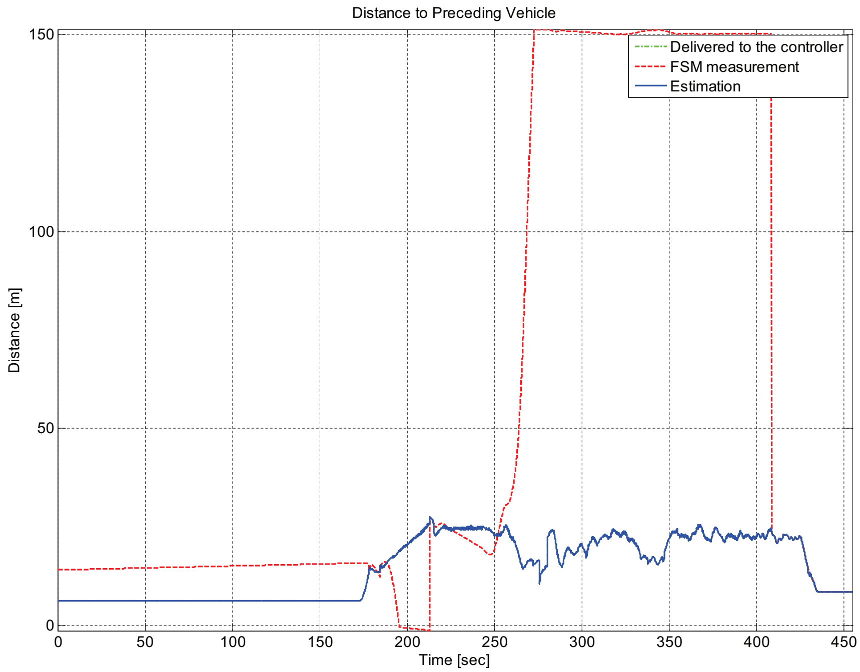
As described earlier, the heading reported by the GPS that we have used is very bad at low speeds as can be seen in initial part of Fig. 6.2.1. The plot also shows the estimated heading which is almost constant when the vehicle is at still or at very very low speeds.

Fig. 6.2.2 shows the ego vehicle's acceleration-both measurement and the estimated signal. We can see from the estimation that it is much smoother and less noisier than the actual measurement, which is desired.

Top plot in Fig. 6.2.3 shows the position of the ego vehicle from the GPS and the estimated position. Bottom plot shows the GPS status of the ego vehicle which tells whether the GPS signal is reliable or not. We can see that in this test, most of the time, the GPS status was flucutuating between 0 (GPS signal bad) and 1 (GPS signal good), however, the estimated signal is not fluctuating as the measurements. This test case was not in the GCDC competition but during the test in Gothenburg. We fixed the problem of the GPS and later the GPS status was mostly 1.

6.3 Leader vehicle

As told earlier, besides filtering the information from preceding vehicle, information from the leader is also filtered. Fig. 6.3.1 shows the actual measurement and trhe estimated acceleration of the leader. One can see that the estimated signal is much smoother and less noisier than the measurement.



