Accurate Data Gathering in Wireless Systems for Robust Traffic Management

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI
de către

ing. Florica Maria Naghiu

Conducător ştiinţific:	prof.univ.dr.ing. Ionel JIAN
Referenţi ştiinţifici:	prof.univ.dr. mat. Viorel NEGRU
	prof.univ.dr.ing. Gavril TODEREAN
	conf.univ.dr.ing. Stefan HOLBAN

Ziua susţinerii tezei: 5 decembrie 2011

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor
12. Ingineria sistemelor
13. Inginerie energetică
14. Calculatoare şi tehnologia informaţiei

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

I would like to thank all people who have helped and inspired me during my doctoral study.

I especially want to thank my advisor, prof. dr. ing. Ionel Jian, for his guidance during my research and study at Politehnica University of Timisoara. He was always accessible and willing to help his students with their research. As a result, research life became smooth and rewarding for me. It was honor for me to be able to learn from him and have his support from over the past years.

I am especially grateful to conf dr. ing. Dan Pescaru. He has been a constant source of advice, guidance and support during all this time. Without his full support, I would probably have never reached the stage where I am today. I could say that most of my work achievements are due to him. During these last year's both Mr. Jian and Mr. Pescaru, have become my scientific parents, I am truly grateful for their trust and investment in me.

My thanks go to the members of SENT team, and in particular to prof.dr. ing. Alexandru Doboli, dr. ing. Codruta Istin, Gabriela Magureanu and Victor Gavrila. They have been my support developing the ideas.

Never the less, I would like to thank all my friends, but mostly to Ioana, Claudia and Ciprian for supporting me over this time, for encouraging me not to give up and to continue on this path. From my heart, I thank you all!

My deepest gratitude goes to my family for their unflagging love and support throughout my life. My parents have overwhelmed me with kindness and they helped me getting back and believing in me even when I lost my focus.

This work was partially supported by the strategic grant POSDRU 6/1.5/S/13, (2008) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People.

Timişoara, decembrie 2010 Florica Maria NAGHIU

To my parents

Motto:

"A small error in the former will produce an enormous error in the latter."

(Henri Poincare)

Naghiu, Florica Maria

Accurate Data Gathering in Wireless Systems for Robust Traffic Management

Teze de doctorat ale UPT, Seria 14, Nr. 5, Editura Politehnica, 2011, 86 pagini, 32 figuri, 6 tabele.

ISSN:2069-8216
ISSN-L: 2069-8216
ISBN:978-606-554-403-1

Cuvinte cheie: Filtre cu particule, corectie erori, sisteme wireless, MCMC.

Rezumat,

Teza de doctorat prezinta o abordare noua, concentrata pe metode utile in culegerea de date exacte de la senzori. Sunt considerate in mod special metode de filtrare a informaţiilor afectate de zgomote sau diverse erori, pentru a obţine date fiabile, utilizate în continuare în aplicaţii de supraveghere a traficului. Lucrarea demonstrează convingător utilitatea unei astfel de abordări, precum şi încadrarea ei în cercetările actuale din acest domeniu. Acest proces se bazeaza pe filtrele de particule, metode care sunt fiabile şi care in acelasi timp reprezinta tehnici puternice de calcul. Metodele existente, prezentate pe scurt în secţiunea „State of the art", nu iau în considerare posibilitatea de a utiliza metode de tip Filtre de Particule Monte Carlo Markov Chains (PF MCMC) bazate pe un model de observatii centrat pe acceleratie. În abordarea clasică filtrarea si predictia se fac cu ajutorul poziţiei observate. Considerand abordarea MCMC şi un model discret al comportamentului soferului, baza pe acceleratie, unui algoritm adaptiv este propus şi evaluat prin simulări. Acest algoritm se bazează pe o matrice de tranziţie cu cinci stari. În fiecare pas al acestui studiu, deciziile se bazează pe argumente teoretice. Algoritmul este validat prin rezultatele relevante obtinute in urma simularii. Pentru demonstrarea aplicabilitatii practice in partea finala a tezei, este prezentata o variatie a algoritmului propus. Aceasta aplicatie evalueaza acţiunile de depăşire, iar performanta este evaluata comparativ din punct de vedere a dependabilitatii, cu o alta abodare bazate pe logica fuzzy. Rezultatele prezentate in cadrul aceste teze, demonstrează faptul că luând în considerare un model de observatie bazat pe acceleraţie, in combinatie cu utilizarea unei metode adaptive poate creşte în mod semnificativ de performanţa sistmului de filtrare, in comparatie cu abordarile clasice.

Table of Contents

Acronyms

MCMC Markov chain Monte Carlo

PF Particle Filters

AMCMC Adaptive Markov chain Monte Carlo

PDM Probability Density Matrix

I**DM** Intelligent Driving Model

RTPF Real Time Particle Filter

1 Introduction

1.1 Abstract

This research is focus on methods useful in gathering accurate data from wireless sensors with direct application in robust traffic management. Traffic surveillance measurements based on wireless sensors can be affected by various noises especially on bad weather conditions. They consider especially filtering noisy information or error affected information in order to obtain reliable data, further used in traffic surveillance applications. This process involves particle filters methods, which are reliable and powerfully techniques. Contributions are brought in determining the suitable variations of particle filters in combination with a proposed model of observations based on drivers' behavior. Existing methods, briefly presented in the state of the art section, don't take into consideration the possibility of using MCMC methods based on an acceleration-driving observation model. In classical approach this is done with the aid of observed position. Considering MCMC and a discrete model of observation based on acceleration a new approach in the field of data filtering is presented in chapter three.

With the aid of simulations and detailed analysis of its performance, in chapter four an adaptive algorithm is proposed and further evaluated via simulations. The core of the algorithm is represented by the five states transition matrix. Considering an adaptive Markov Chain Monte Carlo algorithm (MCMC) based on drivers behavior it has studied the influence of several parameters in order to determine the optimum calibrations values for the surveillance system. It was determinate also the influence of external conditions, in order to find the best set of parameters calibration for applications that needs accurate data from sensors. In each step of this research, the decisions are based on theoretical arguments. A chapter is dedicated for validation through relevant simulations results. Moreover, a practical application for overtaking actions based on the core of proposed algorithm is also presented in the thesis. To prove its reliability the algorithm is evaluated via simulations. Mathematical tools are used in order to describe the model and also to give a proof for convergence of a modified MCMC error correction algorithm, in order to underline its' efficiency. The first step is to describe the mathematical model based on existing simulations. Second step is to define the important cases: no acceleration (constant move), constant acceleration/deceleration, pseudo-chaotic movement. In each of these cases, based on mathematical model, the necessary conditions for convergence in case of infinite time are determinate. Based on these conditions, a mathematical overview between interdependence of parameters it is given. Another step is to determine the moment of time at which the filters is reaching a given value, and remains stable, under that value. Given that fact that convergence and also settling time is dependent on initial conditions and also on sensor error, in the end of the thesis, also some computer simulations are

presented. The focus is putted on determining the number of steps necessary in order to obtain convergence of filter to a desired value or to a desired distribution. In theoretical part and state-of-the-art chapters, some details about convergence of optimal filter and also about integrating observation model into particle filter algorithms are presented.

Conclusions and future development possibilities are presented at the end of the thesis. The results of this work prove that considering a driving model based on acceleration and using an adaptive method can increase significantly the performance of classical methods. Several further improvements of the proposed algorithm, considering the specific context of distributed systems for traffic management, are underlined at the end of this thesis.

1.2 Motivation

The justification for choosing this research theme is complex. The subject of data filtering and accurate information gathering is known in literature for several years. Using sensors for traffic surveillance application faces some safety critical aspects as real-time constraints and a very good accuracy of provided data. Because accuracy is associated with the concepts of bias or systematic error in measurement, it is influenced by the procedure of taking measurements or by the instrument of measure itself. An important aspect is the determination of possible factors that influence the error propagation. Although several viable solutions were proposed so far, none of them combines all criteria for accurate traffic surveillance and driving assistance systems. Many of existing algorithms fulfill real-time requirements or high accuracy, but all this with the cost of computation effort.

Error can affect the process of data gathering in many ways and due to many reasons: sensibility of sensors, visibility area and wrong system calibration, bad conversion. Although not for all science fields and error close to zero is extremely relevant, for automotive industry this could become crucial. In case of driving assistance applications and devices for overtaking, errors correction is crucial. Wrong input in this type of systems can lead to catastrophic result.

Although the automotive industry is in a continuous and rapid development, road accidents are still a major problem without a viable solution. On average, each minute a person dies in a traffic accident. In addition, based on statistics supplied by the Organization for Economic Cooperation and Development Paris, hospital bills, damage to the estate, and other costs can reach up to 1.3 percent of gross domestic product of mankind. Given only the United States, the total annual amount rises to about 200 billion U.S. $. And, very important, the losses that matter most are not captured by these statistics, and cannot be monetized.

Another important factor was that the subject is an interdisciplinary one, combining majority computer science and information technology with transportation, mathematics and physics. Given the fact that these areas are dynamic, numerous opportunities are offered among with a wide horizon for improvement of specific known methods.

The need for mobility and speed specific XXI century led to the development of transport and the occurrence of certain specific problems. Among these, the most stringent are security of traffic participants, congestions and those related to environmental awareness.

In general, congestion occurs at certain times, regarded as the peak hours when a large number of participants use the same segment of road or are heading for the same destination. This is due to the fact that a specific segment of road infrastructure does not support a large volume of participants in traffic simultaneously. Other causes developing congestions are being described in [1]: charge entry tools or exit from certain segments of the road on which a fee, a low capacity road segment to manage a large volume of vehicles insufficient capacity for public transport. In those cases, I would add the following: adverse weather conditions, leading to lower velocity and thus a slower emptying of the segment, reparation work on certain road segments, leading to narrowing or blocking of certain sections, the appearance of a unforeseen event (such as, for example an accident).

Traffic issues are without borders, affecting most states. The results from research on causes and effects of road congestion, both in terms of road infrastructure, economic and environmental aspects, may be consulted in documents made public by the House of Commons of Great Britain [2]. Although figures are not the latest, document dating from 1998, can still illustrate the costs and the urgent need to solve the problem.

Primary solutions described in [1], underline the following actions: building more roads, more efficient tools for tax, gradually building access ramps to the highway, building intelligent systems for traffic management, construction of lanes with a high occupancy, reaction rate increased in case of accidents or traffic incidents, consideration of new areas and increase the density occupancy corresponding segments of the road, offering a regional transport authority growing departments. Since most solutions proposed in that article primary target area and the economic and political carrier, I will continue to focus on the solution that directly impact the field of computer engineering, namely the description of intelligent systems for managing traffic, as the proposed thesis has as targets improvements in this field.

The aim of the research project is to use an optimal architecture based on wireless sensor systems combined with appropriate algorithms to detect abnormal situations of the road (congestion, dangerous situations resulting from an overtaking maneuvers and lane changing).The immediate results are represented by streamlining the movement and ensure greater safety for participants. In order to determine an optimal algorithm for sensor network, the research was focus on conducting numerous experiments based on simulations, and on determining a suitable model describing observations.

1.3 Context

Traffic issues are very urgent, as demonstrated by numerous scientific articles published in the field of transport, and beyond, as well as by achievements of the industry. The automotive industry and the big machine-building concerns, have translated into practice scientific research, by many achievements that have led to the facilitation and streamlining traffic, increase safety of participants, and not the least, which had a favorable impact on the environment.

Further I will briefly present some of these achievements. First I will describe the existing solutions on the market and which have proven to be effective. By reserving certain frequencies and with contributions from traffic participants useful information is transmitted. This is one of the most commonly used methods to inform road users about possible incidents that occurred, or about the weather condition. From time to time it is possible to recall information or updates of earlier status in order to draw attention to an event or emerging conditions. Another possible form of informing traffic participants is by implementation of message-based technique. The disadvantage of these methods presented above is represented by the fact that only in case that one incident or a situation has already created, the information is passed on, but without any intention in anticipating the situation.

Aiming to streamline traffic, reduce time spent in traffic, but also with a declared intention to go green, one of the largest car manufacturers, Audi, implemented at Ingolstadt, a system called Travolution. The system is capable of communicate to drivers the speed with which they have to move so that at the next intersection to catch the green lights, and thereby to reduce time spent in traffic and also to avoid traffic jams. Unfortunately, this system isn't spread on an industrial scale. It is available only for a few pieces of A5 and A6 models of the manufacturer and it cannot be used in other locations outside the city of Ingolstadt. The role of this project was to reduce time spent in traffic, fuel consumption and to avoid frustrating operation of stopping at traffic lights. The immediate results could be seen in traffic flow and CO2 emission optimization. The system relies on communications modules integrated into each traffic light, which are able to send messages to cars in their vicinity, alerting them of the time remaining until the next green phase.

Another way to inform the participants in traffic, Suna, produced and marketed particularly in Australia. Suna GPS Traffic Updates provide real-time traffic information directly to the navigation system. Also, this system is compatible with all industry-leading GPS brands. The system was designed, in real time to determine road conditions and navigation system to inform any potential problems in perspective, and to recommend routes that would reduce congestion. Unfortunately, also this system has a limited scope, is currently available only in some city of Australia. Suna GPS Traffic is continuously updated, based on monitoring traffic and road conditions, and it is bringing detailed information on incidents, traffic congestion throughout the metropolitan area or areas that could affect the trip. The system is based on a combination of sensors and video cameras to continuously monitor the levels of congestion in an area of thousands of kilometers of arterial roads, agencies national, and highways. In addition, warnings about major incidents (such as accidents), major road work, severe weather conditions in terms of road and special events which may impact traffic are sent periodically. The system also is able to estimate the possible delays, and in this case to calculate and recommend an auxiliary route.

There are two basic types of traffic surveillance systems: on side of the road and placed on vehicles. Among the most popular road located are represented by magnetic loops, laser detectors and video cameras. A detector based on a magnetic loop is represented by a wire buried in the road surface and fed by a DC. If a vehicle passes over these devices, it induces an increase of current through the loop. These changes in intensity can be measured and taken into account to obtain information about traffic flow density. Laser detectors do not require installation in asphalt, and therefore they can successfully replace loop detectors based on magnetic and also classic video surveillance devices, especially at night or in areas with low visibility. The conventional video surveillance devices require good visibility. In dense fog, snow, rain, smoke or dust particles in the air at times of low natural light, these methods may be inadequate. However, it is precisely these low visibility conditions is a greater need for trust in traffic monitoring. Under these conditions most likely candidate is based on infrared and radar detector which has numerous advantages especially in darkness and fog. However, the above devices do not fall into the category of low-cost device. The categories of devices that are not located in road infrastructure have to do with the vehicle mount surveillance. These systems involve probe vehicles equipped with tracking devices, such as transponders, which will allow vehicles to be tracked by central computer facilities. Due to various factors such as cost, environment, infrastructure design and site monitoring devices regularly transmit the data which are accompanied by noise and may also be corrupted or unreliable.

Driving assistance systems are intelligent systems that provide driver support in tasks of driving a vehicle. Assistance in driving must be performed by robust systems, as they are incorporated and used in cars that are driven on public roads. By design, the roads have a high contrast predictable scheme and are governed by simple rules. Also, driving assistance systems must be driven and operated in all road conditions. These support systems can be adapted to solve well defined tasks that seek to help, and not to replace a driver. Whether it's co-pilot of a human or an automated system, it requires knowledge of: speed, acceleration, direction, position on the road, driving direction, location of vehicles and potential obstacles, a priori model vehicle dynamics, even the knowledge of the vehicle driver behavior. Assistant in leadership also must be able to deliberate on possible actions based on prior knowledge and possible consequences over time or through communication with the driver or even the secondary control over the vehicle. Taking into account the importance of having that support system, the role fulfilled and provided by real-time human-machine interface must be very carefully chosen. It must inform the driver in friendly manner and at the right time of the decision reached. On the other hand, it must not mislead or disturb the driver when giving an answer.

Nowadays driving assistance systems are not yet able to anticipate and prevent traffic anomalies in real time. Currently as pre-crash systems are operating safety measures approximately, at the best choice, 1-2s before a possible crash. Also this situation stays the same in case of forecasting a possible traffic jam.

Tracking and determining the correct position of a car has become an important issue for a robust traffic management. Therefore, many intelligent surveillance transport systems have been developed in recent years. An important aspect of this work is to obtain accurate results by using sensors and low-cost devices which are sending only a few frames per minute, and thus, by having information incomplete or affected by error, at hand. These systems usually require precise information on the current traffic situation. Some systems extract data and estimate traffic flow based on information from the sensors in a well-defined neighborhood. In order to filter noises and distortions from these measurements, and to use them for calculating in optimal conditions, a possible solution is to use appropriate stochastic filter type methods.

The traffic monitoring devices are incorporated in intelligent transport systems. The usefulness of these systems can be seen primarily related to economic and social problems of transport in most industrialized countries. Their crucial role is incident detection, traffic management, and collection of time travel. More specifically they can improve traffic management in congested networks. This requires a clear understanding of civil and traffic flow and also congestion avoiding methods on road segments. Other problems are to determine the time and location where a traffic jam or congestion happened and to follow how this can be propagated inside the transportation network. To this end, traffic status research and a set of parameters is required. In most cases of determining the state of traffic, such as density, requires some computation, as usually this information are not available directly from measurements at any point of the road network.

In this sense, filtering and predictive algorithms can become a powerful tool. If we consider the price perspective, sensors can have a low density, and base on this scenery information on large sectors of roads can be missing. In this case we need to estimate the information based on poor information received a priori. Particle Filters represent a set of flexible and powerful sequential Monte Carlo methods designed to solve the optimal filtering problem numerically, as they encapsulate the dynamics of the model and the observations [3]. Defining and estimating traffic parameters such as position of a car in a realistic mode, assumes state-space models which include elements of nonlinearity and non-Gaussianity. Also drawing samples of a moving car and filtering information will transform a continuous model into a discrete model, with possibility to introduce approximation errors.

1.4 Problem description - Scope

Traffic surveillance systems based on wireless sensor networks face some important requirements. A major aspect is represented by low-power need, as power consumption is a major design constraint in embedded systems. Indeed, high power consumption reduces the operation time in battery or solar-powered environments. Other important aspects regard heat dissipation, low computational resources or node reliability. Considering the real-time aspect when video sensors are involved, the network bandwidth and computational capabilities are critical. Nevertheless communication and distribution of information plays an important role, and in general these devices are part of a massively distributed system. Regarding this aspect, information extractions algorithms and data aggregation are highly required. Moreover, in critical applications as traffic surveillance and management, the data accuracy is a significant parameter.

Sensing data with nowadays devices is prone to error. In case of video sensors these errors result from sensor sensitivity and resolution, lens quality or shooter speed. There are also external factors that have major influences like various environment condition: fog, rain, snow, light or dust.

Safety critical applications, as traffic surveillance systems [4], require a higher degree of data accuracy than regular or statistical gathering of sensory data. Due to the fact that information can be alliterated in many ways or can be taken in unfavorable conditions, filtering is required. Two directions for this topic are possible as hardware and software error filtering. Hardware filtering can introduce some latency in data gathering and also in power consumption strategy, while software information filtering can be performed in a central node, with a higher computational capability. Also, using software filters and depending on the type of system implemented, one can implement a centralized or a distributed filtering of data, with corresponding benefits.

Particle filter algorithms can draw samples from practically any given distributions. Thus, this family of algorithms is one of the most popular methods of data filtering. Several classical particle filter algorithm variations were developed and successfully used in error correction applications. In this research thesis a brief overview of two main variations as Sequential Importance Sampling (SIS) and Sequential Importance Resampling (SIR) is provided. Contribution is brought through the lens of considering driver's behavior as model observation, a significant part of the algorithm.

The proposed algorithm and model assumes estimation of a continuous movement, transformation to a finite discrete state-space, and backwards to continuous, all in presence of noise affected observation. Based on these assumptions the filter is constrained to perform well in presence of three possible sources of errors: sensor error, discrete time sampling and acceleration adaptation. Another aspect which was considered as a pro for an adaptive implementation was the so called curse of dimensionality, as the rate of convergence of the approximation error decreases as the state dimension increases [3].

Contributions are brought in implementation of a multi-scale adaptive algorithm based on Monte Carlo Markov Chain, which incorporates a longitudinal model of observation, more exactly, intelligent driving model (IDM). Adaptability of algorithm consists in two orthogonal aspects: first of them is represented by the number of particles. These are determinate via calibrations, a minimum number of particles necessary for the convergence of filter it is set in initial phase of algorithm, and also the convenient threshold for filter error. Depending on these aspects more particles can be introduced for a better approximation in case of low performance of filter, and in case of reaching maximum number of particle, resampling is performed for particles with small weight. By these values it is easy to determine the rapport between computational effort and accuracy. The second aspect refers to adaptability of acceleration value. For this it was considered integrating in observation model a microscopic model, IDM. The acceleration value was adapted in order to better reflect reality. In this way continuous values were taken into discrete space, used for state space in order to reduce computational effort and after that were taken back to a continuous variation. In initial stages of the implementation of the algorithm, fix values for acceleration corresponding to each state were considered, but these lead to a non-optimum performance of the filter, as on border values between states, deviations from real values were introduced. By integrating IDM into observation model, each particle could 'follow' observation, by adapting the acceleration value. Considering the fact that each particle had the target to predict as good as possible the position of observation, and based on weight values a mediation it's done, it is obtained a good performance of the filter.

1.5 Structure

In the first chapter entitled "Introduction" a short overview on thesis is given. Shortly, in the abstract subchapter the thesis theme is reveal. Further are described the context of the thesis, a short motivation on choosing this particular subject and also a detailed overview on the purpose.

Second chapter presents the state of the art in the related field of data filtering and also of PF. Some considerations on theoretical aspects of PF are also presented in this second chapter as bases for further argumentation.