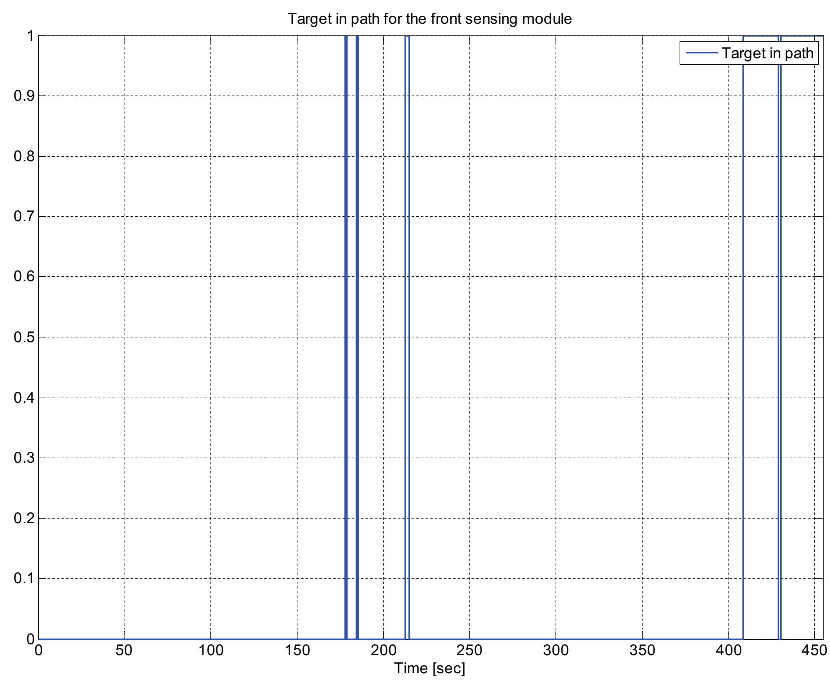


Figure 6.1.4: Distance to the preceding vehicle (top) and target present signal reported by Radar (bottom)