The third chapter presents the proposed approach which encapsulates both MCMC PF methods and also observation model based on driving behavior. A detailed description of the problem is presented in this core chapter.

Based on the results obtain via simulations on the proposed algorithm, a stringent need of improvement appeared. This was due to the fact that acceleration interpretation of the value from PDM, always returned a constant value, and did not suffer on any variation. An adaptive approach is introduced in the fourth chapter. This implementation has two orthogonal directions. One is concerning directly the resource and performance of the algorithms, and the other is related to introduction of variable values for acceleration. For a better understanding some theoretical considerations are presented in the beginning of the chapter.

The validation of the proposed algorithms is done via simulation. In this sense traffic simulator, has implemented in Java. Due to the fact that it was desired to determine the performance of proposed algorithm in observing and tracking one single and in any condition of driving (ideal case of continuous movement, or continuous variation of acceleration, and also in the least probable case of pseudo-chaotic movement) it was decide to implement this traffic simulator and not the adopt and adapt and existing one. In this case it could be easily adapt which kind of input can be given to the algorithm, from ideal input, to pseudo-random values and also real live measurements. Experiments and practical applications are being described in chapter five, respectively six.

The thesis ended by the "Conclusions" chapter. Here are presented some concluding remarks, contributions and future development is presented at the end of the thesis. At the end of the chapter are presented the list of publications which were used for disseminating the results so far. Also this thesis is supported by two scientific reports presented in department of Computer Science in the last two years, and is accordingly to the proposal made at the end of first research year.

Conclusions

In this chapter a short description over the thesis content, motivation, and also objective is described. The research is focus on data filtering and accurate information gathering. The usability of this research it is proved by the constant concern of automotive industry in developing safe systems, user and environment oriented. Error can affect the process of data gathering in many ways. It can affect frequently information coming from wireless sensors, due to many reasons: sensibility of sensors, visibility area and wrong system calibration, or bad conversion.

Figure 1.1 depicts various error sources when gathering information from a real-time wireless sensor distributed system.

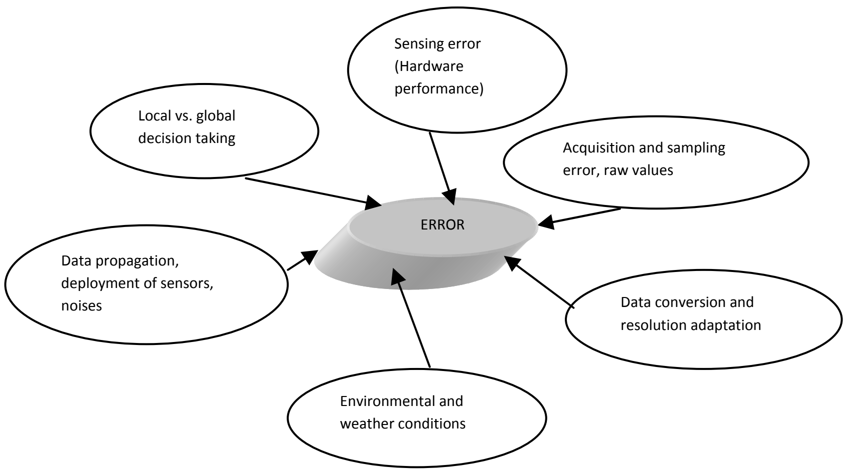

Figure 1.1 Possible error sources in wireless sensors networks

2 State of the art. Theoretical considerations

Algorithms for traffic surveillance represent an important part of all Intelligent Transportation Systems. Their main purpose is to assure a high degree in accuracy of data in order to increase the safety on roads. Classical surveillance equipments such as magnetic devices or video cameras proved to be non-optimal solutions as they bring lots of constraints such as implant in road infrastructure, weather and atmospheric conditions, orientation and nevertheless, costs. In order to assure a better reaction and real-time responses most surveillance systems are specialized on a particular capability (e.g. flow speed monitoring). Wireless sensor networks represent a strong alternative to classical traffic surveillance. This is given by their flexibility, easy deployment, support for remote control and overall, low costs.

In addition to video sensor networks, magnetic wireless sensors represent a particular promising approach. Depending on the strategy used for data gathering, the accuracy of such equipment is between 80% and 98% [5]. Also acoustic sensors where proposed in order to develop a low cost solution. Here the constraints derive from the background noise and also from power of processor and energy consumption [6].

A serious impediment in traffic flow surveillance is represented by acquisition errors due a variety of reasons. This includes sensors errors, hard weather conditions, nodes hardware failures or network transmission errors. Over the last years prediction algorithms turned to be one of the most powerful tools used to overcome these problems. Several classes of algorithms were studied, such as neuronal networks, fuzzy logic, or a large family of Bayesian based prediction methods. The work in this thesis is focused on Monte Carlo Markov Chain Particle Filter. Particle Filters (PF) are used for estimating the state of a dynamical system from sensor measurements as they generally consist on a predict/update cycle. PF are trying to determine the belief about the current state, based on the probability of all observed data until current moment. As inputs they have the observations, the perceptual model, which is the probability that a particular given observation is in a defined state at time t, and the action model, which is the probability that system will end up in state x_t at time t, assuming that it started in state x_{t-1} at time t-1, and also received as input observations from t-1 [7].

Using sensors for traffic surveillance application faces some safety critical aspects as real-time constraints and a very good accuracy of provided data. Because accuracy is associated with the concepts of bias or systematic error in measurement, it is influenced by the procedure of taking measurements or by the instrument of measure itself. An important aspect is the determination of possible factors that influence the error propagation. These errors can differ widely in terms of severity, frequency of occurrence and statistical properties [8]. As presented, data coming from wireless sensor could be affected by errors in many ways. First, we investigate the hardware performance limitations. Usually the precision of a device is provided on its technical specifications and usually it doesn't cause errors higher by 2-3%. Referring to the real-time aspect of the application some other issues are becoming critical. The first of them is the number of samplings taken on a time unit. The second is the way to mix these samples. In many real time applications involving particle filters for error correction, sensor information arrives at a significantly higher rate than the update rate of the filter. A common approach to dealing with such situations is to update the particle filter as often as possible and to discard sensor information that cannot be processed in time.

In [9] the authors introduce a method in which posterior probability density function is represented as mixtures of sample sets, where each mixture component integrates one observation arriving during a filter update. The weights of the mixture components are set in order to minimize the approximation error introduced by this representation. This is done with the aid of a real time particle filter, a method where all sensor measurements are considered by distributing the samples among the observations within an update window. Another aspect that can interfere in data accuracy is the process of data collection and transmission over the network. Subject to error are conversions of raw data and resolution changes as well as distributing information using multiple hops, as noise may interfere. In distributed systems, error correction can be done in two ways, by having a central node or without any central collection point. Scalable and robust distributed particle filter for dynamic environments proves to be a reliable solution only in the second case. For these cases, selective communication schemes enable individual platforms to communicate the most informative piece of information to the others [10].

In [11] the author describes two methodologies for performing distributed particle filtering in a sensor network. The first algorithm relies on likelihood factors and the second algorithm adds a predictive scalar quantized training step into a more standard particle-filtering framework allowing adaptive encoding of measurements. The initial assumption is that a Markovian state-space model can capture the monitored environment. It involves potentially nonlinear dynamics, observations, and observation noises. This will help in factorizing the likelihood, and forming parametric approximations to products of likelihood factors by using the particles and their associated likelihoods as training data. The model parameters are then exchanged between sensor nodes, instead of the data or exact particle information. This assumption determines also a limitation of this method. The second distributed algorithm is more computational excessive and it uses an adaptive data-encoding approach. It involves training of predictive linear quantizes at every time-step based on a common particle filter maintained on all nodes. Then, sensor nodes transmit the quantized data to one another. By these considerations there are no restrictions on the nature of the likelihood function. In case of distributed systems, a challenge is represented by overhead in communication.

In reference [11] is proposed a distributed particle filter implementation in which parallel particle filters run at multiple nodes. These shared filters are used to quantize vectors of measurements. The information obtained from particle filter is then encoded using Huffman and placed on in tree structure. Then it is introduced a vector scheme that reduces the fraction of communication energy wasted through transmitting packet headers. At starting point the particles are blindly propagated. The success of the scheme is guaranteed by the fact that the sets of particles are identical, condition achieved by initializing the filters with the same seed and ensuring that they all propagate based on the same distributed quantized measurements. The cost of the transmission directly depends on the good representation of the state by propagated particles. It is proved that in case of a good representation the measurement should lay in a densely populated bin and the codeword will consist of very few bits. The algorithm is highly computational and the price is reduced only in a few cases. This approach is not suitable if data measurements are not reliable. However, due to the fact that sensors convert physical quantities from the real world into a machine-readable digital representation, possible errors from conversion algorithms or resolution adaptation may appear.

Traditional methods for reliability usually introduce over-heads at different levels and also in terms of real-time processing. In [8] a new approach is presented. It relies on creating predictive models based on the temporal correlation in the data, handling multiple sources and using them for real-time error correction.

Considering these last presented aspects we can refer to some important issues of design factors in wireless sensor applications. First of them is fault tolerance. Indeed, in case of some nodes failures due to lack of power or physical damage or in case of communication problems due to environmental inferences, it should not affect the overall task of sensor networks. This aspect may depend also on the scalability of the network.

To deal with the error aspect two important modes of error control were proposed. These are forward error correction (FEC) and automatic repeat request (ARR). The first solution has problems in terms of decoding complexity, and thus simple error control code might present a better solution. The last one is limited by the power consumption scheme. Usually, in solving link reliability of massively wireless distributed systems, the FEC is considered.

Some solutions were also provided in the field of on-line fault detection for sensor networks. In [12] a generic approach with a flexible trade-off between accuracy and latency for identifying the sensors that have the highest probability to be faulty is presented. It is based on taxonomy for classification of faults in sensor networks on-line model-based testing technique.

In terms of deployment and operation of a wireless system network responsible for traffic management, the challenges are even harder. Small video camera may not provide relevant information on bad weather conditions or in case of dust or smoke. In case of random deployment the system can face lack of sensors in possible key points. Possible alterations of information can also appear due to noise or low resolution in sensing. When information suffers distortions and error steps are over an acceptable range, error correction algorithms are required. There are two fundamental challenges in the event detection problem for a sensor network. First, the detection accuracy is limited by the amount of noise associated with the measurement and the reliability of sensor nodes. Second, collecting up to date data to a central point is hard to be achieved. Therefore, to get optimal detection accuracy, it is essential to consider both factors during detection. It is also necessary to limit the effects of inaccurate measurement or faulty behavior of individual components to a minimum. However, the basic idea of distributed detection is to have each of the independent sensors involve in taking a local decision and then combine these decisions at a fusion sensor to generate a global decision.

If we consider correct information coming from sensing data, a solution is presented in [13]. It provides a designing method for an efficient real-time application of dynamic probabilistic models to streaming data. To achieve that, authors use an abstraction of a user declaratively model. The output of the models is presented to the user as a probabilistic database view. This is done by the aid of a particle filter, by the model-based view through queries over particle tables.

2.1 Particle filter through mathematical glance

Traffic surveillance is a complex task because of non-linearity of the flow of vehicles and the many interactions between them and also the computational complexity and the need for real-time response. Many video surveillance methods based on predictive techniques used to estimate an unknown state dynamic and usually come from a collection of observations and sequential non-linear, affected by noise. Stochastic approaches often reduced to an estimation problem, an estimate of the state for a period of a series of state-space model. Most investigative techniques for non-linear / non-Gaussian models are based on Monte Carlo method, and are known as particulate filters.

The term particle probability density designate applicable to any type of space and state and thus represents a generalization of the traditional Kalman filter. An important step in dynamic Bayesian approaches is estimated to construct the probability density function PDF posterior [14]. To analyze and make inference of a dynamic system at least two models are necessary. First, a model that describes the evolution of states over time (system model) and secondly, a model with the accompanying noise measurements (measurement model).The filter has essentially two stages. These are the prediction and update. Phase prediction and system model used earlier to determine the new PDF value and transmit states at a time to another. Because the condition is usually influenced and disturbed by random noise, the impact is reflected in the general deformation and distortion of probability density function from the previous form. Update operation uses the latest measurement to modify the prediction function (PDF) [14].

Indeed, a particle filter is a technique for implementing recursive Bayesian filters by Monte Carlo method of sampling. As described by Mihaylova in [15], the flow of traffic on the highway is represented by a multi-particle model with non-linear character. This includes also complex interactions between vehicles, such as traffic jams, waves of start-stop and more. More filter configurations can be used to process information from sensors in an operation to estimate the traffic flow. Most common approaches and methods are based on stochastic filtering. In [16]is

presented a comparison between Kalman filter and unscented Kalman filter for state estimation and different parameters for different detector configurations.

Own position estimation of a moving entity is another application of the filter particles. This is a filtering problem rather than a static estimation problem, when an inertial navigation system is used to provide measurements on a moving body. Another problem is shipping, where, in addition to such position, speed, altitude and direction, and angular acceleration are included in the problem subject to filtration. Tracking a target, where the position of another object is estimated based on measurements of relative position and status, is presented in [17].

For a small number of observations, the problem of tracking a target, improved particle filter was proposed using a modified algorithm LS-N-IPS (Local Particle Sampling System interacting N-N-IPS) [18]. This is done using a non-trivial operator local search, with the aim to improve the prediction. At each step, the predictions were refined in a local search procedure using the latest observed data. This would be best if they operate a small number of particles.

Another approach, which has some similarities with the algorithm presented in this paper, is proposed in [19]. Here, the authors used a hybrid Monte Carlo filter for analysis using the posterior distribution of an application for tracking people. Rather than assigning a weight to each particle based on the similarity of risk function, each particle produced a Markov chain to sample posterior distribution using the gradient estimates.

Extended Kalman filter (EKF) [20] is another common method of tracking video-based applications. Disadvantages of this method refer to the algorithm complexity, which increases with the number of measurements. Also, they are considered very sensitive to noise parameters. Therefore, this method expects a reasonable estimate of initial state variables. In contrast, the filter particles can start from a uniform distribution, but poor performance, on the size of the state vector.

If the initial specified conditions are implemented correctly by the previous distribution, fast convergence can be guaranteed when using a particle filtering approach. A hybrid solution was proposed in [21] to improve convergence of particle filters. This approach generates a first phase particles in the same way as the conventional case. Then try to move closer to the previous value of particle distribution generated by a step EKF. This strategy was successfully applied in training neural networks, but the disadvantage is that it must comply with the Kalman filter on the noise distribution. In a recently proposed algorithm based search method on the average shift in a particle filter and a target representation, which uses multiple semi-overlapping color histogram was proposed in [22].

A combination of Monte Carlo filter and Markov chains is presented in the literature as MCMC (Markov Chain Monte Carlo) [23]. Originally a first state x_0 is taken in accordance with a proposed density, which can be Gaussian, where the covariance is also calculated Monte Carlo samples to the last step. Subsequent state vectors are then taken by the step function status. MCMC methods allow estimation of probability distributions using large samples of some standard considered. By using this method may be used to calculate the average of these samples more quantities of interest. These samples can also be used to calculate statistical estimates, such as regions with high probability, or the differences are highlighted. This method is a powerful algorithm with importance sampling and benefits in reducing problems associated with sequential Monte Carlo filters. For some models, however, MCMC method may not be optimal, because a large number of iterations would be needed to achieve the desired density distribution [24].

In case of using mathematical prediction and correction tools, these are based on Bayesian algorithms structure. The advantage is that they use previous states of the system (from previous observations), and phase prediction is corrected by observations of the current step. Studies in the field, have shown that Kalman filters are not the most appropriate tools in the industry because not only suitable for models with linear distribution.

Improving this is achieved by particle filters, algorithms able to respond to system performance with almost any type of distribution. Also another strong point of these algorithms is that in addition may have predictive ability and capacity correction comments, comments which often contain useful information in addition to noise. In the field of particle filters applied in traffic management have been important research conducted by Mila MIHAYLOVA (et all), especially to improve the filter (Unscented filters, operations with intervals), attempts to implement distributed filters to reduce overhead communication (M. Coates) implementation of probabilistic tools for traffic management sites (is theme proposed for research by T. Singliar), the implementation of particle filters and other methods Bayesian able to operate in real time.

In many applications based on particle filters that require real-time execution information from the sensors reach a significantly higher rate than the discount rate of the filter. Most common approach in these situations is to update the particle filter as often as possible and giving up that information coming from the sensor, which cannot be processed in time. Another possibility is to mix the sample sets for a single value PF filter sent back when the upgrade information. [26] Components weights mixes are set so as to minimize the error introduced by this representation. This is done with RTPF (Real Time Particle Filter) [26], a method where all measurements are taken into account from the distribution of samples among sensor observations in a window update.

Models for predicting the behavior of a complex object is based on knowledge accumulated over time and used to store information about relevant changes or can change their parameters in the future. This information is usually represented as a set of data and must be collected in a long term observation in a complex dynamic system.

Particle filter algorithms rely on recursive Bayesian approach and a probabilistic state-space formulation, which attempts to construct the posterior probability density function of the state based on all state and observation information available. Their advantage consists in ability to approximate non-linear and non-Gaussian process, and approximate a continuous density as a discrete one, by representing the required probability density function by a set of weighted particles and using them to estimate the state. [15]

Another important aspect for Particle Filter algorithms is represented by the state estimation especially in high-dimensional systems. In order to perform well in these conditions and to converge to a desire distribution, it requires that the particle number to increase exponentially. As mentioned in [28] quality and efficiency of particles can be improved by two methodologies: "bottom-up" and "top-down". Re-weighting, re-sampling, the kernel based particle filter, the hybrid particle filter and mean-shift tracker, and the annealed particle filter are stated by [28] to be part of the bottom-up methods control the particle quality via direct particle modifications. On the other hand, the top-down methods are described to be focus on the system model which describes the prior distribution and the observation model, the likelihood of Particle Filter algorithms, in order to better describe the problem.

It is well known, that in order to perform well for a given application Particle filters and all Monte Carlo methods need to encapsulate a model of observation. This methods relay on probabilistic Bayesian filters. The principle of those filters is determinate by the Bayes' theorem, which states how to update a prior belief about a variable x given a new observation z and an observation model [29]:

$$\underbrace{p(x|z)}_{posterior} \propto \underbrace{p(x)}_{prior} \underbrace{p(z|x)}_{obs.model}$$

In case of considering integrating model observation for a robot movement video tracking, we can find in [29] the proposed mathematical model of filter is given by the following equation (Bayesian recursive filtering):

$$\underbrace{p(x_t\,|z_{1:t}\,,u_{1:t}\,,m)}_{current\ pose\ estimation} \propto \underbrace{p(z_t|x_t,m)}_{observation\ model} \int \underbrace{p(x_t\,|x_{t-1}\,,u_t)}_{motion\ model} \underbrace{p(x_{t-1}\,|z_{1:t-1}\,,u_{1:t-1}\,,m)dx_{t-1}}_{previous\ pose\ estimation}$$

where z represents observation, m the map set of 3dimensional landmarks, x_t the pose of the robot, and u_t the actions of the robot. Also in this case and for SLAM, filter consists of three major steps: sampling, importance weight, and resampling. First step uses the previous generation $x_{1:t-1}$ and samples the next generation of particles $x_{1:t}$ from the proposal distribution $\pi(\cdot)$. The selection of proposal distribution can greatly influence the performance of algorithm itself.

Particles evolve randomly in time according to the dynamics of the model and the observations [3]. Due to interactions of particles classical limit theorems relying on statistically independent samples do not apply. [3] Presents a survey of convergence results on Particle Filters.

Particle filter (PF) represents a sequential importance sampling method based on Monte Carlo simulation and Bayesian sampling estimation theories, which evolved from the Bootstrap nonlinear filtering algorithm.

PF can estimate non-linear and non-Gaussian dynamic processes and has the property that it could be directly applied on any non-linear system model. The particle filter aims to estimate a sequence of parameters based only on the observed data. Several approximation techniques have been presented in a non-linear, non-Gaussian state space context [30], [31].

Because of their interaction, particles are statistically dependent. Consequently, the classical convergence results on Monte Carlo methods based onindependent and identically distributed assumptions are not applicable [26]. Given this, a target for optimization in the implementation of particle filter is the fact that total error variation vector of observations is consistent with the result that the variance estimator and particle filter is independent of the size of states [26]. If the MCMC method is applied, the idea is to use observations and to generate samples from posterior distribution or likelihood function of interest and use them to extract relevant information, because all known information is kept by their posterior densities or similarity functions. Thus, samples with higher values of posterior densities can be used for future calculations and deduction, because they best approximate the unknown situation.

Using Bayesian sampling estimation theory, the posterior density $p(x_k|y_{1-k})$ can be inferred from the prior density $p(x_k|y_{1:k-1})$, [32]:

$$p(x_k \mid y_{1:k}) = \frac{p(y_k \mid x_k)p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})} \quad , \qquad (1)$$

where

$$p(y_k \mid y_{1:k-1}) = \int p(y_k \mid x_k)p(x_k \mid y_{1:k-1})dx_k \quad , (2)$$

and

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1})p(x_{k-1} \mid y_{1:k-1})dx_{k-1} \quad .(3)$$

PF uses Monte Carlo simulation method to approximate the posterior density by N particles with the associated weight:

$$p(x_k \mid y_{1:k-1}) \approx \sum_{i=1}^{N} w_{k-1}^{i} \cdot f(x_i) \quad . \qquad (4)$$

In the basic bootstrap particle filter [33], a more general approach involves the concept of importance sampling [34]. In order to briefly describe the basic PF algorithms, we assume that a set of particles and their weights are given by a discrete posterior density representation. The bootstrap PF then implements the following three iteration steps [32]:

a) *sampling*: draw N samples from the existing set of particles according to their likelihood weights;

b) *prediction*: propagate the particles through the transition;

c) *update*: each particle and normalize the weights in the end.