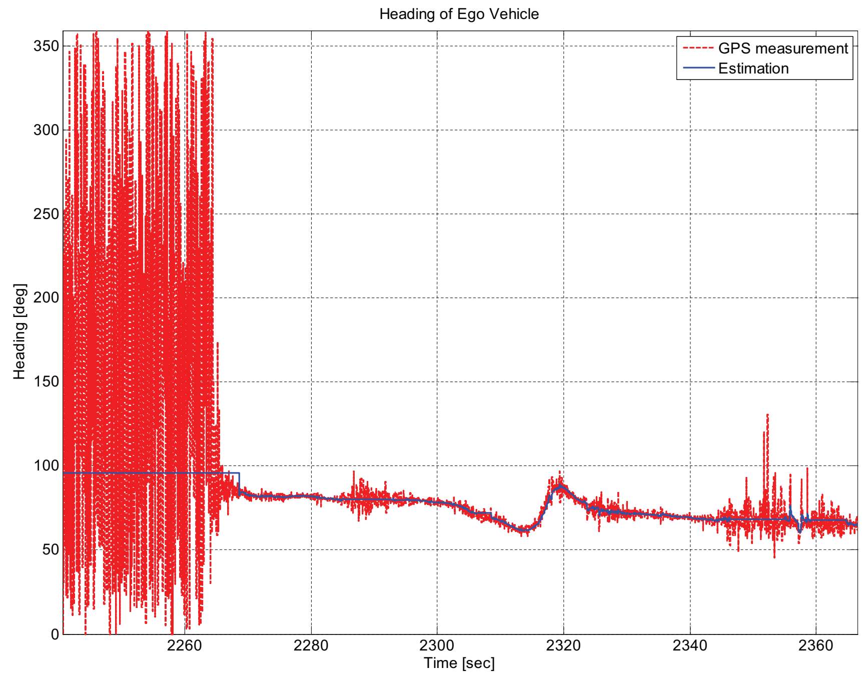


Figure 6.2.1: Heading of ego vehicle

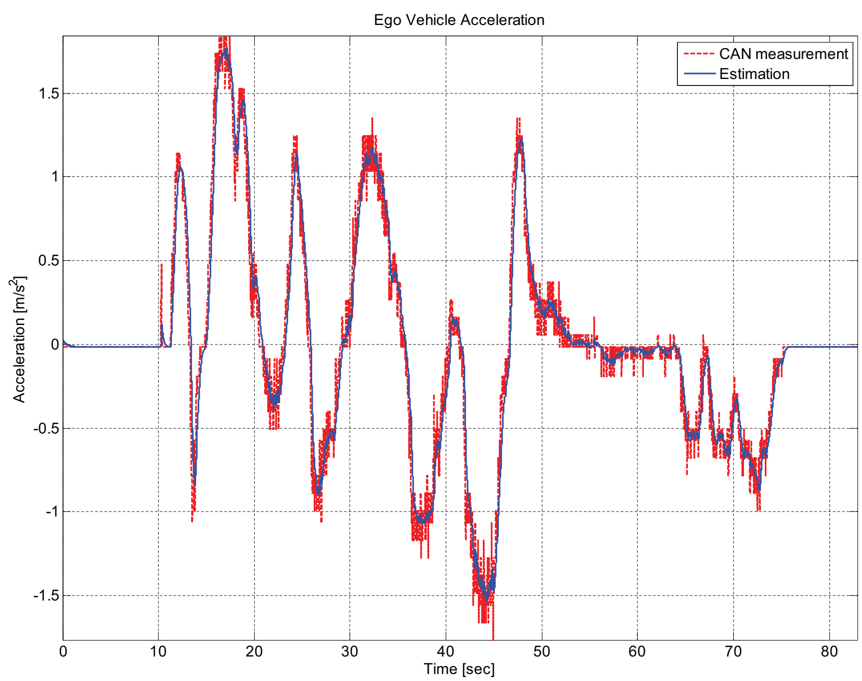


Figure 6.2.2: Ego vehicle acceleration

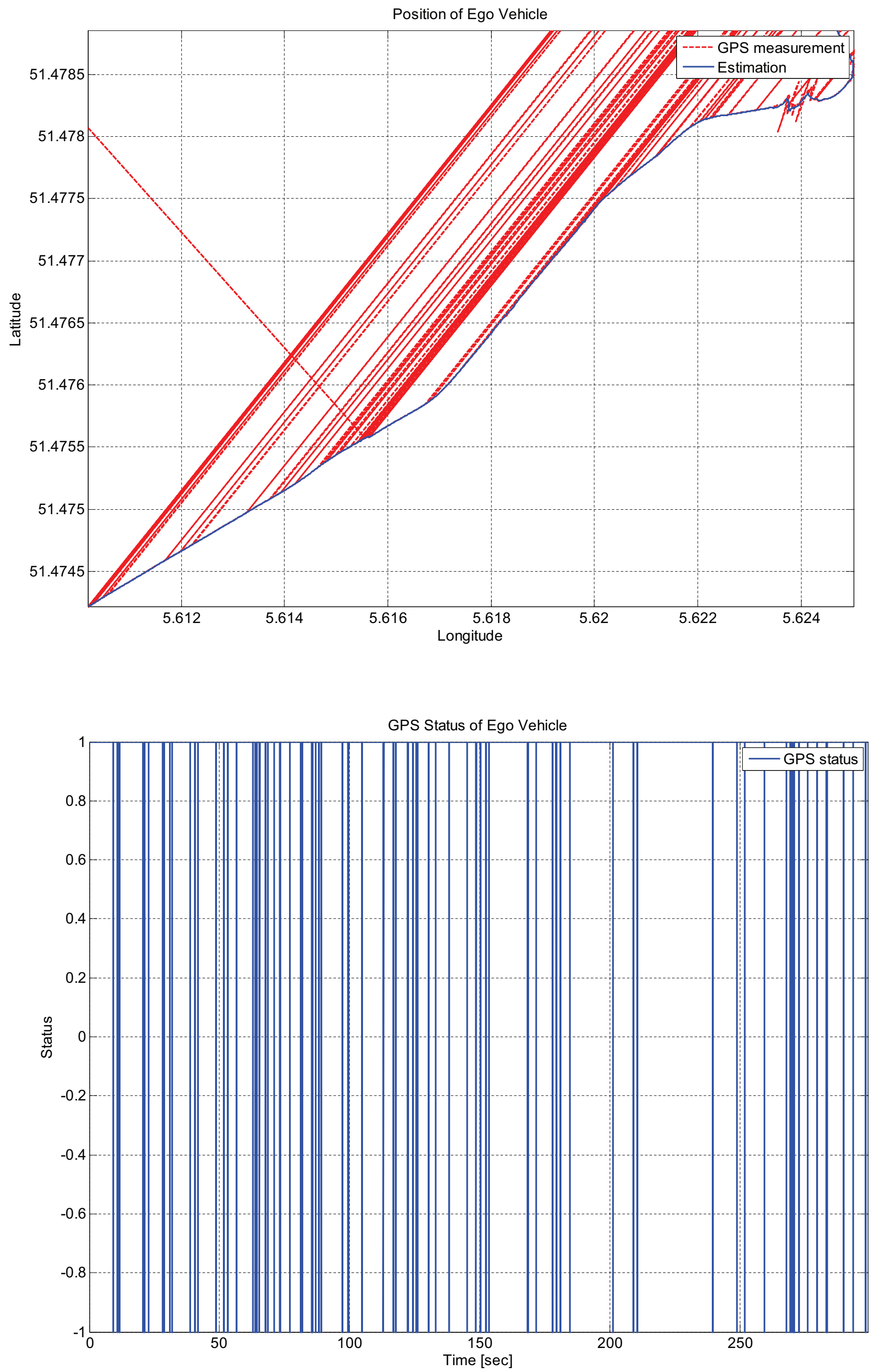


Figure 6.2.3: Ego position and GPS status (0 means data not good and 1 means good)

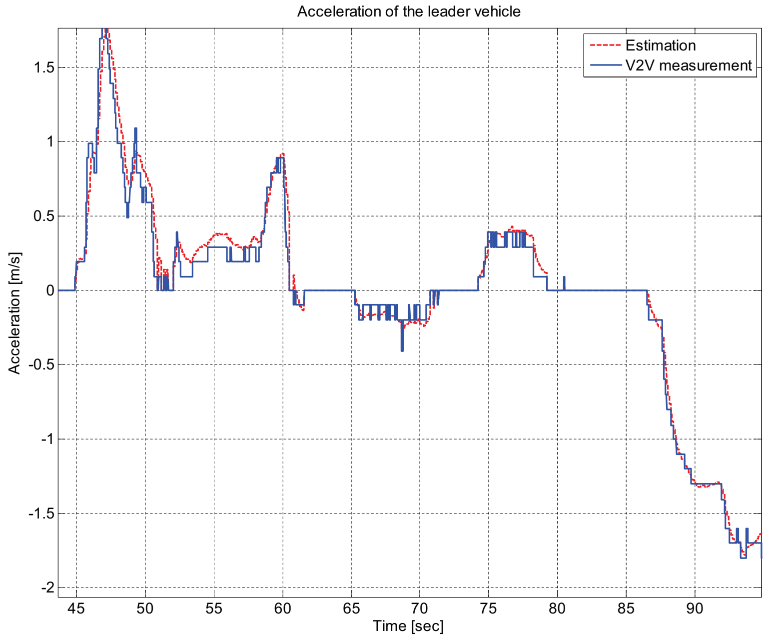


Figure 6.3.1: Acceleration of the leader vehicle

Fig. 6.3.2 shows the speed of the leader vehicle. One can see that there are many dips which indicate the loss of communication information during these periods. However, the filtering algorithm has taken care by not following the dips. This is possible because of the process model of the Kalman filter.

6.4 Problemetic plots

Fig. 6.4.1 has a spike around 60 seconds which is not desired according to the profiles of the other vehicles. Sensor fusion was also not able to delte this spike since in the sensor fusion, high weightage is given to the measurement of the velocity from the CAN as it seems to be a filtered one. So, as can be seen in the bottom figure and compare it with the top, we can see that the sensor fusion output was the same as what was sent by the CAN. So, may be a little bit more modeling woudl have been removed this problem also.

In Fig. 6.4.2, if one observes closely, we can see small spikes around 190, 260 seconds. After analysing the data, I realised that this spikes are because of the bad GPS measurements during that times. This could have been avoided if the GPS velocity has been given higher variance compared to the velocity from the CAN measurement.

The figures in this chapter convey that the sensor fusion algorithm was performing as desired. Filtering out the noisy part in the signal received. Complementing signals of Radar, V2V, GPS whenever one is not available. In this way, the limitations of one sensor is covered by the other sensor.

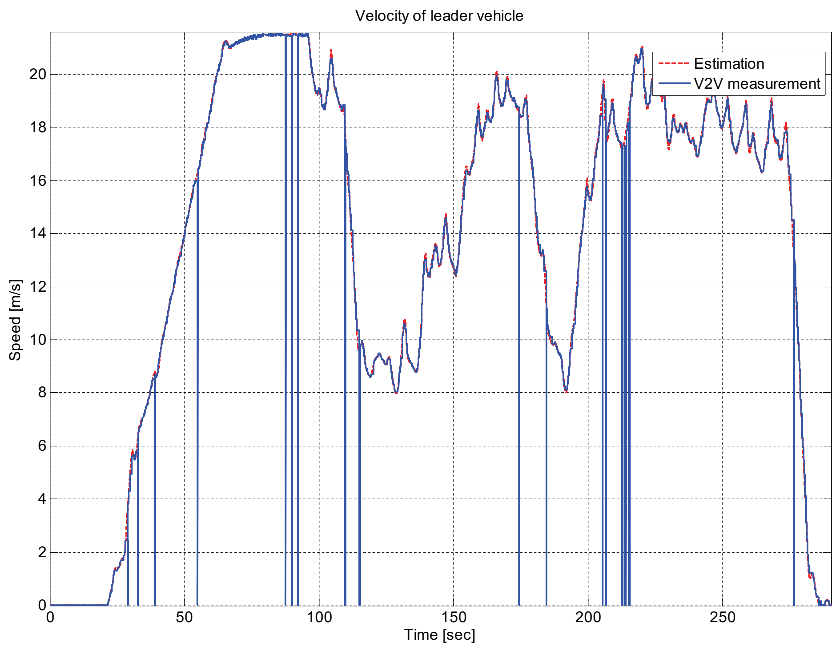


Figure 6.3.2: Speed of leader vehicle

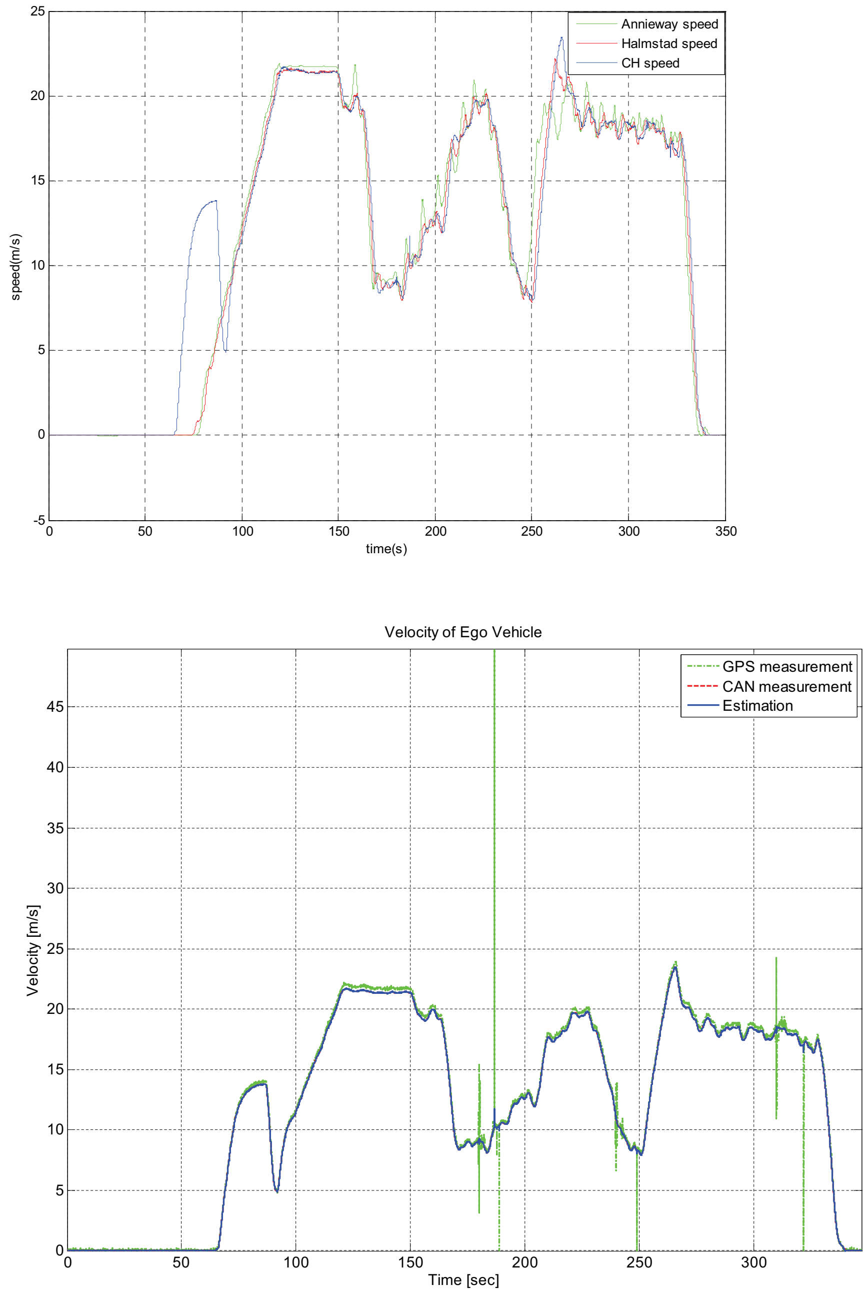


Figure 6.4.1: Speed profiles of different vehicles (top) and speed of ego vehicle from different sources (bottom) for the same test

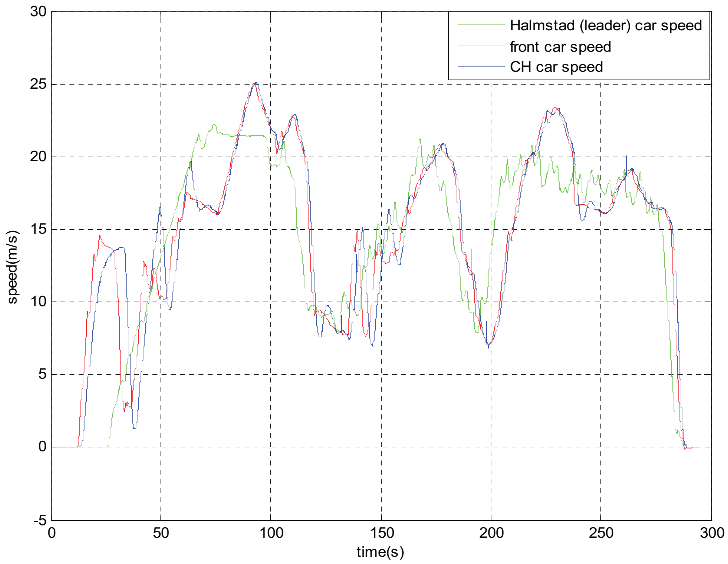


Figure 6.4.2: Speed profile of different vehicles in different test.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The results from previous chapter shows that the sensor fusion system was working fine and was filtering the noisy data and also fusing the information from different sources. When the radar doesn't detect vehicles, then using the V2V and the process model, we were able to estimate the parameters of interest as can be seen in Fig. 6.1.4 and Fig. 6.1.2. Similarly when V2V information is bad/zero, even in this case the fusion system was giving good estimates as can be seen in Fig. 6.1.1. As said earlier, the GPS gives a bad heading information at low speeds but the sensor fusion system has taken care of this as can be seen in Fig. 6.2.1. The power of process model and dead-recknoning system can be seen in Fig. 6.2.3. Even if the GPS was giving many bad values, the sensor fusion system is able to estimate the position of the estimates. From the above fugures, it seems taht the sensor fusion system which is develped for robustness and as a fail-safe design has acheived its goal.

In a bigger perspective if we talk about the whole GCDC project by Chalmers team: In the test during GCDC competition, we have tried to decrease the headway time if we have known that the information from platoon leader and preceding vehicles is good. By this way, we have tested the possibility of using platooning for traffic congestion problem by using cooperative setup. If headway time can be decreased without sacrificing the safety, then more vehicles can cross a traffic light, more throughput on highways. In the GCDC competition, we were also able to successfully dampen the acceleration of the preceding vehicle which is very important for the string stability problem. In this sense, we can say that we have achieved a great sense of satisfaction from the project that we were able to test the system with what we have planned to do.

7.2 Future work

A lot of work has to be done in the theoretical side of the cooperative sensor fusion systems.