Basically, particle filters rely on importance sampling and thus require a good design of proposed distribution in order to approximate the posterior distribution. One possible strategy to fulfill this need is to sample from the probabilistic model of the transitions prior states evolution [35]. In case that the measurement is in prior sample tail or the likelihood function is too peaked in comparison with the prior, filter performance can be very weak. In this reference a dynamic state space model is presented as an alternative for improvement for above mentioned issues. It is based on the fact that states follow a Markov process and observations are assumed to be independent to states. The mix with a sequential importance sample is a combination that outperforms standard particle filtering. Also a good technical and formal PF variations presentation can be consulted in [35].

An adaptive PF is described in paper [36]. Filter efficiency and accuracy improves as the number of particles used in the estimation increases. Better results are obtained by considering the propagation function that reallocates these particles on all iterations. Authors underline that the effect of particles number on filtering accuracy is determined by two intuitive factors: the true density complexity and how closely the proposal density mimics the true density. It has been demonstrated that the improvement is due to the processes varying dynamics and to the fact that models are ignored. The proposed self-adaptive version of the particle filter uses statistical methods to adapt the number of particles and the propagation function on all iterations with similar computational effort.

Several resampling schemes have been proposed in the literature including multinomial, residual and stratified resampling [37], and [38]. Based on resampling strategy, in [38] a multi modal sequential particle filter algorithm used for object tracking is presented. A hidden state sequence linked to several sensory observation sequences in association with a based framework, is considered such as each sensor provides likelihood (weight) associated to each particle and simple rules are applied to merge the different weights, as addition or product. The aim of the proposed multi modal sequential importance resampling algorithm (M2SIR) [38] is to generate a new particle with a three step approach: sampling via Importance Sampling strategy to a set of M candidate samples and their associated weight vector, determination of a likelihood ratio vector and candidate sample selection given by an importance sampling strategy operated on a normalized likelihood ratio vector. Reference presents the algorithm as a mix between condensation algorithms based on likelihood ratios to merge the observations within the sampling step.

A limitation of particle filters is that particle degeneracy appears, in which most particles yield no useful information, thereby negating the advantages of this approach. Auxiliary particle filters address this limitation by constructing proposal densities that better correspond to the true posterior distribution [31]. An extended auxiliary particle filter was developed in [39] where an additional hyper-parameter was added to better adapt the proposal to the posterior.

A comparison between fixed-lag roughening and the block proposal distribution sample strategies is offered in reference [39]. Both exploit “future” information, when it becomes available. Consequently, filter’s estimation for previous time steps is improved. Fixed-lag roughening perturbs trajectory samples over a fixed lag time according to a Markov Chain Monte Carlo kernel. The block proposal distribution directly samples poses over a fixed lag from their fully joint distribution conditioned on all the available data.

State space dimension is taken into consideration in [4]. There are cases of tracking applications where large dimensional problems such as multimodal observation likelihood are often and the state transition prior is often broad in at least some dimensions and direct application of PF requires an impractically large number of particles. A particle filter with efficient importance sampling and mode tracking provides a possible solution to these problems. This is based on estimating a hidden sequence of states, from a sequence of observations, which satisfy the Hidden Markov Model. Also an observation model and a system model are taken into consideration in order to develop the algorithm.

The authors of [30] discuss the dynamic state space estimation problem of an exothermic irreversible parallel reaction in a continuous stirred tank reactor. It presents a solution based on an alternative approach whereby particle filters based on the sequential Monte Carlo method are used for the estimation task. In fact, a Markov chain Monte Carlo method is proposed to enhance particle filters where the estimates of the initial conditions are poor. The above-mentioned mixture is concretized in the Auxiliary Sequential Importance Resampling (ASIR). Reference [30] also invokes that, as the computational cost of MCMC increases with time, MCMC is performed during the first few time steps, before switching over to the conventional particle filter. In their experiments, authors have shown that a MCMC running for only ten steps can significantly improve state estimation performance whilst incurring reasonable computational load.

The vehicle dynamic was taken into consideration in implementing a PF algorithm in the paper [31]. The proposed method is represented by a Rao-Blackwellised particle filter, used to determine the faults in the suspension elements of a railway vehicle via changes in the vehicle dynamic model parameter values. It was considered that parameter estimation of vehicle condition and monitoring system based on PF method is able to detect and isolate incipient faults is usually described by a linear stochastic state space model. The use of Rao-Blackwellisation techniques can increase the efficiency of sampling in PF by reducing the state space size to be sampled through marginalization, which results in RBPF.

2.2 Classification of mathematical models

Theories related to traffic flow modeling try to describe in a precise manner, based on mathematical descriptions, the interactions between vehicles, drivers and infrastructure. Infrastructure is represented by the road system and all its operational elements, including control devices, signs and markings. These theories are an indispensable element of all traffic models and analysis tools that are used in the design and operation of streets and highways, and the analysis of the behavior of participants in various traffic situations.

Scientific study of traffic flow has its beginnings in the 1930s, starting from the application of probability theory to describe traffic and, with pioneering studies by

Bruce D. Greenshields at Yale. Those studies were performed on various models of vehicle volumes, speed and investigation of traffic performance at intersections. After the Second World War, there was an increase in the number of cars in use and expansion of the highway system. It was also not reflected by an increase in the study of traffic characteristics and the development of theories on the flow traffic.

A traffic simulation model represents the dynamic change over time, the traffic status. Macroscopic level of traffic modeling can be likened to a pipeline that crosses water. A mesoscopic model aggregates behavior of individual vehicles, and if the microscopic level has individual behavior details as central points of the model.

Another possible classification of different traffic models is given the stochastic, deterministic, and event-oriented theories. Stochastic models capture the variation in reaction time, reaching the destination and routing choice. After each simulation, results differ because of the influence of different factors leading to the need to save and replicate results. For these models, the next states cannot be determined with great precision. Deterministic models are based on stable physical laws, and next state can be determined with great precision.

Mathematical models of vehicular traffic gain field lately due to their applicability in solving problem in terms of robust traffic management situations such as congestions and accidents avoidance systems. Classification of mathematical models of vehicular traffic can be done considering several parameters and also the scale of representation. In literature we have the following classifications: microscopic models, mesoscopic model and macroscopic model.

Microscopic traffic simulation models differ significantly from the conventional "aggregated" traffic models. This is due to the fact that instead of shaping traffic flows, the model simulates the microscopic behavior of all individual vehicles in the network. These models allow consideration of some important phenomena such as drivers on high traffic roads, behavior near their maximum capacity, and also the complex interaction between vehicles at intersections, pedestrians and traffic and interaction between different categories of vehicles. In the microscopic models are snapped some behavioral patterns, such as: "Car following attitude" describes acceleration, deceleration and keeping a safe distance between cars, models of this type are stimulus - response, safety distance; "Changing lanes; "Stop-and-go". Microscopic models focuses on each particular vehicle considered and tries to describe via differential equations interactions between individuals and infrastructure. In case of microscopic model the accent is putted on the time-space behavior base of individual drivers and their influence in the proximity. One of the main disadvantages of this kind of representation is represented by the facts that if the number of observed cars is growing also the complexity and the equations evolve rapidly.

The most popular models at the microscopic level are represented by the Cellular Automata and Cognitive multi-agent systems. Cellular Automata or a robot cell is a collection of cells "colorful" belonging to a network that has a specified shape and evolve through a series of discrete time steps, in accordance with a set of rules based on the state of neighboring cells. The rules are then applied iteratively whenever desired. Cellular Automata can be presented in a variety of forms and versions. One of the most fundamental properties of such robot is the type of network is calculated. The simplest type of "network" is the one-dimensional, based on just one line. Variations may include: two sizes, shapes, square, triangular, hexagonal or the networks. Cellular Automata can also be built on grid networks with random numbers dimensions, but integers are the most common choice.

The main characteristics of microscopic simulations are the following, as described in [51]: detailed representation of road network geometry; representation based on individual characteristics, includes stochastic components.

Mesoscopic models use both macroscopic level by aggregating the individual components and interactions as the microscopic level. These models describe the traffic participants with a high level of detail, but also the behavior and interaction are described by a lower level of detail.

Macroscopic and kinetic models describe the traffic at a high level of aggregation as macroscopic density, average speed, or kinetic distribution function as continuous functions of space and velocity, without considering its individual constituent parts. In this case accent it is putted on time-space behavior of the whole collectivity.

Macroscopic level of traffic simulation is based on a model that discusses the relationship between the main flows of traffic parameters [52]: speed, flow and density. Variables that reflect macroscopic traffic model can be calculated for each location, at any moment in time for each measurement interval. In practice, most often are used detectors that measure traffic flow and speed over a certain period of time. If you want to calculate the average speed for a time, must be individual harmonic gears. When times extend beyond five minutes, some dynamic features are lost.

Currently for determining a mathematical model on the macroscopic level, are analogous to the known physical laws phenomena and so the kinetic theory and fluid dynamics, and gas kinetics, obtaining differential equations describing the relationship of traffic [53]. Level simulation and modeling macroscopic traffic METANET model [52] is the reference for many researchers. It is based on certain similarities with the gas kinetic law (based on equations that correlate with running speed traffic density).

Macroscopic models have an advantage in that data needed for such models such as traffic density and speed are at the same level of aggregation that the data provided by measurements from the aggregation devices [52]. While macroscopic models have the ability to efficiently simulate large networks generally lack the individual details. Because of this modeling response to various incidents is difficult.

Considering these aspects this representation is computationally more efficient as it involves fewer partial differential equations and the global characteristics of the system, which are readily accessible, but all of the in the detriment of accuracy [54]. In case of macroscopic model the number of considered car should be large enough in order to represent approximations of temporal and spatial dynamics of the traffic system. Due to their likelihood with the flow of the fluids, modeling of vehicular traffic the flow of cars along a road, macroscopic models are often called hydrodynamic models [55].

Mesoscopic models are based on an intermediate level of detail, describing the individual vehicles, but not their interactions, or in other words, are based on stochastic methods, mainly using master equations describing the time propagation of a traffic state probability function either of single cars or car clusters [56]. For single car states case, Boltzmann like master equations are considered, and for this reason this type of modeling it is called kinetic model [56]. In this case representation of quantity of interest it is given by: f (x, v, t), where f (x, v, t)d*x*d*v* is the probability to find a car at place between x and x + d*x* with velocity between v and v + d*v* at time t. Derivation of Boltzmann equation can be done in one of the following ways: by heuristic plausibility consideration, by defining a stochastic Markov process or by using the BBGKY-hierarchy [57].

Heuristic model derivation for f (x, v, t) is given by equation:

$$\frac{\partial f}{\partial t} + v\,\frac{\partial f}{\partial x} = \left(\frac{\partial f}{\partial t}\right)_{\text{acc}} + \left(\frac{\partial f}{\partial t}\right)_{\text{interact}} \quad [57]$$

In fact, mesoscopic models present the intermediate step between microscopic and macroscopic models, and represent mutual interaction of considered individuals. Kinetic modeling was first used by Prigogine and relies on the principles of statistical mechanics introduced by Boltzmann to describe the unsteady evolution of a gas. Mesoscopic models are ideal for prediction applications, where the detailed modeling of route choice and other strategic driver choices are essential, but where the detailed modeling of driver interaction with the road network and other drivers is not needed. Based on the mathematical model presented above, traffic simulators are being developed. A survey on several mesoscopic traffic simulator, as well as pro and cons for choosing a particular type of model between microscopic, macroscopic and mesoscopic it is presented in [53]. Along it is also presented a mesoscopic prediction model, called Predikt [53]. [54] Depicts some of the major macroscopic and kinetic mathematical models of vehicular traffic and also some application to road networks.

A state of the art survey on vehicular traffic model is presented in [58]. Same classification as above, based on the level-of-detail for vehicular flow is used, but additional, for each of the categories, issues like modeling accuracy, applicability, possibility of generalization, and model calibration and validation, are discussed. An interesting classification about usage of mathematical models and describing traffic via equation it is also done. Thus, models are being divided between purely deductive, purely inductive, and intermediate approaches. First class, purely deductive models are based on accuracy due to the physical laws applied. In the second case modeled data from real systems are used to fit and finally in intermediate approaches, first basic mathematical model-structures are developed, after which a specific structure is fitted using real data.

2.3 Driving model

Modeling driver behavior is a complex and heavily researched in recent years. Driving activities is involving numerous cognitive sub-tasks, such as a lane change, adapting speed to road condition and legal regulations, avoid obstacles, choosing the right path and so on. For these many methods have been developed over the years. The methods are explored by cognitive architectures [43,44], and interfaces with the management support systems, such as adaptive cruise control speed [45] In this last reference empirical observations about different scenarios in the activity of driving are also mixed with scientific research. At the microscopic level, driver behavior is modeled in [46], with a strong emphasis on modeling the acceleration in different conditions.

Due to urgent need of mobility in our days, number of cars and drivers has constantly increased. Driving activity has become a normal day activity. In order to develop safer cars and to move closer to driverless cars, several models where introduced in literature over the years. Steel, due to highly cognitive characteristic, influence of emotions it is very hard to predict the next move of a driver. Alcohol, drug, fatigue of simply lack of attention can transform an easily predictive driving session into one with unexpected moves. Based on collected information from environment, drivers adapt their speed. Based on Fuzzy inference process, and the microscopic model based on a car-following attitude, a model of driving MITRAM is proposed in [47]. In this model a fuzzy neural network is integrated, and the actual driving can be simulated through learning the vehicle's movement data, with a high adaptability for changing the characteristic of learning based on presented data set.

Driving assistance devices are becoming more and more part of regular cars as they add a plus of safety and also of comfort. These devices facilitate the interaction between human driver on one side and cars and infrastructure on the other side. A good knowledge of driving intentions can improve the safety of the system and also could lead to a better automated system. Some typical patterns of behavior drivers can be easily reproduced by driving assistance systems. By using these devices, such as control devices, adaptive cruise control, is guaranteed an adaptation of the vehicle speed to medium traffic. When using the device, the radar system attached to the front of the vehicle is detecting if a vehicle is moving slowly forward. When a slower moving vehicle is detected, the system will slow down the vehicle and keep a safe distance between the vehicle and the vehicle ahead. If the system detects that the vehicle is not in the way of the vehicle, the system will accelerate again to set the vehicle cruising speed. This maneuver allows the vehicle to slow down and speed up traffic autonomously without any intervention from the driver.

The most common task in the management process may be associated with maintaining a constant speed around the vehicle. This is normally associated with free lanes, and sometimes with a cruise control device. A speed limit is imposed and maintained them as well during driving. Scenario undergoes modifications, if a vehicle is being pursuit. In order to maintain safe distance proximity between vehicles, speed must be adapted and thus low. If the driver decides to exceed a machine speed increase will be felt. Depending on the environment and maneuver to overcome, there are three possible situations, further increases up to a calibrated speed, braking smooth and constant motion. Other possible scenarios are represented by movements such as on-off and oscillatory movement. However, these behaviors are influenced by the degree of traffic congestion. In a traffic jam situation, the natural response is represented by a startup- stop script, and in some road sections with different densities oscillatory motion a scenario is plausible. Road traffic for a road segment is a complex system consisting of participating vehicles. To understand and analyze driver behavior, and to achieve a short-term prediction of their actions, a modeling and simulation system is required.

Determining human intentions can be inferred from several possible sources, including the driver's current control actions, their visual scanning behavior, and the traffic environment surrounding them [48]. [48] describes a method similar to hidden Markov models, where the intended actions are modeled as a sequence of internal mental states, each with a characteristic pattern of behavior and environmental state. Patterns of control actions of the driver such as steering and acceleration actions offer information in the process of prediction of driver behavior. In order to be efficient, such a system model must capture the behavior of the overall population but also have facilities to adapt to a particular person or driver. The model proposed in [48] relies on the basic homogenous traffic flow model equation. The interaction between the cars in a given leading car pair is assumed to be a Markov jump process in the acceleration variable of the following car.

An interval compositional model based on a stochastic macroscopic traffic model, used in to vehicular traffic flow modeling is proposed in [49]. In this case prediction of traffic flow it is done without the assumption of uniform distribution of vehicles along the road and can be integrated in road traffic surveillance and control systems real time. In [50] we can find an approach in order to define a general driving model. Interactions between variables in different segments of the road are being described by means of interval analysis, and with some advantages that it can take into account the prior information for the allowed intervals of the system states and noises, such as the minimum and maximum values of the measurements and system noises are usually known in advance and also with the possibilities to include uncertainties. Another benefit of the proposed algorithm was the fact that the measurements of the number of vehicles and speed were received only at boundaries between some segments, but algorithm performed well on real data input analysis.

Conclusions

The subject of data filtering and accurate information gathering is known in literature for several years. Although several viable solutions were proposed so far none of them combines all criteria for accurate traffic surveillance and driving assistance systems. Many of existing algorithms fulfill real-time requirements or high accuracy, but all this with the cost of computation effort.

Table no. 1 summarizes particulate filter variations from survey presented in [25]. Here have been mentioned the most representative algorithms derived from general particle filter. Many of these algorithms suffer themselves several variants. For example SIR algorithm can be implemented in several forms, one of the frameworks is using a threshold value for particle selection. Following this selection, one possibility would be for an amount varying weights layered directly connected with the threshold value. In some applications where a certain pattern (transition state) is repeated, SIS can be the best solution because of the growing importance of the set of particles associated with the real situation, surprised by this pattern.

Name	Description	Comments
Sequential Importance Sampling (SIS)	Propagation and recursive computation of the weights as long as measurements of observations are received	• major problem: the phenomenon of degeneration: some particles are privileged, the rest will have a share a insignificant weight; • a high computational effort to calculate values that will contribute to the final result with values close to zero; • appropriate choice of probability density function, so that minimized variation in weights. [25]
Sampling Importance Resampling (SIR)	Elimination of small particles that Have weight and focus on those who have higher rates; thus will be mainly used for particles which will have the highest probability	• particles are regenerated at each step based on their share value, leading to increased probability; • SIR algorithm can be easily derived from the SIS with a proper choice of importance, and the pace of regeneration, to be applied each time; • improved version of this algorithm is given by the "Auxiliary Sampling Importance Resampling Filter (Assyria)" regeneration with equal weight, which could lead to obtaining a state much closer to the real state; • poor results for a low number of particles, because of regeneration more weight particles, some states may become privileged, [25] • MCMC methods represent an improved type of this kind of

		particle of this type of Particle Filter
Likelihood Particle Filter	Importance density function is given by similarity	• Produces superior results of density estimation methods based on the calculation of posterior importance of this value. [25]

Table 1. Brief comparison of the variations in particle filter

In the distributed implementation of particle filters we are dealing with several variations. The most popular approaches are summarized in table number 2. The main issues involved in addition when using a distributed algorithm is given by: execution time (for this application requires real time response), computing power and accuracy of results (when using wireless sensor-based systems resources are limited both in terms of energy and complexity of calculations) and not least the system scalability.

Name	Description
Distributed computing with the central node	Central node is responsible for retrieving and aggregation of data; here are also being performed calculations
Distributed computation without central node	Lack of central node tries to reduce the excessive communication network by limiting the exchange of information only with neighboring nodes. [27]

Table 2 Ways of implementing particle filters

3 Data Filtering in Traffic Applications: a MCMC approach

3.1 Using MCMC for Error Correction in Dynamic Conditions

Markov chain Monte Carlo is used in general for drawing samples from a multidimensional distribution and estimating expectations with respect to this distribution. They are a collection of techniques that use pseudo-random values to estimate solutions to mathematical problems.

The Metropolis-Hastings (MH) algorithm for MCMC provides a general approach for producing a correlated sequence of draws from the target density that may be difficult to sample by a classical independence method. The goal is to simulate the multi-dimensional distribution $\pi^*(\psi)$ that has the density $\pi(\psi)$ with respect to some dominating measure. To define the algorithm, let $q(\psi, \psi')$ denotes a source density for a candidate ψ' draw given the current value ψ in the sampled sequence. The density $q(\psi, \psi')$ is referred to as the proposal or candidate generating density.

The MH algorithm consists in two steps. In the first step a proposal value is drawn from the candidate generating density. In the second step the proposed value is accepted as the next iterate in the Markov chain according to the probability $\alpha(\psi, \psi')$. If the proposal value is rejected, then the next sampled value is taken as the current value (1).

$$\alpha(\psi,\psi') = \begin{cases} min\left[\frac{\pi(\psi')q(\psi',\psi)}{\pi(\psi)q(\psi,\psi')}, 1\right] & if\ \pi(\psi)q(\psi,\psi') > 0; \\ 1 & otherwise. \end{cases} \quad (1)$$

Typically, a certain number of values at the start of this sequence are discarded after which the chain is assumed to have converged to its invariant distribution. Summarizing, Metropolis-Hastings algorithm generates a random walk using a proposal density and a method for rejecting proposed moves, where calculating the candidate is equal to the current value plus noise. Another possibility is to have samples that are drawn from the proposal density then conditionally rejected to ensure that the samples approximate the target density, or to adaptively modify the proposal density on the fly [41]. Although this method is simple, it is not scalable.

Gibbs Sampling was another improvement made to MCMC algorithms. It requires all the conditional distributions of the target to be known in closed form. Gibbs Sampling has the advantage that it does not display random walk behavior. However, it can run into problems when variables are strongly correlated. When this happens, a technique called simultaneous over-relaxation can be used. In case of hybrid Markov chain Monte Carlo [42], a random walk MH is proposed. It is achieved by introducing an auxiliary momentum vector and implementing Hamiltonian dynamics where the potential function is the target density. The momentum samples are discarded after sampling. The end result of Hybrid MCMC is that proposals move across the sample space in larger steps and are, therefore, less correlated and converge to the target distribution more rapidly. Slice sampling depends on the principle that one can sample from a distribution uniformly from the region under the plot of its density function. This method alternates uniform sampling in the vertical direction with uniform sampling from the horizontal slice defined by the current vertical position.