There are atleast three different approaches how one can do fusion in vehicular networks set up where each vehicle exchanges information. The optimal approach is to exchange the actual measurements and their uncertainties. That is each vehicle has access to all the measurements with their uncertainties and each vehicle can run their own algorithms and can get the best result. Since there are lot of measurements that are involved in the exchange, the computational and storage complexity will be very very high. A second way to do this fusion is a more practical way: exchanging the posterior density functions. Using this way, we more or less get a good solution, even though not optimal, can give good acceptable results. A third way to do cooperative senor fusion is using factor graphs. So, one future work can be analysing the complexity and gain in these three techniques. When we want to use factor graphs approach for cooperative sensor fusion problem, may be it would be interest also to see OOSM problem using factor graphs.

Regarding the fusion of information, there are lot of questions as to which is good-centralised fusion or de-centralised fusion? The information that is exhanged between the vehicles-should it be sent in a raw form (actual measurements) or should it be fused in each vehicle and send only the fused estimates? The advantages and disadvantages of each method has to be explored. Also, one important question to think is what is the maximum delay that can be beared after which fusing information from that delayed data is not useful?

These are some of the future directions in the theoretical side for the cooperative sensor fusion for vehicular networks according to our view.

Bibliography

[1] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T, “Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software”, New York: Wiley, 2001

[2] S. S. Blackman and R. F. Popoli, “Design and Analysis of Modern Tracking Systems”, Artech House, 1999.

[3] Fredrik Gustafsson, “Statistical Sensor Fusion”, Studentlitteratur AB, 2010.

[4] X.R. Li and V.P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models", IEEE Trans. Aerosp. Electron. Syst., vol. 39, no. 4, pp.1333 - 1364 , 2003.

[5] Rules & Technology Document, [Online]. Available at: http://www.gcdc.net (Accessed: February 2011).

[6] Kalman, R. E., “A New Approach to Linear Filtering and Prediction Problems,” Transaction of the ASME—Journal of Basic Engineering, pp. 35-45, March 1960.

[7] R. D. Hilton, D. A. Martin and W. D. Blair, “Tracking with time delayed data in multisensor system,” Technical Report, NSWCDD/TR-93/351, Dahlgren, VA, USA, August 1993

[8] M. Mallick, J. Krant, Y. Bar-Shalom, "Multi-sensor multi-target tracking using out-of-sequence measurements," Information Fusion, 2002. Proc. Fifth International Conference on , vol.1, no., pp. 135- 142 vol.1, 2002

[9] Y. Bar-Shalom,“Update with out-of-sequence measurements in tracking: exact solution,” IEEE Trans. Aerosp. Electron. Syst., vol.38, no.3, pp.769–777, Jul.2002.

[10] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn, “One-step solution for the general out-of-sequence measurements problem in tracking,” Proc. First IEEE Aerosp. Conf., BigSky, MT, Mar.2002.

[11] Saving millions of life, Decade of action for road safety 2011–2020, [Online]. Available at: http://www.who.int/violence_injury_prevention/publications/road_traffic/booklet_en.pdf (Accessed: May 2011).

[12] Vision 2020, VolvoCars, [Online]. Available at: http://www.volvocars.com/intl/top/about/corporate/volvo-sustainability/safety/pages/vision-2020.aspx (Accessed: May 2011).

[13] Verkehr in Zahlen 2003, Deutscher Verkehrs-Verlag

[14] Soeren Hess, “Cooperative Intelligent Transport Systems” [Online]. Available at: http://docbox.etsi.org//workshop/2009/200911_GREENAGENDA/07_HESS_C2CCC.pdf (Accessed: January 2011).

[15] Vehicle Platooning and Automated Highways, [Online]. Available at: http://www.path.berkeley.edu/path/Publications/Media/FactSheet/VPlatooning.pdf (Accessed: April 2011).

[16] J. Luo and J.-P. Hubaux, A Survey of Inter-Vehicle Communications, tech. report IC/2004/04, EPFL, Mar. 2004 [Online] Available at: http://infoscience.epfl.ch/record/52616/files/IC_TECH_REPORT_200424.pdf (Accessed: May 2011).

[17] Lidstrom, K.; Larsson, T.; Stranden, L.; , "Safety Considerations for Cooperating Vehicles using Wireless Communication," Industrial Informatics, 2007 5th IEEE International Conference on , vol.2, no., pp.995-1000, 23-27 June 2007.