In current work, it was started by assuming that it is desired to estimate an expectation of a function with respect to the probability distribution. To obtain a Monte Carlo estimation of desired value it can be sample N pseudo-random values from the distribution function. Then the average function value in considered points is used to estimate the result. As number of samples gets larger, the estimation converges to the true expectation. To improve the results, a better technique for estimating expectations named importance sampling was introduced. It produces draws from a different distribution and compute a specific weighted average of these samples to obtain estimates of expectations with respect to function considered. In this case, a Monte Carlo estimated result is obtained by simulating N pseudo-random values from the distribution simply taking the average of function value multiplied by its weight. Often, for numerical stability, every weight is normalized. This implies dividing each of them by the sum of all weights.

Then, it appears the need to specify a probability model for data. It was considered a given space of states, transitions between them and based on this model as probability density matrix (PDM) was developed. Indeed, it was desired to infer from the fixed, observed dataset. A value from PDM based on the likelihood, which is obtained from the probability distribution, is maximized. This is the maximum likelihood estimation based on the previous knowledge.

Following that, a Markov chain that is moving around quickly enough to produce good estimates or "good mixing" is constructed. If the samples are heavily auto correlated it is desirable to redesign the sampling scheme or, at the very least, run the chain for much longer.

If the factor that determines the transition is chosen to be too small, then the Markov chain will nearly always accept its proposed value. However, the proposed value will be usually extremely close to the chain's previous state, so that the chain will move extremely slowly. This determines a very high acceptance rate, but demonstrates very poor performances. On the other hand, if transition factor is chosen to be too large, then the proposed values will usually be very far from the current state. Therefore those proposed values would usually be rejected, so the chain will tend to get stuck at the same state for large periods of time.

3.2 Proposed probability model

It was considered the probability model as a Probability Density Matrix (PDM), which is used to map the transitions between states. In order to improve the filtering efficiency the state space was restricted to the range of moves that a real vehicle can physically achieve. It was considered that states are based on accelerations as they give the best driver behavior reflection. Therefore, five major states were chosen: strong acceleration, small acceleration, constant movement, deceleration and strong break.

Relevant changes in driver behavior between two time intervals considered are considered to be transitions in state space. The transition between a speed greater decrease in the range was considered to be state "- -" a slow vehicle is assigned a lower status "-" is associated with approximately constant speed "0", a slight acceleration in "+ ", and last, a large increase in speed is represented by "++".

Given the physical law of motion, in Fig. 3.1 describes the possible transitions between associated states. State jumps should be regarded as transitions between the last and the current state of the system, based on observations, and previous estimations. Using this transition of states, which are also the bases of Markov chain jump, a PDM is generated.

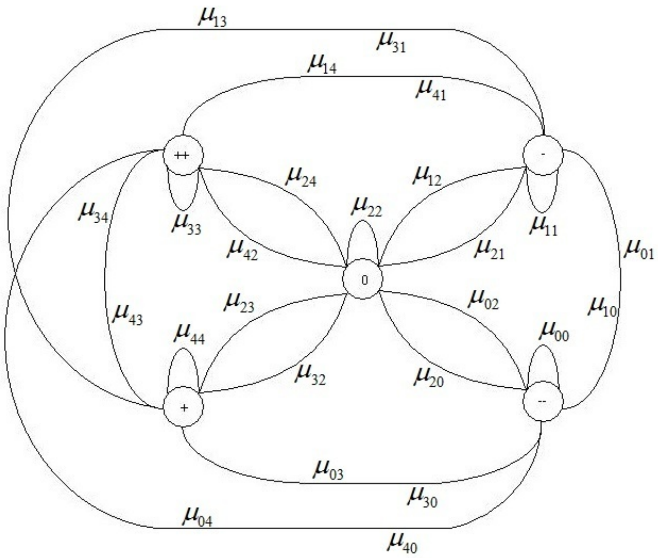

Fig 3.1 Possible transition between states

Because we are dealing with a normal Markov chain, condition $\sum_{i=0}^{4} \mu_{ai} = 1$ must be fulfilled. Determination of possible next state of the car can be seen in terms of Bayesian estimation as a problem of a certain degree of probability of the state of an object x at a time t, with a sequence of observations $z_{1:t}$.

$$\begin{array}{cccccc} & -- & - & 0 & + & ++ \\ -- & \mu_{00} & \mu_{01} & \mu_{02} & \mu_{03} & \mu_{04} \\ - & \mu_{10} & \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} \\ 0 & \mu_{20} & \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} \\ + & \mu_{30} & \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} \\ ++ & \mu_{40} & \mu_{41} & \mu_{42} & \mu_{43} & \mu_{44} \end{array}$$

Fig 3.2 Probability density matrix

At a particular moment of time t it was estimated the most likely transition from current state based on the probability density matrix, associated to each particle. Consequently the update of the PDM was done according to the prediction we made and the observed acceleration at time t. Thus, is compared the estimated state with data arrived from sensors and modify the PDM with values that describe Gaussian distribution where the bonus for the correct state represents curve's peak. The matrix is kept normalized. As a result, in time, a correct driver behavior history is formed, increasing vehicle's probability to hit the right transition in next steps. Filtering recursive Bayesian posterior density calculated, which can be written as:

$$p (x_{t+1} \mid z_{t+1}) \approx p (z_{t+1} \mid x_{t+1}) \, p (x_{t+1}) \quad (2)$$

In order to construct the PDM, we assume that the posterior distribution from the previous step filter p (x_{t-1} | $z_{1:\ t-1}$) is available and can be used for prediction. This is achieved by applying Markov assumptions [44], the prior density, which becomes the posterior density from the previous time step using a density transition (dynamic model):

$$p(x_{t+1}) = \int p(x_{t+1} \mid x_t) p(x_t \mid z_t) dx_t \qquad (3)$$

Approximated value of prior distribution is given by (4)

$$p(x_{t+1} \mid z_{t+1}) \approx \sum_{n=1}^{N} w_t^n p(z_{t+1} \mid \mu_{t+1}^n) p(x_{t+1} \mid x_t^n) \qquad (4)$$

where μ_{t+1}^n is the probability density value derived from Markov dynamic model PDM. Particle filter, the SIR variant is generally based on the following three operations: the generation of new particles (sampling of state space unnoticed), the calculation of weights associated with particles and resampling (removing particles with small weights and replace them with particles with higher rates).

Each vehicle is detected using a separate particle filter. The state tracked by the particle filter is $x_t = \{ p_t, v_t, a_t \}$ where x_t is the vector encoding the state of the object. The pt is used to describe the vehicle position at time t. The vehicle velocity and acceleration are described by v_t and a_t. The vehicle is supposed to move on a planar ground. From the differential equation (5)

$$\dot{p}_t = v_t \quad , \qquad (5)$$

and (6)

$$\dot{v}_t = a_t \quad , \qquad (6)$$

we obtain the relation (7)

$$p_t = p_0 + v_0 t \qquad (7)$$

if velocity is assumed constant and (8)

$$p_t = p_0 + v_0 t + at^2 / 2 \qquad (8)$$

if acceleration is assumed constant. In (7) and (8) the p_0 and v_0 are the initial position and velocity of the vehicle. It was assumed that the velocity evolves from one time step to the next by addition of acceleration

$$v_t = v_0 + at \quad . \qquad (9)$$

The exact geometric shape of a vehicle can be complex and difficult to model precisely. For simplicity it was approximated by a rectangular shape with fixed width and length. The height of the vehicle is not important for these particular driving applications.

3.3 Error correction and observation model

The early stages of this research project were a microscopic model and the tentative of implementing a predictive algorithm able to estimate a series of parameters such as position and speed of a vehicle based on observations data read from the video cameras. This algorithm was implemented based on a particle filter.

Algorithm was used for a camera model with a known error distribution. Also, the absolute error of the car position is obtained with the use of a high definition video camera, which will play the role of the real position of the machine as shown in Figure 3.3. This error was an important role in calibrating the surveillance camera/ video sensors. Based on the calculated position of the filter and relative error, further the sensors can be calibrated, with contribution in reducing errors.

The proposed solution considers the processing of hidden information such as acceleration changes to better reflect driver behavior. Efficiency and accuracy of a particulate filter depends mainly on two key factors. They are the number of particles used to estimate the posterior distribution and the spread function used to re-allocate these particles throughout all iterations.

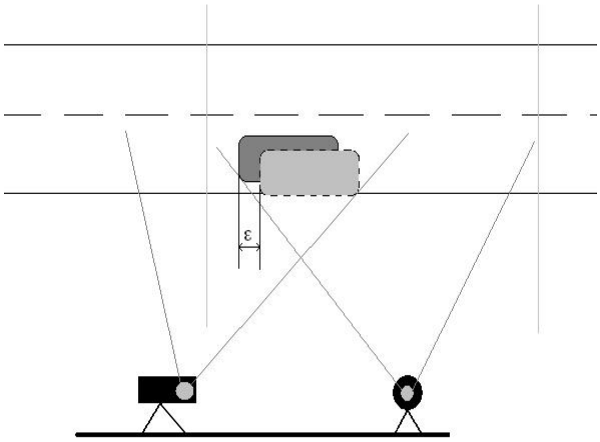

Figure 3.3 A high resolution camera offers real position of the vehicle. The error introduced by the camera low end rate is reflected by the deviation from actual position of vehicles

Key idea is to represent the posterior distribution of states given a sequence of sensor measurements and re-allocate them constantly as new information are available to update the system state estimation. Tracking problem can be seen from the perspective of Bayesian estimation as a matter of a degree of confidence of a state x_t of an object at time t and at a given step in a series of observations $z_{1:t}$ recursive Bayesian filtering calculates the posterior density which can be written using Bayes rules that:

$$p(x_{t+1}|z_{t+1}) \approx p(z_{t+1}|x_{t+}1)p(x_{t+1}) \qquad (1)$$

In order to build a particulate filter, it is assumed that the posterior distribution from the previous step of filtering p (x_{t-1} | $z_{1:t-1}$) is available and can be used to build the prediction based on a priority density current filtering step. This is done by applying Markov assumptions [20], prior density, posterior density that is propagated from the previous step using a transition time (dynamic model):

$$p(x_{t+1}) = \int p(x_{t+1} | x_t) p(x_t | z_t) dx_t$$

This probability value is used in the following steps to make predictions. Generally, particle filter algorithm goes through the following steps:

(1) Initialization: create an initial set of particles (standards) based on systems' equations;
(2) Prediction: with particles from the previous steps determine the future most likely position of the observed vehicle;
(3) Update weights: the weight calculation of each particle;
(4) State estimation: estimating the state of each object is made in each step;
(5) Regeneration: The particles with higher weight generate other particles, while those with a lower weight are removed.

To fill the previous probability model we define the following observation model. We compute the observation model p(z | x) using a given vehicle state x and the measurement z. Next, the vehicle is positioned according to x. The representation of the vehicle is described in section 3. Then we include points, surrounding the rectangle shape, representing particles. All particles are spread within a predefined distance around the vehicle. Assuming that each particle is a ghost of the actual vehicle we expect that all particles would eventually fall into line and estimate the correct position.

The actual observations, denoted by z, consist of data coming from a wireless network sensor. Due to the noise in measurements resulting from sensor sensitivity, resolution and possibly bad environment conditions: fog, snow, rain, the vehicle's motion is typically better modeled by probabilistic densities. We assume that the system and observation model is driven by Gaussian random noise. Therefore, the particle filter is appropriate to estimate the motion state.

Each particle's weight is calculated according to the mentioned measurement model, w ≈ p (z | x). The basic effect of this equation is to reward particles according to the old state and new measurements. Therefore, particles that are closer to the observed position and those that have a right state transition receive higher weights. For example, the jump from a small accelerating state to a large one is a more probable transition in reality than a transition from a large acceleration to deceleration, so it gets a higher bonus. To maintain a consistent sample the new importance weights are set to

$$w_k = w_{k-1} \frac{p(z_k \mid x_k) p(x_k \mid x_{k-1})}{\Pi(x_k \mid x_{k-1}, z_k)}, \qquad (10)$$

where $p(z_k \mid x_k)$ is the likelihood of making the observation z_k given that the object is at x_k location. The weights are normalized so that they sum up to 1,

$$w_k = \frac{w_k^*}{\sum_1^N w_k^*}, \qquad (11)$$

where N is the number of particles. Finally we estimate the position as weights sum from all particles.

3.4 Algorithms steps

3.4.1 Initialization step

As a recursive algorithm, the particle filter needs initialization to track a state vector as the observations arrive in sequence.

At starting point the set of particles is created by sampling from the observed position of the vehicle. The standard deviation for the initial distribution is set quite high to increase the spread of particles. The weights attached by the particle filter are balanced. Each particle gets the same weight as 1/N, where N is the number of particles.

Each particle state is updated according to the prior density $x_t \sim p(x_t \mid x_{t-1})$. This is calculated separately for the position and velocity. Particles have the same dynamic model as vehicle's model presented in (7), (8) and (9).

Part of each generated particle we have considered the Probability Density Matrix (PDM), which is used to map the transitions between states. We have considered that states are based on accelerations as they give the best driver behavior reflection. Therefore, we consider five major states: strong acceleration, small acceleration, constant movement, deceleration and strong break. At a particular moment of time t we estimate the most likely transition from current state based on the probability density matrix, associated to each particle. There are to possible ways to generate the PDM matrix: with a fix value of 0.2 corresponding for each state transitions, even if transitions from states like strong acceleration to strong brake are nearly physical impossible in a small time sample, or, the second possibility is to have pseudo-random value, with unique condition that the sum of values on each raw to be equal to 1.

3.4.2 Resampling

The SIS particle filter, discussed by this point may suffer from sample degeneracy problem, where almost all particles have negligible weights after a few iterations. This implies that most of the computation time will be spent on updating particles whose contribution to the posterior probability density function approximation, $p(x_t \mid z_t)$, is almost zero. This means that the sampled particles contain little information about the true target state and the tracking performance will consequently be degraded. In order to reduce the degeneracy problem, resampling has been incorporated in the particle filter to eliminate the particles that are far away from the observed position. A threshold is applied to detect the particles that leave the correct tracking state. More exactly, at a moment of time t, particles with negligible weights, namely those that are further from the observed position will be discarded after resampling and replaced with new particles initialized with weights equal to 1/N, where N is the number of particles. We chose the threshold value with respect to filter root mean square error and the resource consumption. Detailed discussions on this aspect could be found in the next sections.

4 Adaptive Filtering based on Driver Behavior

4.1 Observation Model of Dynamic Systems: Intelligent Driver Model

Improving traffic management applications is directly related on the way traffic models are chosen. Theories related to traffic flow modeling try to describe in a precise, mathematical base the interactions between vehicles, drivers and infrastructure. These theories are an indispensable element of all traffic models and analysis tools that are used in design and operation of streets and highways, and the behavior of participants in various traffic situations. Microscopic traffic simulation models differ significantly from the conventional "aggregated" traffic model such as macroscopic ones. This is because instead of shaping traffic flows, the model simulates the microscopic behavior of all individual vehicles in the network. Macroscopic models employ equations on the conservation of flow and on how traffic disturbances broadcast through the system like shockwaves. They can be used to predict the spatial and sequential extent of congestion caused by traffic demand or incidents in a network. However, they cannot model the interactions of vehicles on alternative design configurations. Mesoscopic models combine the properties of both microscopic and macroscopic simulation models. These models simulate individual vehicles, but describe their activities and interactions based on aggregate relationships.

In general, when it comes into discussion the need of modeling systems, a predefined set of finite states is used. In most of the cases, when dynamic systems are involved, state transitions are not matching a well-defined pattern and thus are becoming unpredictable. In this case, the main idea is to model this systems as stochastic dynamic systems where to encode a probability distribution over one key component [5]. In [6], among a very good survey on possible methods of adaptation of particle filters, it is presented another adapting method for sampling in continuous time, by introducing an auxiliary variable. Using this method and performing particle filtering in a higher dimension, unreliability of the empirical prediction density in the tails of the distribution was diminished; adaptation of SIR rejection and efficiency of SIR or MCMC sampling can be done without slowing the running filter. Still, difference in comparison with classical approach is small, and it does not worth the computational price effort when it comes to limited resources of wireless sensors.

It is well known that the performance, accuracy and convergence of PF methods depend mostly on the model of observation included in the algorithm and also on the number of particles used in estimation. The observation model mainly depends on target descriptions and/or sensor models. A high number of particles will lead to a better accuracy, with the cost of computational effort. Nevertheless the quality of particles themselves and their weight it is also important. In this sense, a balance compromise between error estimation and computational effort has to be done.

For implementing a PF based on a top-down strategy it is very important to follow a certain motion model, a white noise acceleration model, a random walk model or an adaptive motion models which can be learned from the arrival of the new observations. Such an example it is mention in [28] to be the Adaboost particle filter incorporates the detection hypothesis in the proposal distribution.

Adaptive implementation it was proven to be a reasonable solution in this sense. Several versions of this approach are available in literature. In [58] we can find one of them. Here authors present a self-adaptive version of the particle filter that uses statistical methods to adapt the number of particles and the propagation function for each iteration, with the benefit of the same computational effort. The number of particle used in the algorithm is adaptive, with the only condition that for practical implementation a minimum number of particle is being maintained in order to ensure the convergence of the filter. Also in their implementation a threshold error value of filter and a confidence level it is set. Authors also proved that using an approximation of the observation (movement) model can lead to degeneracy of the filter and as a consequence there is a large mismatch between the dynamic prior and the posterior distribution, which produces an inefficient allocation of the samples and the estimate without adapting the importance function needs a larger set of samples to populate the relevant parts of the posterior.

Adaptive algorithm presented in [7] is built on four basic ideas. The first preliminary adaptation combines those into a single strict estimation method. The second tries to estimate the mixture of normal distributions based on previous drawing and use as independent MH algorithm proposed distribution on both sides of the adjustment process. The third algorithm concerns the common achievement of the estimate, especially in pre-adaptive stage, a stage known as intensive adaptation. Last idea is to verify the theoretical conditions for ergodicity extraction system during adaptation strict values. To implement these ideas is fast and robust estimation of mixture parameters. In [59] is studied the case where the Markov chain transition kernel depends on unknown parameters and also the opportunity of building a particle filter to estimate the unknown parameters and partially observed Markov chain based on an adaptive estimation. In this case the results are showing an optimal filter convergence in time and the number of particles tends to infinity, with strong limitations in case of restricted resources.

Micro simulations describe in general three types of behavior: acceleration, deceleration and lane changing. Based on the fact that is built on a "car-following attitude", we have decided to investigate more the Intelligent Driver Model (IDM). Model describes the traffic state at a given time by the positions, velocities, and the lane index of all vehicles. The decision of any driver to accelerate or to brake depends only on his own velocity, and on the front vehicle immediately ahead of him. In general this model simulates single-lane main road and simple lane-change model for the on-ramps. There are seven parameters involved: desired velocity, safe time headway, maximum acceleration, comfortable deceleration, minimum distance, and jam distance and acceleration exponent.

IDM was successfully used in developing collision avoidance systems. A particular example of using an adaptive driver model in this way can be found in [60] with application in indirect collision avoidance. Here, on an artificial neural network platform the model inputs are chosen to be the past history of throttle angle, controlled vehicles' speed, range and range rate to the front vehicle whereas the model output is chosen to be the current throttle angle. The past history of the throttle dynamics plays a critical role in reducing the deviation of the error correction.

In order to perform well, prediction algorithms need to incorporate a model of the observed system. Depending on the specific model and the way this model is chosen, it has a huge influence on the success of the algorithm. If the model is well chosen, then the way particles move around in state space, matches the way the values of measured variables change over time. Indeed, each update or time-step represents generally a fixed amount of real time. On repetitive iteration of process this can converge to a desired value. Unfortunately, this is happening only in case of discrete-time dynamics. In case of real time it can follow a divergence behavior. Moreover, the driver behavior is hard to predict and hard to be modeled.

Therefore, the desired model will act in continuous time and will integrate an approximation of the driver behaviors. In our work we concentrate on the aspects of dynamical state process changes, such as driving activity, combined with prediction algorithms. In order to be able to construct a strong prediction algorithm the main concern is to incorporate the model of observation. Starting from the physical equations of movement and we have developed a simplified model based on initial velocity and acceleration of car on time axis. The Markov transitions state space kernel was constructed linked with car acceleration. We have considered a finite space of only five states in order to describe in a discrete manner the continuous transition of the gas pedal and, respectively with movement of the brake pedal.

In current work, the proposed relies on a mixture of particle filtering method and the classical sequential Markov chain Monte Carlo algorithm for performing state estimation. The purpose is to reduce the noise that affects data read from sensors in order to determine the correct position of a moving vehicle. The state transition between behavior patterns is then predicted instead of just determining the next position. Driving behavior is modeled by five major car movement states as strong acceleration, smooth acceleration, constant movement, smooth deceleration and strong break. Therefore, considering the transition between the physical states we have implemented the particle filter based on a suitable MCMC approach. The history of driver behavior is the main issue considered when the new state is predicted. Particles are being considered as ghosts of the cars, having an associated probability density matrix, speed and positions. The probability density matrix is used to map the transitions to the space states as in [A].

Therefore, the desired model will act in continuous time and will integrate an approximation of the driver behaviors. In this thesis accent is putter on the aspects of dynamical state process changes, such as driving activity, combined with prediction algorithms. In order to be able to construct a strong prediction algorithm the main concern is to incorporate the model of observation in PF. Starting from the physical equations of movement and we have developed a simplified model based on initial velocity and acceleration of car on time axis. The Markov transitions state space kernel was constructed linked with car acceleration. We have considered a finite space of only five states in order to describe in a discrete manner the continuous transition of the gas pedal and, respectively with movement of the brake pedal. Due to the fact that pedals movements in a car are physically limited by an initial and a maximum position, both for acceleration and breaking, we have consider the magnitude of car acceleration between a lower and an upper limit. Figure 2.2 depicts how the state space where chosen in the proposed algorithm.

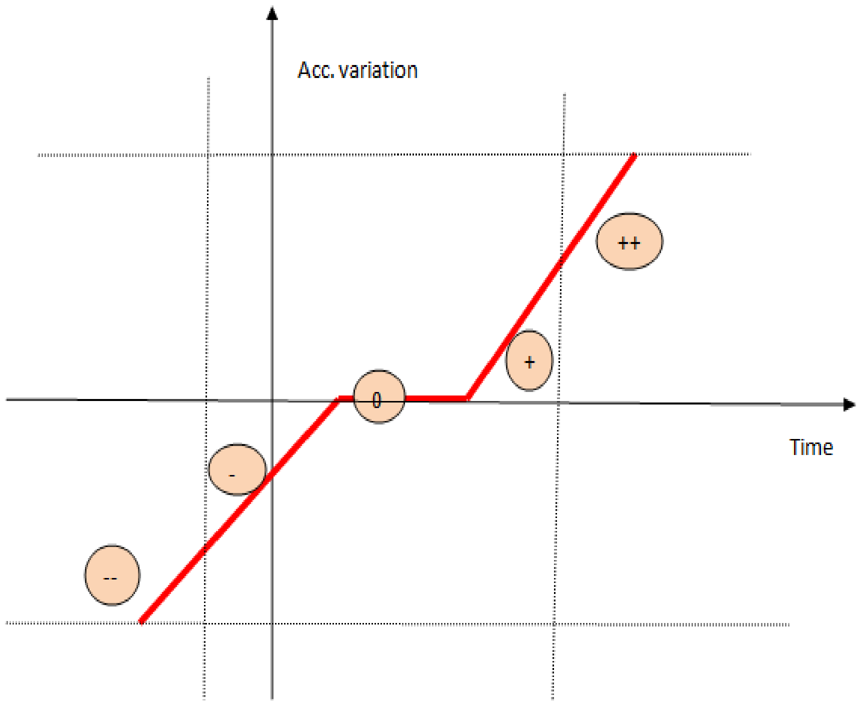

Figure 2.2 Discrete state space for car acceleration.

The key concept of the method is deriving from Markov kernel and transition determined on hidden parameters such as acceleration of the observed target. We consider a given space of states based on accelerations and transitions between them, and use this model as probability density matrix (PDM). The transition between state at time t-1 and t is given by the values from this matrix. Also the most likely transitions are determined based on the PDM.

The implementation of an adaptive algorithm in this case is following two orthogonal steps. The first one is related to the number of particle contained by the filter and the second one is the adaptation of PDM values.

MCMC is used in general for drawing samples from a multidimensional distribution and estimating expectations with respect to this distribution. The implementation of an adaptive algorithm has two steps [A].

The first one is related to the number of particle contained by the filter and the second one is the adaptation of PDM values, more precisely on interpreting the value of acceleration. For this step a detailed IDM description of observation model was considered [66,67].

Each of the particles contained by the filter is described as successions of state vectors. The state at sample time t is represented by $x_t = \{ p_t, v_t, a_t \}$ where pt is used to describe the particle position at time t, v_t the velocity and at acceleration of the particle.

A particle filter algorithm consists mainly on these steps: initialization, estimation and update. The estimation of the posterior distribution is usually done in three main steps: sampling, weighting, and re-sampling. The sampling step consists of taking samples from the dynamic prior distribution. Re-sampling step is usually applied to avoid the degeneracy of the particle set.

The idea behind it is to start the filtering process with a relative small number of particle, but yet reasonable. In the next steps, the filter will "watch" its deviation from the observation, and whenever this will be over a calibration threshold, a new infusion of particles will be perform, up to well determined number. Also, if the maximum number of allowed particles was reached, the particles with minimum weight are trimmed. This way a compromise in terms of computational price and filter performance is established. The need of resampling can be shortly explained by the fact that keeping old particles around forever without resampling them, they will drift around according to observation model. Highly unlikely particles will be kept around and transitioned to more unlikely states, in a so-called "particle depletion" [68].

The second adaptation step consists of adapting the meaning of PDM values, more precisely on interpreting the value of acceleration. For this step a detailed description of observation model is needed.

The proposed approach relies on a mixture of particle filtering method and the classical sequential Markov chain Monte Carlo algorithm for performing state estimation. The purpose is to reduce the noise that affects data read from sensors in order to determine the correct position of a moving vehicle.

The approach is different in this case. The state transition between behavior patterns has to be predicted instead of just determining next position. Using a fuzzy implementation driving behavior is modeled by five major states: strong acceleration, smooth acceleration constant movement, deceleration and strong break. By considering the transition between the physical states, we have implemented the particle filter based on a suitable MCMC approach. Thus the history of driver is the main issue considered when the new state is predicted. Particles are being considered as ghosts of the cars, having an associated probability density matrix, speed and positions. The probability density matrix is used to map the transitions to the space states. Indeed, for each particle we use car acceleration to establish the possible state transitions. More details on this issue are presented in [69]. Basically, the algorithm computes and maintains sets of particles to describe the historical and present states of the model. The filtering technique consists of the following steps: initialization, prediction, sensor reading and update, filtering error and smoothing. In the initialization step the set of particles is created by randomly sampling from the observed position of vehicle. In prediction step, the state at time t + 1 is estimated using the state at time t. This is done using the probability density matrix distributions associated with each particle. In the filtering procedure we use the data from sensors, which arrive at time t + 1 and are being compared with the estimated state at time t + 1. For each particle a weight based on the values of the observed variables at time t + 1 is assigned. Particles closer to the observed values from sensors and with right state transitions receive higher weights compared to the particles that are further from the observed values. Weights are normalized so they sum up to 1. In the smoothing step we use the current state distribution to correct the state at previous times. This is done via the probability density matrix, and thus we update the history of each particle. This is performed to reduce variance of the filtering step.

The key concept of the method is deriving from Markov kernel and transition determined on hidden parameters such as acceleration of the observed target. It was considered a given space of states based on accelerations and transitions between them, and use this model as probability density matrix (PDM). The transition between state at time t-1 and t is given by the values from this matrix. Also the most likely transitions are determined based on the PDM.

The need of implementing an adaptive algorithm can be easily justified after analysis of Figure 4.1. For this example, a pseudo-chaotic movement of car was considered and a non-adaptive implementation of MCMC PF proposed method. Jumping through a finite number of states, without adapting particle acceleration can lead to a slow convergence of filter in early stages, and also to poor performance of filter in the first steps. Also, this is the reason why the filters in not convergent to zero but to a finite distribution.

The implementation of an adaptive algorithm in this case is following two orthogonal steps. The first one is related to the number of particle contained by the filter and the second one is the adaptation of PDM values.

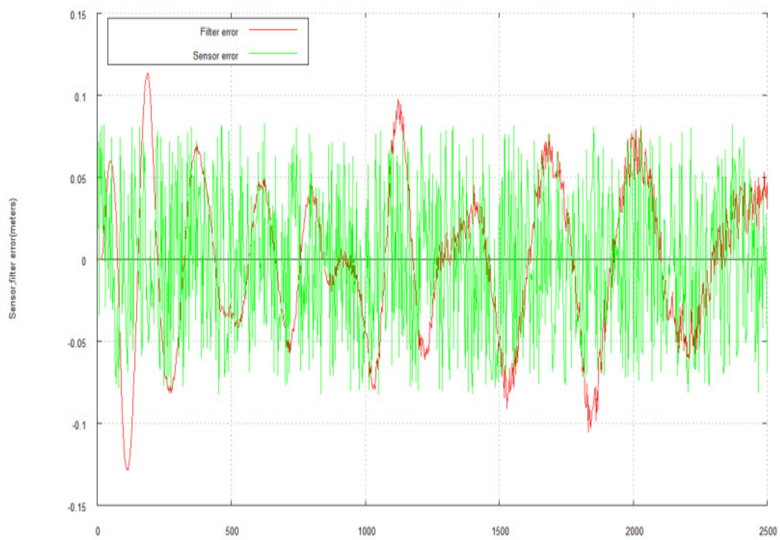

Figure 4.1 The error of the common MCMC filter in combination with driving behavior

Due to the fact that is known that a significant number of particles will lead to an improvement in filter performance, but also will imply a higher computational effort, an adaptive SIR algorithm was implemented.

By implementing an adaptive driving model based on observation, and having as foundation the IDM [13], acceleration dv/dt of a given particle depends on his velocity v, on the distance s to the observation, and on the velocity difference dv from own particle velocity and calculated velocity of observation,

$$\frac{dv}{dt} = a\left[1-\left(\frac{v}{v_0}\right)^{\delta}-\left(\frac{s^*}{s}\right)^2\right], \quad [9] \qquad (1)$$

where

$$s^* = s_0 + \frac{vdv}{2\sqrt{ab}}. \quad [9] \qquad (2)$$

In case of approaching, this dv is becoming positive. The acceleration is divided into a "desired" acceleration needed to come closer to observation

$$a\left[1-\frac{v}{v_0}\right], \quad [9] \qquad (3)$$

and braking deceleration

$$-a\left(\frac{s^*}{s}\right), \quad [9] \qquad (4)$$

introduced by the fact that particle is coming more close to observation. The braking term is based on a comparison between the "desired dynamical distance" s*, and the actual gap s to the observation. [13] If the actual gap is approximately equal to s*, then the breaking deceleration essentially compensates the free acceleration part, so the resulting acceleration is nearly zero. This means, s* corresponds to the gap when following observation. In addition, s* increases dynamically when approaching slower moving observation and decreases when this becomes faster. As a consequence, the imposed deceleration increases with decreasing distance to observation, increasing own velocity, increasing velocity difference to observation. The acceleration coefficient δ affects how the acceleration changes when it approaches v0. When δ = 1, we have exponential approach, but when δ is very large, dv is constant with acceleration a, and drops to 0 when it reach v0. The parameters for adaptive model are somehow easily intuitive, as we desire to have a dv equal to zero, and also s* as close to zero, meaning that the filter relative error is becoming zero. Concerning a and b from equation (2) their meaning is given by an average acceleration in everyday traffic, respectively a comfortable braking deceleration in everyday traffic, s0 minimum distance to observation. [13]

4.2 The Algorithm Steps for Adaptive Implementation

1). *Initialization step*

(1) Particles generated, form a cloud around each car. Each particle has a random position in a given range, depending on the sensor error.

(2) Uniform distribution of particles weights is initially considered.

(3) Initialize the PDM for each particle

2). Prediction phase

(4) For each component do:

- *Determine the most probable transition state*
- *Based on adaptive observation model determine the value of acceleration*
- *Calculate position according to the estimated state*

3). Sensor reading and update

(5) For each component do:

- *Compare the estimated position with the position read from sensor*
- *Update the weight of each particle according to the estimation accuracy (made in terms of jump state and estimated position)*
- *Update the PDM according to the estimated jump accuracy. Use a Gaussian distribution to compute the correct factor.*
- *Update parameters needed of construction of adaptive observation model.*

- *Calculate the position as weights sum from all particles.*
- *Determine filter error.*
- *If filter error is above desired threshold and particle number is in given range increase the number of particles. In case the maximum number of allowed particles is reached, regenerate particle with minimum weight*.

The main purpose was to develop an efficient common platform which will work for a filter and also in case of prediction algorithm. The filter is able to reduce the noise which affects data coming from sensors. In case of prediction algorithm, it should be able to estimate accurate the next parameters of a given car, based on observation. The proposed platform is based on a modified Particle Filter Monte Carlo Markov Chain. The core is represented by a five state matrix, called Probability Density Matrix (PDM) [1]. PDM encapsulates the transitions between states, and thus it reflects the driver behavior. PDM it is considered to be stochastic, and during adaptation phase, values are being kept normalized. It was proved that integrating observation model in platform will reduce filter error [1].

The algorithm runs several particles (in case of adapting version, this number can vary from a minimum number up a maximum limit. Each of the particles contained by the filter is described as successions of state vectors. The state at sample time t is represented by x_t = { p_t, v_t, a_t } where pt is used to describe the particle position at time t, v_t the velocity and at acceleration of the particle.

Based on physics law we can consider

$$p_t = p_{t-1} + v_{t-1}t + at^2/2 \quad (1)$$

and also we note by zt = { p_{zt}, v_{zt}, a_{zt} } observation coming from sensors.

Based on implemented algorithm we can express the calculated (estimated) position of the car as:

$$P_{est} = f(x_t, \varepsilon(t)) * w_t \ ,$$

where $\varepsilon(t)$ represents the sensor error, as time distribution (2)

For each particle, transition between states depends mainly on the previous considered state and observation coming from sensors. At each step an update based on the received observations is performed, and this is reflected in the next considered state of the particle:

$$x_k = p(x_{k-1}|z_{k-1}) \quad (3)$$

Expanding equation (2), and based on [1] we can state that

$$p(x_k) = p(x_{k-1}) + \mu_c(z_k, x_k) \quad (4),$$

$$\mu_c(z_k, x_k) = \mu_{cp} f_1(\Delta(p_{zk}, p_k)) + \mu_{ca} f_2(\Delta(a_{zk}, a_k)) \quad (5)$$

Both f_1, f_2 are functions which return a constant according to a Gaussian distribution. Based on the fact that for each next state the most likely transition is made based on the best probability possible we can rewrite equation (5) in the following manner:

$$p(x_k) = \max\{p(x_{k-1}) + \mu_c(z_{k-1})\} + \mu_c(z_k, x_k) \quad (6)$$

For normalization step, weight of each particle are being computed, and is being updated according to the performance of each particle

$$w_k = w_{k-1} \frac{p(z_k|x_k)p(x_k|x_{k-1})}{\pi\,(x_k|x_{k-1}, z_k)},\ w_0 = \frac{1}{N}, N = \text{number of particles} \quad (7)$$

We introduce following notation:

$h(t) = x_{zt} - x_t$ (8),

as the difference between observation and particle. By introducing IDM in model observation and thus in (2), the target is that:

$\lim_{t\to\infty} h(t) = 0$ (9),

Equation (9), in other words, states that particle will tend to follow observation, and on weight average will give a better approximation of position.

5 Experimental results

In order to determine the efficiency and to validate the proposed algorithm experiments where performed with a simulation-based system. Simulated sensor values provide noisy information about the monitored vehicle position. Different types of scenarios were considered in order to analyze the algorithm performances.

The set of tests for this algorithm is based on driving simulation patterns through a Java traffic simulator. This simulator is able to manage multiple behavior patterns of drivers, such as constant or free movement constant, mild acceleration within a given calibration value combined with a constant motion, including a negative acceleration (deceleration to a limit in order to track a vehicle in traffic), and oscillator behavior within two known values of speed. It was designed to manage multiple distribution, such as linear, Gaussian or random. Maximum error monitoring device falls within a radius of 1 m. It is known that, when implemented effectively, particle filters require a cost calculation in proportion to the number of particles. Therefore it was considered a variable number of particles, in order to determine the influence of this factor in determining the value of absolute and relative error.

5.1 Scenario for Non-Adaptive implementation

The first set of test sets were constructed in the following conditions. We have considered the ideal case of simulating a vehicle with a constant acceleration through all simulation period. The PDM was initialized with a constant value of 0.2 for all the considered states. The test was performed on a PF with 500 and 1000 particles. This ensures a based for comparison with results from the other test sets. The obtained results are presented in Fig 5.1 and Fig. 5.2.

In order to determine the PDM influence we have done simulations using a random PDM, but keeping the matrix normalized.

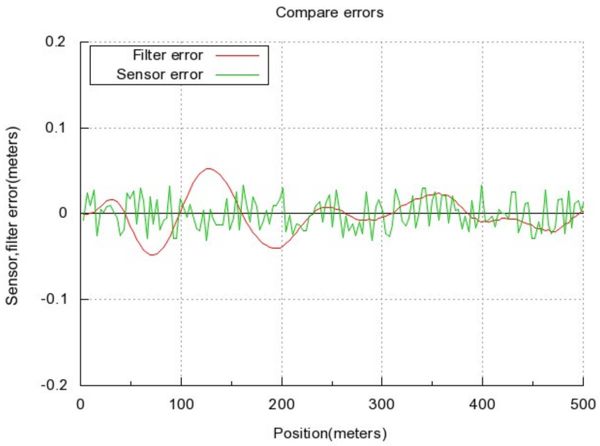

Figure 1. Filter error for PDM initialized with constant values, 500 particles and constant acceleration

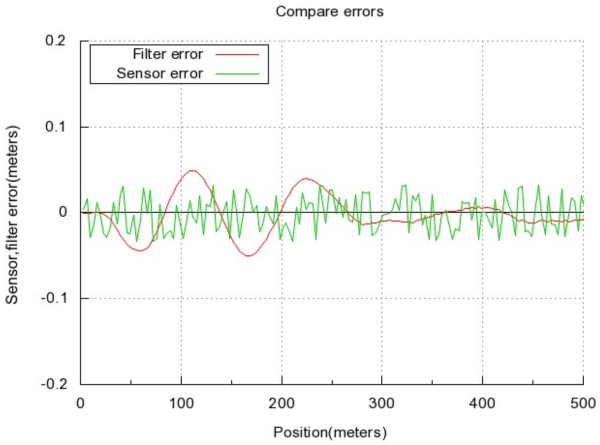

Figure 2. Filter error for PDM initialized with constant values, 1000 particles and constant acceleration

The results are shown in Fig. 5.1 and Fig. 5.2. In this case we have performed tests with 200 and 1000 particles. The car was considered accelerating. These diagrams demonstrate a strong dependence between the filter convergence and the number of particles considered.

The next set of tests consists of the same initial conditions for the PDM and the same number of particles, but the ideal case that the vehicle is moving with a constant velocity during the simulation time, was considered. Graphs for this experiment are presented in Fig. 5.3 and Fig. 5.4.

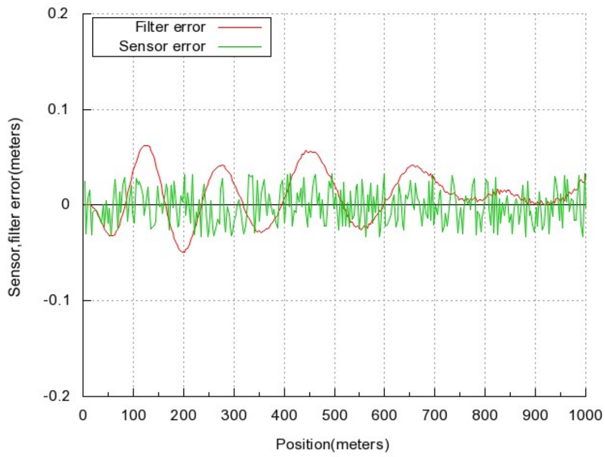

Figure 5.3. Filter error for PDM initialized with random values, 200 particles and constant acceleration

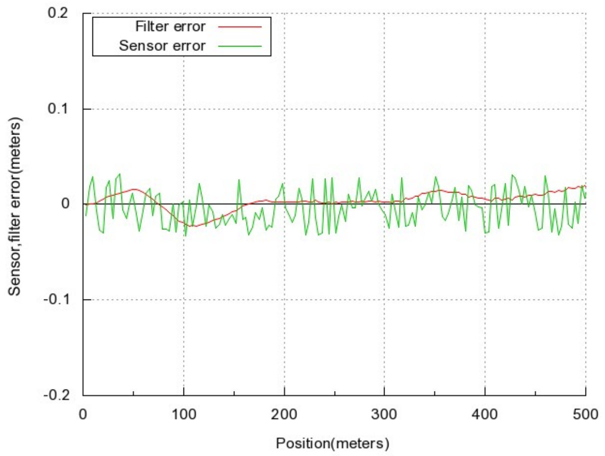

Figure 5.4. Filter error for PDM initialized with random values, 1000 particles and constant acceleration

Another set of tests were performed having in consideration that the car has a varied acceleration. The PDM was initialized with random values. Results for both 500 particles and 1000 particles are shown in Fig. 5.5 and Fig. 5.6.

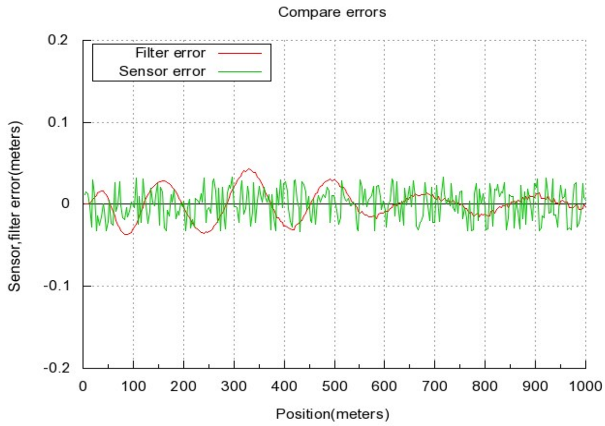

Figure 5.6. Filter error for PDM initialized with random values, 500 particles considered, varied velocity of car

It can be easily observed that in case of a pseudo-chaotic movement, the filter is responding with a inertia, and that error correction requires several steps, but still filter relative error is maintained mostly under the sensor error. Again, it is proved that a bigger number of particle determine a better performance of filter.

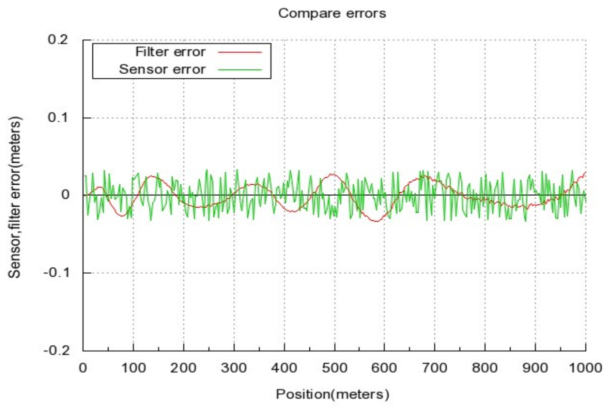

Figure 5.7. Filter error for PDM initialized with random values, 1000 particles and variable acceleration

For all tests presented above we have performed a resource consumption analysis. In perspective of implementation on low resources WSN hardware, was concentrated on memory and on execution time. The conclusion of this analysis can be drawn from Fig. 5.8 and Fig. 5.9. As expected, the resource consumption increases directly proportional with the number of particles used.