[18] Lars Danielsson, "Tracking and radar sensor modelling for automotive safety systems", PhD thesis report, Chalmers University of Technology, 2010.

[19] Fredrik Bengtsson, "Models for Tracking in automotive safety systems ", PhD Lic. thesis report, Chalmers University of Technology, 2008.

[20] GPS, [Online]. Available at: http://www8.garmin.com/aboutGPS/ (Accessed: Oct. 2010).

[21] Lindsay Kleeman, "Understanding and Applying Kalman Filtering" [Online]. Available at: http://www.cs.cmu.edu/~biorobotics/papers/sbp_papers/integrated3/kleeman_kalman_basics.pdf (Accessed Nov. 2010).

[22] Doug Nychka, "A Kalman Filter tutorial" [Online]. Available at: http://www.cs.sandia.gov/CSRI/Workshops/2007/LSIP2007/talks/Nychka.pdf (Acessed: Sep. 2010).

[23] Masreliez, C.; Martin, R.; , "Robust bayesian estimation for the linear model and robustifying the Kalman filter," Automatic Control, IEEE Transactions on , vol.22, no.3, pp. 361- 371, Jun 1977

[24] S. Challa, R. Evans, and J. Legg, "A fixed lag smoothing framework for oosi problems", Commun. Inf. Syst., vol. 2, no. 4, pp.327 -350 2002 .

[25] S. Challa, R. Evans, and X. Wang, "A Bayesian solution to OOSM problem and its approximations", Inf. Fusion, vol. 4, no. 3, 2003.

[26] M. Mallick, S. Coraluppi, and C. Carthel, "Advances in asynchronous and decentralized estimation", Proc. IEEE Aerosp. Conf., 2001.

[27] Zhang Keshu, X. R. Li, Zhu Yunmin, "Optimal update with out-of-sequence measurements," Signal Processing, IEEE Transactions on , vol.53, no.6, pp. 1992- 2004, June 2005

[28] M. Mallick, S. Coraluppi, Y. Bar-Shalom, "Comparison of out-of-sequence measurement algorithms in multi-platform target tracking", Proc. 4th Annual Conference on Information Fusion, vol. II, Montreal, Quebec, Canada, August 2001, pp. ThB1-11–18

[29] SAVE-U, [Online]. Available at: http://www.save-u.org/download/PDF/D19_V0.6.pdf (Accessed: Sep. 2010).

Appendix A

Estimation theory

Example. ML, MAP, MMSE

- A common way of estimating non-random parameters is the maximum likelihood (ML) method that maximizes the likelihood function as:

Example.

$$\hat{x}^{ML}(Y) \quad = \quad arg \max_{x} p(Y|x) \tag{A.0.1}$$

$\hat{x}^{ML}(Y)$ is a function of random measurements Y.

- The corresponding technique for estimating a random parameter is the maximum a posterior (MAP) estimator defined as:

Example.

$$\hat{x}^{MAP}(Y) \quad = \quad arg \max_{x} p(x|Y) = arg \max_{x} [p(Y|x)p(x)] \tag{A.0.2}$$

the last inequality comes from the Bayes formula 3.3.1 ignoring normalization constant k since it doesn't affect the maximization process.

- An alternative technique of estimating randon parameters is minimum mean square estimation (MMSE):

Example.

$$\hat{x}^{MMSE}(Y) = arg \min_{\hat{x}} E[(\hat{x} - x)^2|Y] = E[x|Y] \tag{A.0.3}$$

We get the last equality by differentiating $E[(\hat{x}-x)^2|Y]$ wrt $\hat{x}$and equating it to zero. For evalauting $E[x|Y]$, we need posterior distribution since

$$E[x|Y] = \int {}^{+\infty}_{-\infty} x \ p(x|Y) \ dx \tag{A.0.4}$$

ML and MAP maximizes the likelihood and posterior function respectively. MMSE tries to minimize the expected value of squared estimation error conditioned on the measurements. Similarly another technique called Least squares (LS) minimizes sum of the squares of the error between the measurements and the observed function of the parameter [1].