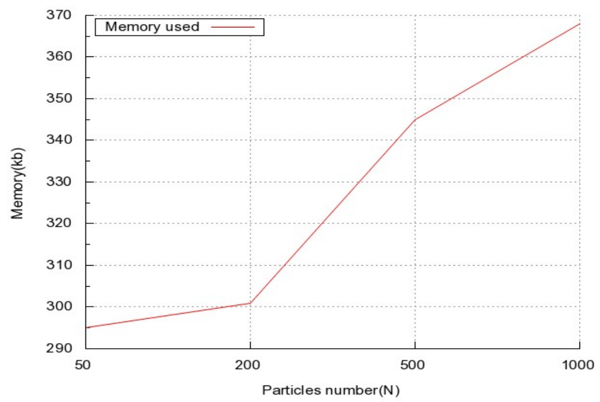

Figure 5.8. Memory consumption for various PF size [B]

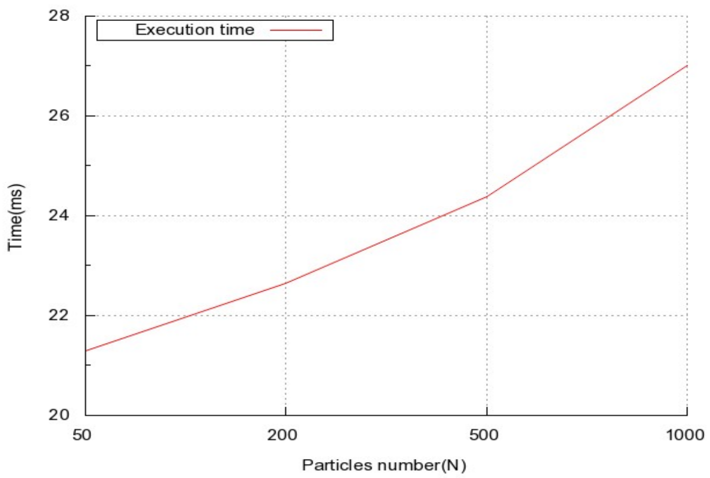

Figure 5.9. Execution time for various PF size [B]

The particle filter implementation is evaluated towards its accuracy considering different parameter choices. All the experiments discussed below use the same standard deviation for the measurement noise. At a particular time t, the tracker's state is computed as the mean over all particles. Then the error, in terms of location, at time t is computed as the distance between the estimated value and the ground truth-value.

The experiment investigations where continued by another set of simulation the tracking accuracy using a fixed number of particles (N=1000). In this test, we considered that the vehicle was moving with constant velocity through all simulation period. The PDM was initialized with a constant value of 0.2. The experiments results plotted in Fig. 5.10 show the absolute sensing and filtering error. It can be observed that sensors have a standard deviation around 0.03 meters while the filtering has a worse case error over 0.05 meters. For example, at 100 m from the starting point, the particle filters estimates our vehicle position about 0.05 meters ahead from the real position. After 200 m the filter starts to converge to the real pose.

While the vehicle state and the PDM remain unchanged, Fig. 5.11 presents the sensor and filter error for the Sampling Importance Resampling algorithm. It can be seen that the particle filter obviates from the true path in the first part of the simulation owning to the fact that the filter needs about 200 m to keep the history of the car.

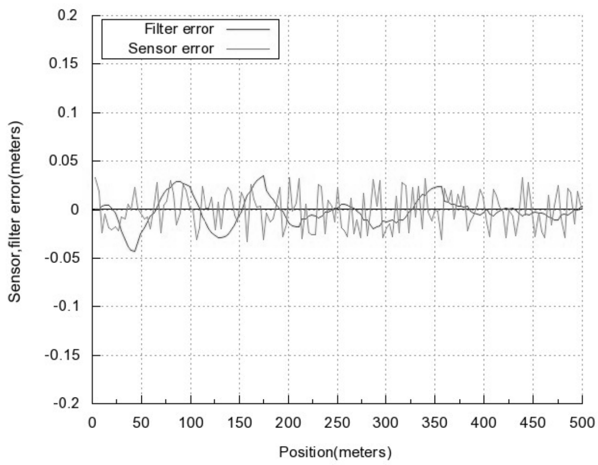

Figure 5.10. Filter error for 1000 particles, constant PDM and constant velocity for the SIR algorithm [B]

In contrast, we present another test where we considered a smaller threshold for the resampling step. In other words the resampling was performed much more many times, even from the first meters of the simulation. In spite of the fact that the particle filter maintains a good estimation all the way, the solution is not the best one as the resource consumption, the memory needed, the execution time increases drastically. The result is shown in Fig. 5.12

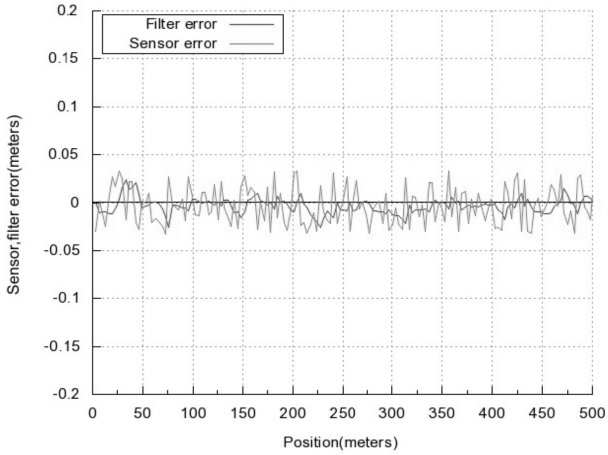

Figure 5.11. Filter error for 1000 particles, constant PDM and constant velocity for the SIR algorithm, using a small threshold [B]

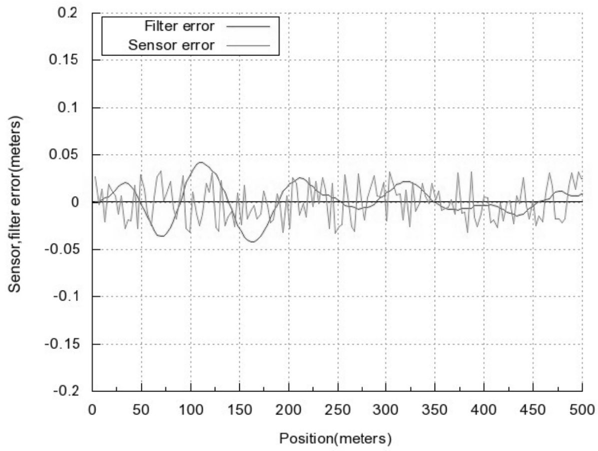

Figure 5.12. Filter error for 500 particles, constant PDM and constant acceleration for the SIS algorithm [B]

To observe the influence of parameters on both algorithms we performed another set of test. We used a larger number of particles, 500 and a constant PDM. The car acceleration was considered constant for the entire simulation period. As a consequence, vehicle speed increases after each simulation step. At starting point we used following initial conditions: vehicle speed was 50 km/h and the time step was 0.3 s. As expected, a smaller number of samples and the increasing velocity for the car cause a worse position approximation. Results are shown in Fig. 5.12 [B]

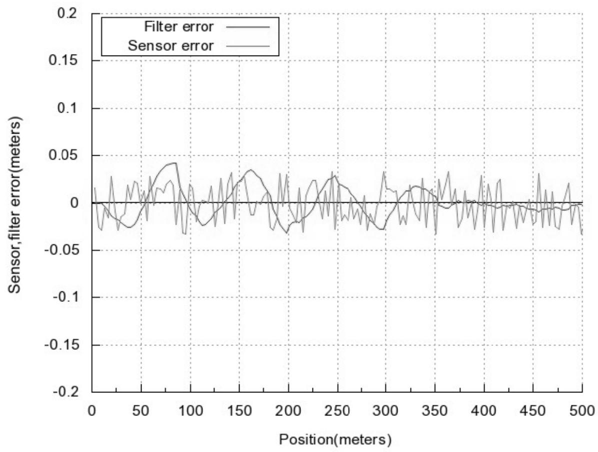

Figure 5.13. Filter error for 500 particles, constant PDM and constant acceleration for the SIR algorithm [B]

Next experiment was conducted under the same scenario as the previous one: the behavior of the car was considered accelerating, each particle had associated a constant PDM and PF was initialized with 500 particles, but the algorithm considered was SIR particle filter. We can observe in Fig. 5.13 that PF estimates vehicle position ahead and behind relative to the real position for several steps. Filtering error decreases in time and after 300m covered filter starts to stabilize and converge to the real pose, giving a better approximation than previous test.

To obtain a statistical comparison between regular PF and the SIR particle filter, the experiment for both algorithms was repeated 10 times with the following initializations: the simulation distance was 1000 m, random PDM for each particle, the car had constant velocity. The graph 5.14 illustrates filter output error for both algorithms, defined as root mean square error. This parameter gives an indication of how much the estimated location deviates from the true position. A high MSE value hence always reflects an inaccurate tracking ability. Fig. 5.14 shows that the proposed resampling algorithm significantly outperforms regular PF. On the other hand, the experiment investigates the relationship between the tracking accuracy and the number of particles. We can observe that the tracking accuracy increases when the total number of particles increases from 50 to 1000, thus a big number of particles are required to correct sensor error. [B]

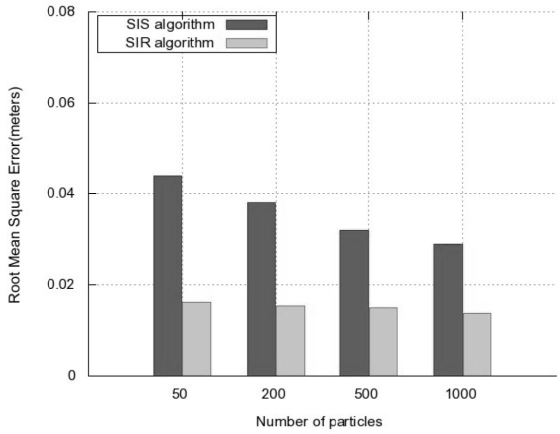

Figure 5.14. Root mean square error comparison between presented algorithms for different number of particles. The car velocity used in simulation was considered constant [B]

The fig. 5.15 illustrates the effectiveness of the resampling approach over the sampling importance algorithm considering different initial conditions. The vehicle movement wasn't ideal as previous experiment (constant velocity), but it was various; the simulation time was three times larger, the PDM was randomly initialized, keeping the matrix normalized. The proposed algorithms give an appropriate estimation when using a large number of particles, but at the same time it requires more computational power while a small number of particles affect significantly the performance of the proposed sampling importance algorithm. [B]

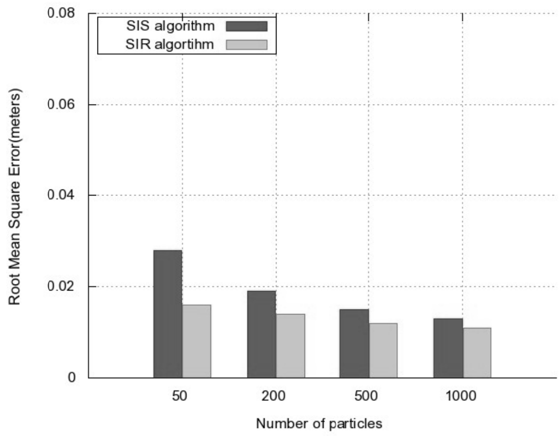

Figure 5.15 Root mean square error comparison between presented algorithms for different number of particles. The simulation distance was 3000m and the PDM was initialized with random values [B]

Table 1 shows a comparison between SIS and SIR algorithms in terms of RMSE and maximum error rates. The experiment was performed ten times and we considered that the car was moving with constant velocity and the PDM was initialized with constant values. Then we calculated the maximum filtering error for different number of particles.

The results have close values for both SIS and SIR approach. As seen in previous experiments, large deviations from the real path are only in the first part of the simulations, where particles do not have enough history of the car. Similar to experiments results, the RMSE analysis gives better results for the resampling algorithm and a large number of particles. We can conclude that even if both filters obviate from the true path in early steps, SIR filter corrects its track faster and converges to a better pose.

Table 1. RMSE and maximum error comparison between SIS and SIR algorithms (using constant velocity and constant PDM values) [B]

Method	SIS			SIR		
Particles	200	1000	3000	200	1000	3000
RMSE	0.036	0.032	0.025	0.018	0.016	0.013
Maximum error	0.058	0.052	0.046	0.051	0.047	0.044

5.2 Scenario for Adaptive implementation

The set of tests for this algorithm is based on driving simulation patterns through a Java traffic simulator. This simulator is able to manage multiple behavior patterns of drivers, such as constant or free movement constant, mild acceleration within a given calibration value combined with a constant motion, including a negative acceleration (deceleration to a limit in order to track a vehicle in traffic), and oscillator behavior within two known values of speed. It was designed to manage multiple distribution of automobiles, such as linear, Gaussian or random. Maximum error monitoring device falls within a radius of 1 m. It is known that, when implemented effectively, particle filters require a cost calculation in proportion to the number of particles therefore we considered this to vary the number of particles, and to see the influence of this factor in determining the value of absolute and relative error. Following these experiments, it was determinate that five hundred particles to ensure a reasonable computational effort.

In order to validate the proposed approach we used a simulation-based system. Simulated sensor values provide noisy information about the monitored vehicle position. Different types of scenarios were considered in order to analyze the algorithm performances.

The sensor error was distributed with the mean at ground truth and added Gaussian noise with a variance of 0.03 meters. At the starting point, the vehicle velocity is set to 50km/h. The particles are randomly spread around the vehicle with a standard deviation of 0.01 meters. At the end of the scenario the associated PDM should reflect the correct behavior of the driver in described conditions. PDM's were initialized with pseudo-random values. Initially the number of particles that describe the filter is 100. The purpose of the adaptive particle filter is to find an optimal filter size as a compromise between accuracy and computational effort. Figure 5.16 illustrates the first scenario. We considered a vehicle moving with 50km/h between the starting and the ending points. In the first steps of the simulation the inferred trajectory is relatively far from the real one. This happened because we assumed that particles are spread randomly around the vehicle's initial position. After the stabilization time, the filter converges close to the real position and maintains a very low error.

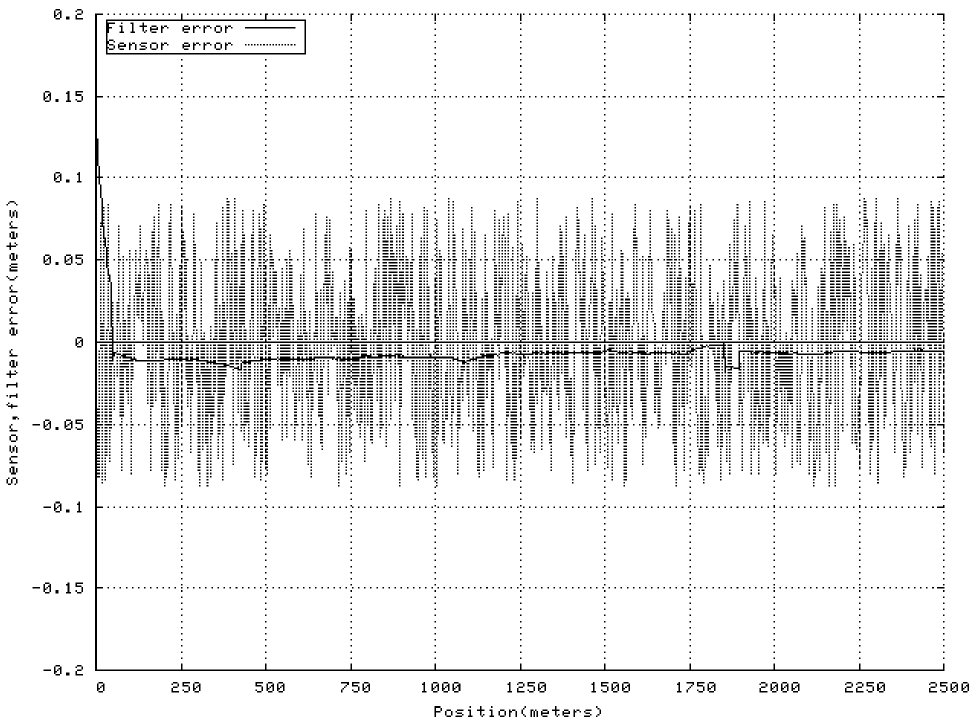

Fig. 5.16. Constant movement, 1000 particles, low sensor error

Figure 5.17 depicts results obtained in those conditions. We considered the vehicle moving on a free road and thus with constant acceleration. The simulation distance was set to 2500 m. The maximum number of particles that describe the filter was set to 1000. Also a threshold T = 0.02 m was considered. When the filter error is above this value, new particles are selected in order to increase the accuracy of the filter. The starting position for each particle was chosen with a zero mean noise and a 0.05 meters standard deviation. Although it presents a slight degradation from other experiments, the proposed method still attempts good performances in terms of mean absolute deviation.

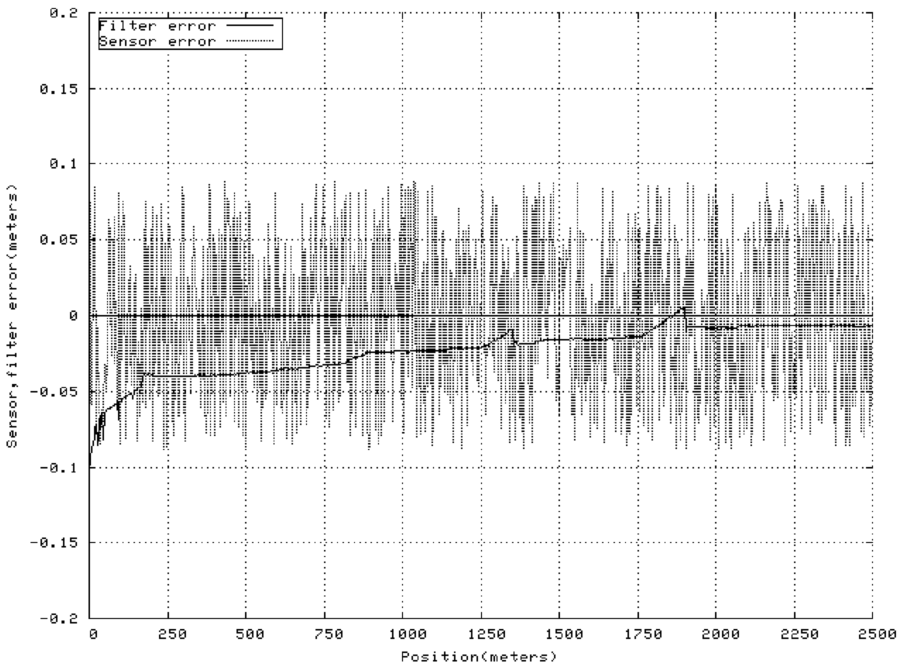

Fig. 5.17. Accelerate behavior, 1000 particles, and low sensor error

As a baseline for comparison with previous experiment, Figure 5.18 illustrates the evolution of the filter error in same conditions excepting the driver's behavior. Vehicle's motion was set to constant velocity. The error is the Euclidean distance of the filter estimation from the car real position. In case of reaching the maximum number of particles, as 1000 for this scenario, particles with low confidence are re-sampled. Therefore we obtain a compromise between the filtering accuracy and computational performance. Indeed, a very high rate of particle resampling implies big computational effort.

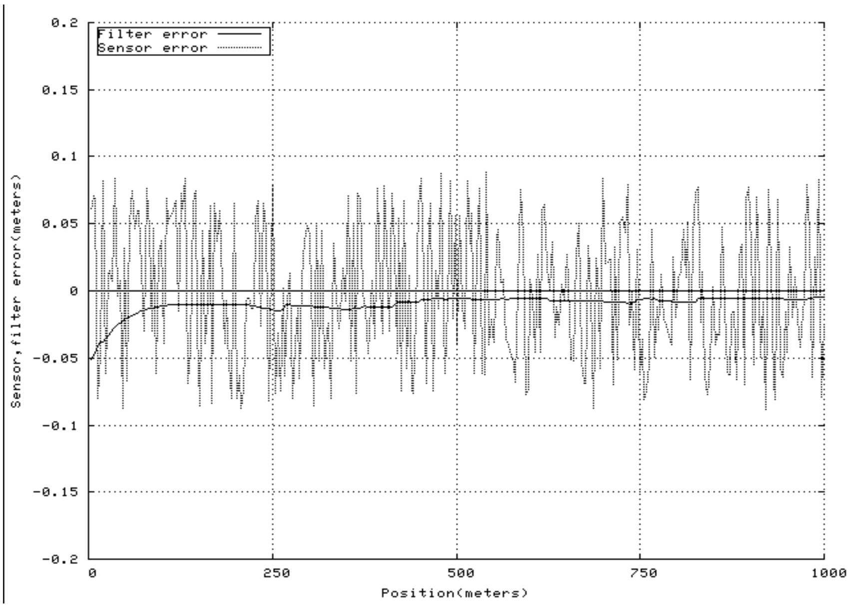

Fig. 5.18. Constant velocity, 1000 particles

Figure 5.19 shows the RMSE filtering error as a function of maximum number of particles allowed and driver's behavior. In order to have a better RMSE accuracy we considered the results from 20 experiments with random values for each type of movement.

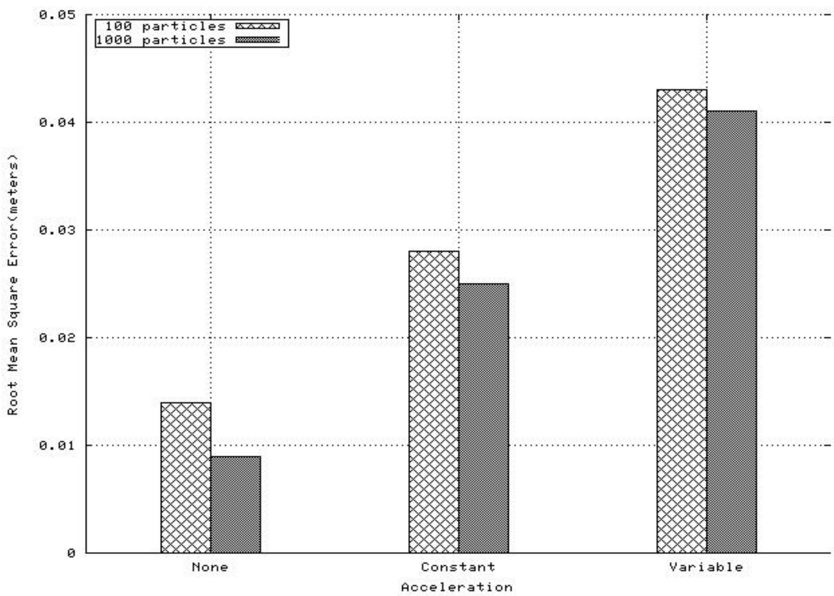

Fig. 5.19. RMSE filtering error

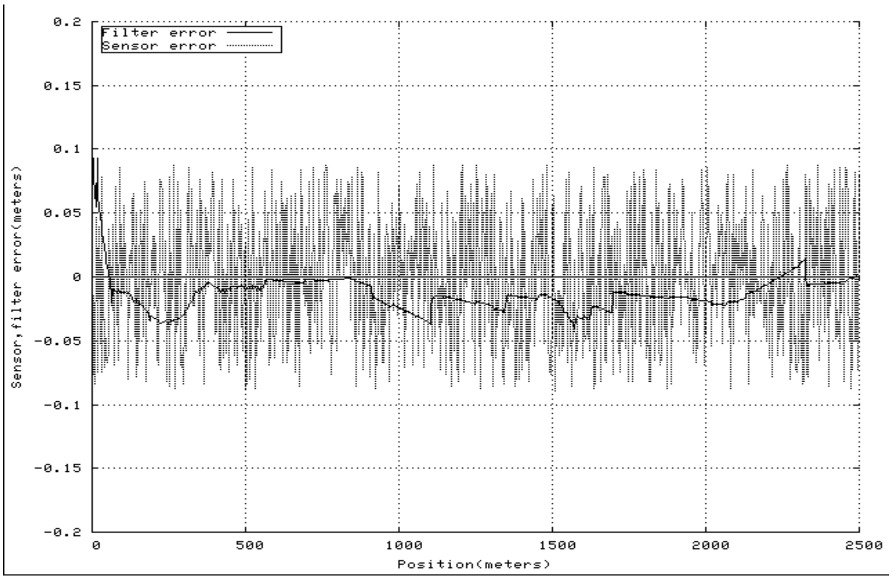

Fig. 5.20. Variable drivers' behavior

Table 2. Samples of the maximum computed error and root mean square error

	No Acceleration		Constant Acceleration		Variable Acceleration	
Number of particles	1000	10000	1000	10000	1000	10000
Maximum Error (meters)	0.13	0.11	0.16	0.13	0.14	0.14
RMSE (meters)	0.014	0.09	0.028	0.025	0.043	0.041

Samples of the maximum computed error and root mean square error are listed in Table 1. The obtained values do not show significant improvement in filter accuracy when the maximum number of samples was increased to 10000. Moreover the computational effort is much higher in that case as the localization speed depends on the number of samples.

The next experiment addresses a different scenario. The type of movement was randomly chosen instead to be fixed. The results show that in this case the filter is far for the performance proved in previous case. However the filter initial distribution is very important and if we consider a lower variance of the particles at the starting point, the filter slowly converges to the real position and the result is maintained on the entire period below sensor's error. Figure 20 depicts this situation.

Conclusions

Several set of tests were performed with different number of particles in order to determine the optimal filter reaction and the resource consumption. These were necessary due to real-time and limited resources requirements. Moreover, in order to obtain an accurate and fast filter response, a considerable number of particles should be taken into account. The compromise in terms of reaction time of the filter and resource consumption was solved using an average number of particles. However, the size of the filter is balanced by resources consumption, like memory or computational intensity. Considering the adaptive approach, in both cases the filter obtains a relative good performance. If it is taken into consideration only the first step, related to a variable number of particles, the benefit comes through the fact that filter error is maintained bellow the value of desired threshold. Still, choosing a very small threshold value will lead to a high rate of regenerated particles, and as result, to a medium performance of the filter.

6 Case Study: using Prediction Algorithms for Overtaking Assistance

In this chapter it is analyzed and presented the possibility of using the above described algorithms in order to design systems for helping a driver to get a proper decision in overtaking action on two-lane national roads. Method relays on input information like observations received from surveillance devices containing the speed or position of each car. Several patterns of overtaking situations were investigated.

One of the most important and risky actions of a driver, is to handle the overtaking. A brief overview of action to overcome identified two situations. Overtaking conditions are slightly different in the case of motorways, compared with the national roads. The first case involves only vehicles running in the same direction on both bands involved, and thus a unidirectional flow of traffic. On the other hand, the roads common situation involves two bands, and thus a two-way traffic flow. A new model to overcome on the road is proposed in [71]. The model considers as important factors during the late reaction of the vehicle for measures acceleration, deceleration, and change lane safely and away from the car ahead. Time to overcome the loss of time and procedure to overcome space-time evolution is numerically investigated using the model mentioned above.

Recent research (e.g. IVSS project, 2007) demonstrates that active safety of vehicles to avoid crash and reducing the damage effect of crash relies on dependable implementation of driving assistance algorithms. Inadequate driver actions, such as overtaking or lane changing with vision obscured by rain, snow, fog, sand, dust, or following improperly another car, combined with drowsy or fatigued driving can lead to fatal car crashes. Another strong root cause is represented by unanticipated action performed by one of the vehicles, and thus leaving insufficient time to react. A good estimation of the remaining distance and also of the velocity of all involved vehicles can improve significantly road safety.

Road and passenger safety remained over the years a major concern for automotive industry and also for researchers. Accident analyses indicate that overtaking on improper conditions can become a dangerous situation for drivers. In a public thesis of crash analyses, Clarke et al. [72] concluded that the majority of UK accidents arose from a decision to start overtaking in unsuitable circumstances. Also they considered that "the problem stems from faulty choices of timing and speed for the overtaking maneuver, not a lack of vehicle control skills as such". Consequently, high dependable driver assistance systems designed to help this maneuver can bring real benefits.

However, investigation on dependability of these kinds of systems is not a trivial problem. As overtaking action relay hardly on human reaction, relevant dependability attributes are hard to be identified. Moreover, there are many other external factors that have to be considered, as road infrastructure or traffic participants. Considering traffic participants we identify also several aspects that have a great impact such as driving skills, emotional factor or age. Overtaking can be considered through the lope of dependable systems as a partially ordered set of actions that needs to be performed and then discharged within a specific causal relationship. The success of one action determines the following ones [73]. Considering this aspect, it is very important to determine the minimal conditions and actions that are need to be fulfilled by the investigated algorithms in order to provide an accurate answer to driver. These aspects will be detailed in this chapter as conditions derived from physical laws applied to instant data received from sensors. Research on advanced driver assistance systems (ADAS) allows reducing the number of accidents resulting from overtaking actions results. This was done via some major projects in past few years. One of them is ROADAS (Research on Overtaking and Advanced Driver Assistance Systems) carried by TU Delft, Nederland. In this project, observing cameras and instrumented vehicles were used in order to gather relevant data on overtaking frequency and overtaking behavior. The system is designed as a warning device. A green light on means safe condition for overtaking, while a red light means that conditions for a proper overtaking are not accomplished [74].

Researcher Geertje Hegeman has developed a warning system which displays a green light when it is safe to perform a maneuver to overcome another vehicle. If action is unsafe, a red light is displayed. She tested this in a driving simulator on a two-lane road. A conclusion is that the assistant driver's leadership gives a sense of ease in making overtaking maneuver and can have a positive effect on road safety and efficiency. Hageman has used in his project of overcoming behavioral observations maneuver on the road N305 between Almere and Zeewolde in the Netherlands. Following these comments concluded that a move to overcome may take on average about eight seconds. Ten percent of cases are less than three seconds between lanes on the retreat and meeting with the opposite vehicle. [75] The study is part of her ROADAS (Research on Overtaking and Advanced Driving Assistant Systems), which is a project of the six subprojects of the research program under Dutch leadership, BAMADAS (Behavioral Modeling Analysis for the Design and Implementation of Advanced Driver Assistance Systems). BAMADAS intended to improve knowledge about the behavior of drivers of road vehicles in interaction with ADAS. For this project that incorporates advanced navigation systems, is developed in collaboration with BMW. ADAS commercial version is expected in the market over the next 10 years.

Other solutions for maintaining an active safety on roads are based on restraint systems. One example is the Intelligent Speed Adaptation (ISA) system [76]. It is based on a device that knows the speed limitation and gives feedback to the driver or limit maximum speed automatically. Technically, it enables an external regulation of the car velocity [77]. Based on how permissive the systems are, we can identify several categories. The first one is called advisory and assumes displaying of the speed limit to line out its changes. Next one is called voluntary, or “driver-select”, which allows the driver to enable or disable control of the vehicle maximum speed. Finally, a mandatory variant means that the vehicle is limited at all times. Another possible classification is based on the speed. Relying on that we identify the following categories: fixed speed limits; variable, where the speed limits are current spatially; and dynamical, where speed limits are expressed in terms of time. Usage of such system has impact on traffic congestion, speed distribution and thus is very likely to have a positive impact also on safety. Liu and Tate propose in [78] an intelligent adaptation speed system that uses in-vehicle electronic devices to enable the speed of vehicle to be regulated automatically, with impact on slow moving queues and higher emission of pollutants gases.

Another important factor in designing an advance driver assistance system is time to collision (TTC), one main indicator of traffic flow. Because this directly depends on distance and velocity of involved cars, the result can be quite accurate. The driver or ADAS must simultaneously estimate the time to collision with an oncoming car and monitor it to avoid a rear-end collision. Then it have to estimate the time required to complete overtake based on the current speed, road conditions, and knowledge on the capabilities of own vehicle. Some other devices [79], will display a warning when the TTC is between 10 seconds and the human reaction time - about one second. Moreover, when TTC estimation is almost one second, it switches from the warning to active driving assistance. Based on this coefficient the Forward Collision Warning (FCW) [80] system is designed to reduce rear-end vehicle collisions.

Another aspect that needs to be considered is the road itself. We can distinguish between two different situations represented by highways and national road. First case implies only interacting vehicles running in the same direction, both on current and overtake lanes. On the other side, on national roads a common situation implies multi-lane (usually two-lane) bidirectional traffic flow.

In the category of active safety, one that is gaining more and more importance is represented by car-to-car communication systems. Besides the ability of wireless communication between cars, these systems provide also communication with road infrastructure. Some technical aspects and problems of ad-hoc wireless car-to-car and also car-to-road infrastructure are presented in [69]. Due to its potential to enhance the safety and also passengers' comfort, many solutions and protocols are proposed also by automotive industry or by researchers [81], [82]. Reference [81] presents concepts and prerequisites for car-to-car communication defined by a team belonging to BMW Corporation. Reference [82] describes a possible cooperative solution based on car communication for collision avoidance systems.

Prediction of human reactions from sensory observations is an extremely challenging task. The complexity resides in the interaction between two main behavioral levels as individual behavior and group dynamics. Overtaking action directly derives from emerging interactions between the individuals. A decision and a success of an action can determine the following activity. For example, slowing down of the leading vehicle can determine next vehicle to slow down or to overtake. MCMC methods have been intensively studied over the last years due to their ability of estimating samples from basically any distribution. Also, a wide variation of this algorithm is known so far, such as Metropolis-Hastings Random Walk, resampling (RMCMC), and Reversible Jump (RJMCMC). As best fitted for overtaking application it was chosen the Metropolis Hastings Random Walk. In this approach a random walk is performed through the configuration space of interest based on a probability distribution. At each point on the walk a random trial move from the current position in configuration space is selected. This trial move is then either accepted or rejected according to a simple probabilistic rule. In our implementation we have used a rejection mechanism, with a local-proposal, in our case a probability density matrix. We let the newly proposed X depend on the previous state of the chain X(t-1). The samples (X(0),X(1),…) are derived from a Markov chain and represent the state of the system at a given moment. The classical Metropolis-Hastings Random Walk algorithm [83] is briefly summarized in figure 6.1.

Starting with $X^{(0)} := (X^{(0)}_1, \ldots X^{(0)}p)$, iterate for $t = 1, 2, ..$

1. Draw $X \sim q(\cdot \mid X^{(t-1)})$.

2. Compute $\alpha(X \mid X^{(t-1)} = \{1, \dfrac{f(x) \bullet q(X^{(t-1)} \mid X)}{f(X^{(t-1)}) \bullet q(X \mid X^{(t-1)})}\}$

Fig. 6.1 The Metropolis-Hastings Random Walk algorithm

Using the MCMC algorithm it was determined whether an overtake action can be performed in safe conditions or not. This action depends directly on several parameters and conditions. One of the most important issues is the clearance of the road. If the road is clear on opposite direction, the major condition for a successful action is to have a greater speed than overtaken car. Another scenario implies a fix obstacle or a car coming from opposite direction. In this it was considered the distance to the car and also the velocity of all three cars.

After having several observations on all involved participants we can determine a movement prediction matrix for each car. This is based on all possible transitions from any considered states noted with "++", "+", "0", "-", "--", and which code a particular coefficient of acceleration. These states and transitions between them, together with constructions of probability density matrix are presented in details in reference [31]. We use an adaptation of the algorithm described in figure 1. Considering the physical characteristics of an average car, the upper limit for "--" was set to -0.34 m/s2. An acceleration up to -0.14 m/s2 was coded with "-". The state "0" designates a constant speed. A smooth acceleration up to 1.12 m/s2 was coded by "+". And finally, a strong acceleration "++" was considered any value over 2.4 m/s2. In any moment the system read from sensors the position and also the velocity of cars. Considering the most probable transition and having the last position we can determine the most likely next location of a car. Having this information is easy to determine from physical laws if a collision can be avoided or not. To achieve this we have to consider also the error rate from sensor. In general the sensor manufacturer provide it as a given range (e.g. +/-0.3%). Based on the observation we can estimate that every resulting difference less than 10m can lead to a very likely collision. Between 10-20 m we have a possible collision. Between 20-30 m is only small chance for collision. Finally, every distance greater that 30 m can be considered a premise of a safe overtakes.

If the resulting difference is negative we have 100% chances for an impact. Starting from this example we need to determine all values based on the actual velocity of the car. We determine this probable velocity (v_{prob}) based on the last read velocity from sensor and from related transition from matrix. Considering also the acceleration value resulting from transition matrix we can determine using Galilee's' formula the distance corresponding for the time step considered for algorithm. This is determined for all three cars involved in this scenario. First condition is to have a positive result for difference between overtaking and overtaken cars. After this condition is fulfilled we need to determine if the difference between the estimated position of the overtaking car and the position of the car from opposite lane gives a positive result or not.

We are dealing with two major sets of tests in order to study the robustness of proposed system. The first one considers each case without considering the position or speedup. The second set of tests is based on the first case presented in the previous section. It considers the scenario of continuous speeding-up until either the system response is affirmative or the vehicle reaches top speed of 130 km/h. This speed is set as a maximum speed both in terms of safety and in terms the legal regulations.

The following two figures present the response of MCMC predictive algorithm. MCMC based algorithm proves to have a small inertia in case of sudden change of state.

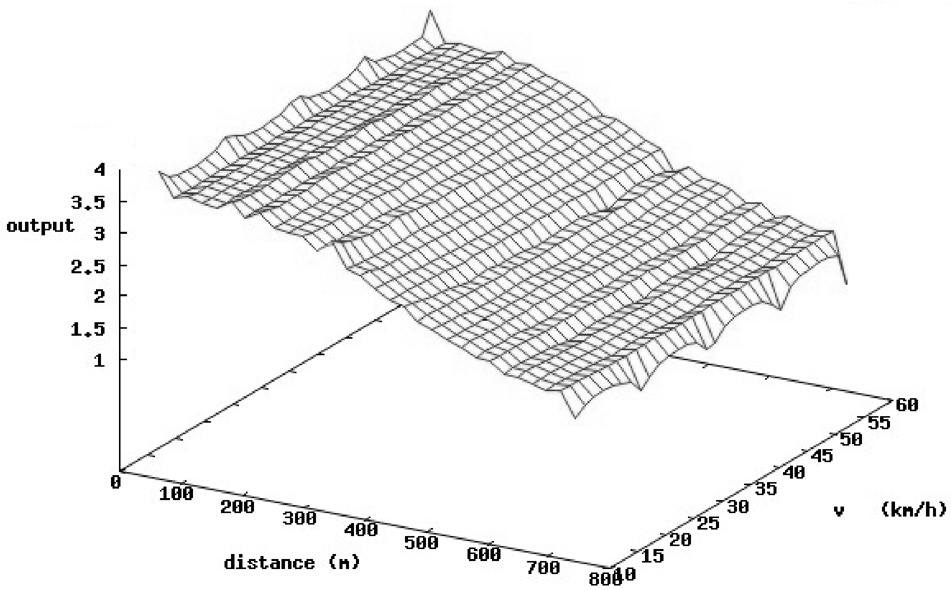

Fig 6.2 MCMC based algorithm response when no speedup was considered [C]

For example in case of continuous acceleration to receive a positive response, the most probable state is estimated at the beginning as the state that was predominant in the past.

The dependability of such systems can be expressed through a coefficient for the risk of overtaking maneuver. We calculate it through the expression $R_{dep} = C \times P$, where C is the consequence of a crash and P is the probability of a failing action (100% – system output) [84]. We can observe in our experiments that R_{dep} risk coefficient is lower in case of a speeding up, and thus is more dependable. On the other side, MCMC method considers also the history of drivers' action and gives more accurate answer in case of no speed up. In [C] a comparison between Fuzzy Logic Expert system for overtaking and MCMC predictive algorithm can be consulted.

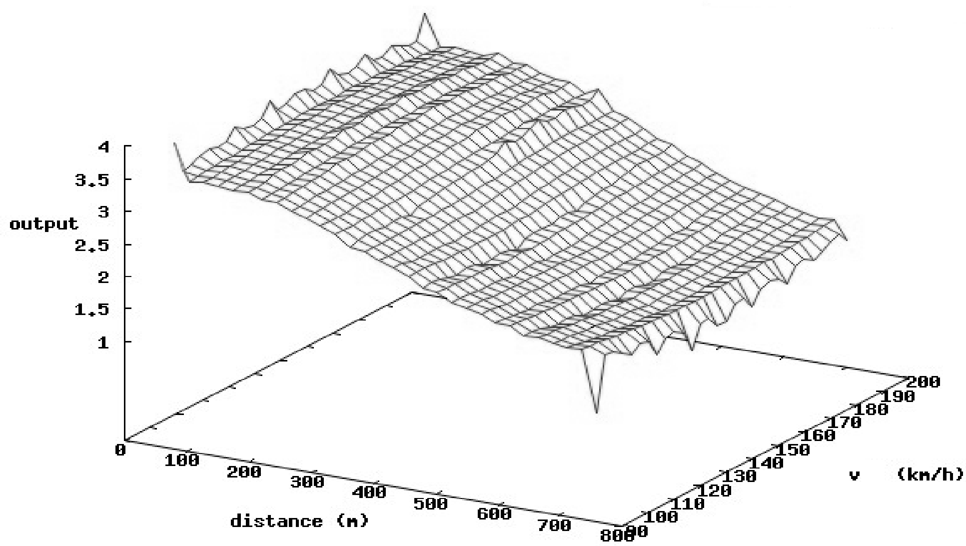

Fig 6.3 MCMC algorithm response when speedup was considered [C]

7 Conclusion and future work

7.1 Concluding remarks

This thesis is concentrated on algorithms for data filtering in traffic systems.. It contains studies on the opportunity of using Particle Filters, and especially MCMC methods in the context of dynamical conditions and restrictive resources. It also briefly presents the available state-of-the-art in this field. Based on this, it was determined that dynamic and adaptive model of observation based on vehicle acceleration could be a promising approach. Possible problems introduced by the presence of continuous probability space are solved by discrete based on driver behavior observation in order to make more efficient resource consumption. This approach was used for development of an adaptive version of the algorithm. The two steps of adaptation are useful in obtaining superior results of filter.

Several set of tests were performed with different number of particles in order to determine the optimal filter reaction and the resource consumption. These were necessary due to real-time and limited resources requirements. Moreover, in order to obtain an accurate and fast filter response, a considerable number of particles should be taken into account. The compromise in terms of reaction time of the filter and resource consumption was solved using an average number of particles. However, the size of the filter is balanced by resources consumption, like memory or computational intensity. Data intensive systems like a video traffic surveillance platform capable of 30 frames/second [70], or in case of night mode with 12 frames/second [85], imply higher error rate and thus require more particles to improve the error corrections. However, considering results from graphs presented above, we can conclude that the proposed solution is competitive in terms of real-time systems like WSN. In that case a special attention should be paid also to intra-network information exchange. This information is needed to keep the history of the car together with other information about PDMs implemented at the level of different network nodes.

Moreover, we can observe that the worst case represented by variable acceleration test sets requires, as expected, a higher number of samples to obtain a good result. Considering exploration of methods like reversible jumps and resampling [40] in the second part we have implement and performed a set of experiments for comparison. The vehicle state vector has been estimated by considering noisy measurements coming from sensors. A first conclusion drawn from the experiments we made is that both algorithms provide reliable estimation of the vehicle’s position. Simulation results show that the SIR particle filter outperforms the SIS algorithm in terms of root mean square error. It was also inspected and discussed how tracking accuracy is affected by choosing of parameters in the particle filter. It has been observed that both algorithm SIS and SIR particle filter depend on their initialization and succeed better as the number of particles increases.

Considering the adaptive approach, in both cases the filter obtains a relative good performance. If it is taken into consideration only the first step, related to a variable number of particles, the benefit comes through the fact that filter error is maintained bellow the value of desired threshold. Still, choosing a very small threshold value will lead to a high rate of regenerated particles, and as result, to a medium performance of the filter.

This thesis presents a study regarding parameters optimization for data filtering algorithms based on particle filters. It was investigated the opportunity of using MCMC methods in the context of dynamical conditions and restrictive resources. Several tests were performed considering different filter size in order to determine the optimal filter reaction and also the influence of several parameters on the adaptive algorithm. As revealed by experiments, a considerable number of particles should be used to obtain an accurate and fast filter response. However, in all the cases the filter error was significantly maintained under the sensors error.

The investigated solution behaves satisfactorily, in case of a known pattern, due to the Markov chain incorporated. If the initial conditions deviate far from the actual and the system is not accordingly to a well known model, the probability of particles approaching the true state can be very small in the early stages. Therefore, the filter may converge slowly in this case. However, the MCMC method is proven to generate samples from almost any distribution.

7.2 Contributions

According to the defined objectives, the author has studied the opportunity of using MCMC PF based on driver behavior in the scope of filtering errors, but also in sense of short term prediction.

Contributions were brought in determination the appropriate variation of PF algorithm, in combination with driving behavior. The author has studied and synthesized the state-of-the art work in the field of prediction algorithms, filtering and error correction algorithms (here including PF, MCMC methods, Kalman filters and so on), traffic flow model and also modeling of driving behavior.

Main contribution was introduced by proposing and implementing an adaptive MCMC algorithm type PF able to estimate continuing value (acceleration of a car), with a discrete state space (PDM), and reducing resource consumption. It was also studied the applicability of the proposed algorithm assistance system such as for overtaking actions. The results were compared with a conventional fuzzy implementation. Simulations have proved significant improvements when implementing predictive.

This thesis presents a study on optimization of parameters for data filtering algorithms based on particle filters. It was investigated whether PF MCMC methods used in the context of dynamic conditions and with restrictive available resources can perform with satisfying results. Several tests were performed considering different filter sizes in order to determine optimal filter reaction and also the influence of several parameters on the adaptive algorithm. As shown by the experiments, a considerable number of particles should be used to obtain an accurate and fast response of the filter. However, in each case presented, the filter error was significantly retained under the value sensor errors.

The proposed algorithm model assumes a continuous motion, based on a matrix that summarizes the finite state space (mesh), and return to the continuously distribution, all this in the presence of observations affected by noise. Based on these assumptions, the filter is forced to perform well in the presence of three possible sources of error: error sensors, sampling and adaptation in discrete time acceleration. Another aspect that was considered was the rate of convergence.

Contributions are made in the implementation of an adaptive algorithm based on multi scale Markov Chain Monte Carlo model that includes a longitudinal observation, intelligent driver model (IDM).

Adaptability of the algorithm consists of two orthogonal aspects: the first of them is represented by the number of particles. They are determined through a calibration value. A minimum number of particles needed for convergence of the filter is set in the initial phase of the algorithm, and also convenient for the error threshold filter. In light of these issues more particles can be introduced for a better approximation performance of the filter, and when it reaches the maximum number of particles, re-sampling is performed for particles with low weight. With these values, it is easy to determine the ratio of computation effort and accuracy.

The second aspect concerns the adaptability value of acceleration. For this integration model has been considered a model of microscopic observation, IDM. Acceleration value was adjusted to better reflect reality. Thus, the values were discretized, and used for finite state space in order to reduce computation effort and then were again adapted to continuous variation. In the early stages of the implementation of the algorithm, using a fixed value for acceleration for each state was taken into account, but this resulted in a non-optimal filter performance for the limit states entering values deviations from true values. By integrating IDM in the model of observation, each particle could "follow" the observation, by adapting acceleration value. Given the fact that each particle has the objective to better predict the position of observation, based on weighted mediation is performed and thus get a better filter performance.

7.3 Future research directions

Future work will be focused of development of a distributed version of proposed algorithm and optimization in order to fulfill low resources consumption request from wireless sensors.

It will be considered also the opportunity of adapting the algorithm in order to run it on PSoC sensors available in the SENT Laboratory. Also it will be studied the opportunity of using smart agents in context of distributed algorithms. Also theoretical considerations regarding strong and weak convergence criteria will be study for the proposed model.

It will be studied, from the mathematical point of view, the dependency between input parameters such as: sensor error, number of particle, initialization of PDM and distribution of particles and the settling time of the filter. Also another fact that will be taken into consideration it will be the determination of the function which gives the time point where the filter error goes under a set values and remains under that calibration value.

7.4 Published papers

[A] **"Adaptive Filtering Algorithm Based on Drivers' Behavior",** Florica Naghiu, Dan Pescaru, Ionel Jian, *Proc. IEEE ICINC 2010* , Kuala Lumpur, Malaysia 26-28nov 2010, ISBN 978-1-4244-8270-2 , IEEE Catalog Number: CFP1076K-PRT, pp 174-178

[B] **"Parameters Optimization for an Adaptive MCMC Algorithm based on Drivers Behavior",** Florica Naghiu, Dan Pescaru, Victor Gavrila, Ionel Jian, *Workshop Craiova*, 24-25feb 2011

[C] **"Corrections of Sensing Error in Video-based Traffic Surveillance",** Naghiu F., Pescaru D., Magureanu G., Jian I., Doboli A. , *Proc. 5th International Symposium on Applied Computational Intelligence and Informatics,* SACI 2009 , Timisoara, Romania, ISBN 978-1-4244-4478-6, May 2009, pp.217- 224.

[D] "**Scalable Metric for Coverage Evaluation in Video based Wireless Sensor Networks" ,** Pescaru Dan, Istin Codruta, Naghiu Florica, Gavrilescu Madalin, Curiac Daniel, *Proc. 5th International Symposium on Applied Computational Intelligence and Informatics,* SACI 2009 , Timisoara, Romania, ISBN 978-1-4244-4478-6, May 2009, pp 323-328.

[E] "**Influence of Driver Behavior Patterns in Correcting Video Sensing Errors in Traffic Surveillance Applications"**,Naghiu Florica Maria, Pescaru Dan,*Proc. 5th IEEE International Conference on Intelligent Computer Communication and Processing,* IEEE-ICCP 2009, Cluj-Napoca, Romania, ISBN 978-1-4244-5007-7, Aug. 2009, pp.173-176.

[F] "**Influence of Particle Filter Parameters on Error Correction Accuracy in Traffic Surveillance",** Florica Naghiu , Dan Pescaru, Victor Gavrila, Ionel Jian, Daniel Curiac, P*roc. 14th IEEE International Conference on Intelligent Engineering Systems 2010,*INES *Las Palmas, Spain,May 5-7, 2010,* Print ISBN: 978-1-4244-7650-3 ,Digital Object Identifier: 10.1109/INES.2010.5483851 *pp.183-188*

[G] **Prediction Algorithms for Overtaking Actions**, Florica Naghiu , Dan Pescaru, Ionel Jian „*Monographs of System Dependability*", DepCoS-RELCOMEX, Wroclaw,Polonia, 29iunie-2iulie 2010, Chapter 6

[H] "**Towards using Particle Filtering for Increasing Error Correction Accuracy in Traffic Surveillance Applications"** , Victor Gavrila, Florica Naghiu and Dan Pescaru, *Proc. IEEE ICCC-CONTI* , Timisoara, 27-29mai 2010. **P**rint ISBN: 978-1-4244-743, Digital Object Identifier: 10.1109/ICCCYB.2010.5491257, pp 321-326

References

[A] "Influence of Particle Filter Parameters on Error Correction Accuracy in Traffic Surveillance", Florica Naghiu, Dan Pescaru, Victor Gavrila, Ionel Jian, Daniel Curiac, IEEE INES 2010, Spain, 5-8 May 2010.

[B] "Towards using Particle Filtering for Increasing Error Correction Accuracy in Traffic Surveillance Applications", Victor Gavrila, Florica Naghiu and Dan Pescaru, IEEE ICCC-CONTI, Romania, 27-29 May 2010.

[C] "Prediction Algorithms for Overtaking Actions", Florica Naghiu, Dan Pescaru, Ionel Jian, Monographs of System Dependability, Chapter 6, DepCoS-RELCOMEX, Poland, ISBN 978-83-7493-528-9, July 2010.

[1] A. Downs, "Why Traffic Congestion Is Here to Stay. . . and Will Get Worse", ACCESS, Nr 25, fall 2004.

[2] http://www.parliament.uk/commons/lib/research/rp98/rp98-016.pdf , 2009.

[1] http://www.motorauthority.com/blog/1023468_audi-starts-trial-of-fuel-saving-travolution-traffic-system , 2009.

[2] http://www.sunatraffic.com.au/index.html , 2009.

[3] Crisan D., Doucet A., "A Survey of Convergence Results on Particle Filtering Methods for Practitioners", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 3, MARCH 2002 736-746

[4] M. Greiffenhagen, D. Comaniciu, H. Niemann, and V. Ramesh, "Design, analysis and engineering of video monitoring systems: An approach and a case study," Proceedings of the IEEE, Vol. 89, No. 10, 2001, pp. 1498-1517.

[5] Sing Yiu Cheung, Sinem Coleri Ergen and Pravin Varaiya, "Traffic Surveillance with Wireless Magnetic Sensors," in Proc. of the 12th ITS World Congress, San Francisco, Nov. 2005.

[6] Prabal Dutta, Mike Grimmer, Anish Arora, Steven Bibyk, and David Culler, "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events," in Proc. of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, UCLA, Los Angeles, California, USA, April 25-27, 2005.

[7] Kaijen Hsiao, Henry de Plinval-Salgues and Jason Miller, "Particle Filters and Their Applications," Cognitive Robotics, April 2005

[8] S. Mukhopadhyay, C. Schurgers, D. Panigrahi, S. Dey, "Model-Based Techniques for Data Reliability in Wireless Sensor Network", IEEE Transactions on Mobile Computing, Vol. 8, No. 4, pp. 528-543, April 2009.

[9] C. Kwork, D. Fox, M. Meila, "Real-time particle filters", in Proceedings of the IEEE, Vol. 92, Issue 3, pp. 469 – 484, March 2004.

[10] M. Rosencrantz, G. Gordon and S. Thrun, "Decentralized Sensor Fusion with Distributed Particle Filters", in Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03), pp. 493-500, 2003.

[11] M. Coates, "Distributed Particle Filters for Sensor Networks", in Proceedings of 3nd workshop on Information Processing in Sensor Networks (IPSN), pp 99-107, 2004.

[12] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli, "On-line Fault Detection of Sensor Measurements", in Proceedings of IEEE Sensors 2003, Vol. 2, pp 974 – 979, 22-24 Oct. 2003.

[13] B. Kanagal, A. Deshpande, "Online Filtering, Smoothing and Probabilistic Modeling of Streaming Data", in Proceedings of the 24th International Conference on Data Engineering (ICDE), 2008.

[14] F. Wood, "Particle Filtering, a brief introductory tutorial", Gatsby, August 2007

[15] R. Boel, M. Mihaylova, "Modelling freeway networks by hybrid stochastic models", In Proceedings of the IEEE Intelligent Vehicle Symposium, Parma, Italy, pp. 182-187, 2004.

[16] A. Hegyi, D. Girimonte, R. Babuska, and B. De Schutter, "A comparison of filter configurations for freeway traffic state estimation", in Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference (ITSC 2006), Toronto, Canada, pp. 1029–1034, September 2006.

[17] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.-J. Nordlund, "Particle filters for positioning, navigation and tracking", IEEE Transactions on Signal Processing, vol. 50, no. 2, February 2002.

[18] P. Torma, C. Szepesvari "LP-N-IPS: An improvement of particle filters by means of a local search", The 5th IFAC Symposium on Non-linear Control Systems, 2001, pp. 715–719

[19] K. Choo, DJ Fleet, "People tracking using hybrid Monte Carlo filtering", IEEE International Conference on Computer Vision, Vancouver, pp. 321–328, 2001

[20] A. Davison, "Real-time simultaneous localization and mapping with a single camera", the IEEE Int. Conference on Computer Vision, ICCV'03, 2003.

[21] JFG de Freitas, "Bayesian methods for neural networks", PhD. Thesis, Department of Engineering, Cambridge University, http://www.cs.ubc.ca/nando/publications.html, 1998.

[22] E. Maggio, A. Cavallaro, "Accurate appearance-based Bayesian tracking for maneuvering targets", Computer Vision and Image Understanding, vol. 113, issue 4, pp 544-555, 2009.

[23] CP Robert, G. Casella, "Monte Carlo statistical methods", Springer Verlag, Berlin, 1999.

[24] C. Andrieu, PM Djuric, and A. Doucet, "Model selection by Markov chain Monte Carlo computations," Signal Process., pp. 19–37, 2001

[25] S. Arunlampalam, S. Maskell, N. Gordon, T. Clapp, "A tutorial on Particle filters for On-line Non-Linear/ Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, 2002.

[26] C. Kwok, D. Fox, M. Meila, "Real-time Particle Filters", Proceedings of the IEEE, vol.92, Mar. 2004

[27] M. Rosencrantz, G. Gordon, S. Thrun, „Decentralized Sensor Fusion with Distributed Particle Filters", Proceedings of the 19th Annual Conference on Uncertainty în Artificial Intelligence (UAI-03), 2003

[28] Guoliang Fan, Vijay Venkataraman, Li Tang, Joseph Havlicek, "A Comparative Study of Boosted and Adaptive Particle Filters for Affine-Invariant Target Detection and Tracking", Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)

[29] Francisco Angel Moreno, Jose Luis Blanco and Javier Gonzalez, "A Probabilistic Observation Model for Stereo Vision Systems: Application to Particle Filter-Based Mapping and Localization" In Proceedings of IbPRIA (1) 2007. pp.346~353

[30] S. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," Statistics and Computing, Vol. 10, No. 3, 2000, pp. 197-208.

[31] M.J. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particlefilters," J Am Stat Association, Vol. 94, No. 446, Jan 1999, pp. 590–599.

[32] S. Arunlampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on Particle filters for On-line Non-Linear/ Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, Vol. 50, No. 2, February 2002, pp 174-188.

[33] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in the IEE Proceedings of Radar and Signal Processing, Vol. 140, No. 2, April 1993, pp. 107–113.

[34] E. Lehmanny and R. Williamson, "Particle Filter Design using Importance Sampling for Acoustic Source Localization and Tracking in Reverberant Environments", EURASIP Journal on Applied Signal Processing, special issue on Advances in Multi-Microphone Speech Processing, Vol. 2006, Article ID 17021, 2006.

[35] R. Merwe, A. Doucet, N. Freitas, E. Wan, "The Unscented Particle Filter", Research thesis CUED/FINFENG/TR 380, Department of Engineering, University of Cambridge, 2000.

[36] A. Soto, "Self Adaptive Particle Filter", in Proc. International Joint Conferences on Artificial Intelligence, Edinburgh, England, 2005.

[37] N. Doucet, N. de Freitas, N. Gordon, ed., "Sequential Monte Carlo in Practice," Springer-Verlag, New York, 2001.

[38] T. Chateau, Y. Goyat, L. Trassoudaine, "M2SIR: A Multi Modal Sequential Importance Resampling Algorithm for Particle Filters," in Proceedings of the 16th IEEE International Conference Image Processing (ICIP'09), Cairo, 2009, pp 4073-4076.

[39] J. Liu and M. West, "Combined Parameter and State Estimation in Simulation-based Filtering," Sequential Monte Carlo Methods in Practice, Chapter 10, Springer, 2001.

[40] P. Resnik, E. Hardisty, "Gibbs Sampling for the Uninitiated", unpublished manuscript, Version 0.3, October 2009.

[41] J. O Ruanaidh and W. J. Fitzgerald. Numerical Bayesian Methods Applied to Signal Processing, chapter 2, pages 8-9. Springer-Verlag, New York, 1996.

[42] C.P. Robert and G. Casella. "Monte Carlo Statistical Methods", (second edition), New York: Springer-Verlag, 2004.

[43] U. Kiencke and L. Nielsen, "Automotive control systems for engine, driveline and vehicle", Springer, Berlin, 2000.

[44] DD Salvucci, ER Boer and A. Liu, "Toward an integrated model of driver behavior în a cognitive architecture", Transportation Research Record 1779, 2001.

[45] John Jairo, Martinez, C. Canudas-de-Wit , "A safe longitudinal control for adaptive cruise control and stop-and-go scenarios", IEEE Transactions on Control systems Technology, vol. 15, pp. 246-258, March 2007.

[46] KI Ahmed, "Modeling drivers' acceleration and lane changing behavior", Ph.D.Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, 1999.

[47] Wang,W., N.Itakura, N.Honda, "Evaluate the driving model through analyzing various driving characters"

[48] Liu A., Salvucci D., "Modeling and prediction of human driver behavior", Proc. of the 9th Intl. Conference on Human-Computer Interaction, Aug, 2001

[49] Gning, A.; Mihaylova, L.; Boel, R.; "An interval compositional vehicular traffic model for real-time applications", IEEE Intelligent Vehicles Symposium, 2008

[50] Fuller R., "Towards a general theory of driver behavior", Accident Analysis and Prevention 37 (2005) 461–472

[51]Barceló J., Delgado M., Funes G., D. García, J. Perarnau, and A. Torday„Online Microscopic Traffic Simulation Supports Real-time Traffic-management Strategies", SIAM News, Volume 40, Number 9, November 2007

[52] A. Messmer and M. Papageorgiou, "METANET: A macroscopic simulation program for motorway networks", Traffic Engineering and Control, nr 31, 466-470, 1990

[53]Burghout W., "Mesoscopic Simulation Models for Short-Term Prediction" Centre for. Traffic Research, Royal Institute of Technology, Stockholm 2005

[54] Piccoli B., Tosin A., "A Review of continuum mathematical models of vehicular traffic" in Collection of Encyclopedia of Complexity and Systems Science, 2009: 9727-9749

[55]S. Mukhopadhyay, C. Schurgers, D. Panigrahi, S. Dey , "Model-Based Techniques for Data Reliability in Wireless Sensor Network", IEEE Transactions on Mobile Computing, Vol. 8, No. 4, pp. 528-543, April 2009.

[56] K.T. Waldeer, "Numerical investigations of a mesoscopic vehicular traffic flow model based on a stochastic acceleration process " in Proc. Europ. Conf. on Computational Physics (CCP 2001), vol. 147 of Computer Physics Communications, p. 650. Elsevier, Amsterdam, 2002.

[57] K. T. Waldeer, "Kinetic Theory in Vehicular Traffic Flow Modeling", 25th International Symposium on Rarefied Gas Dynamics,Saint-Petersburg, Russia, 2006

[58] Serge P. Hoogendoorn, Piet H.L. Bovy, "State-of-the-art of Vehicular Traffic Flow Modeling"

[59] A. Papavasiloiu, "Uniformly Convergent Adaptive Particle Filter," Journal of Appl. Probab., Vol. 42, No. 4, pp. 1053-1068, 2005.

[60]An, P. E., Harris, C. J., "An intelligent driver warning system for vehicle collision avoidance," IEEE Trans. System, Man and Cybernetics vol. 26 (2), 1996, pp. 254-261.

[61]Sing Yiu Cheung, Sinem Coleri Ergen and Pravin Varaiya, "Traffic Surveillance with Wireless Magnetic Sensors," in Proc. of the 12th ITS World Congress, San Francisco, Nov. 2005.

[62] Pitt M. K., Shephard N., "Filtering via Simulation: Auxiliary Particle Filters," Journal Of American Statistical Association, Theory and Methods, Vol. 96, no. 446, June 1999.

[63] P. Giordani, R. Kohn, "Adaptive Independent Metropolis-Hastings by Fast Estimation of Mixtures of Normals," in press.

[64] Ping Tan, Zixing Cai. "An adaptive particle filter based on posterior distribution‟, July 2010, WICICA, PP 5886 – 5890.

[65] An, P. E., Harris, C. J., "An intelligent driver warning system for vehicle collision avoidance," IEEE Trans. System, Man and Cybernetics vol. 26 (2), 1996, pp. 254-261.

[66] T. Martin, A. Kesting, and D. Helbing, "Delays, inaccuracies and anticipation in microscopic traffic models", Physica, vol. 360, 2006, pp. 71–88.

[67] M. Treiber, "Longitudinal Traffic model: The IDM", http://www.vwi.tu-dresden.de/ ~treiber/MicroApplet/IDM.html, 2010.

[68] Kaijen Hsiao, Henry de Plinval-Salgues and Jason Miller, "Particle Filters and Their Applications," Cognitive Robotics, April 2005.

[69]Straka O., Simandl M., "Particle Filter with Adaptive Sample Size", KYBERNETIKA — VOLUME 47 (2011) pp 385–400.

[70] R. Kostia, "Night-Time Traffic Surveillance: A Robust Framework for Multi-vehicle Detection, Classification and Tracking", in Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp 1-6, 2009.

[71] Tanga TQ, Huangb H., Wongc S. and Xu X., "A new overtaking model and numerical tests", Statistical Mechanics and its Applications, Elsevier, vol. 376, pp. 649-657, March 2007.

[72] Desrosiers D., Birdsong C., Schuster P., A Pre-Crash Simulator to Evaluate Vehicle Collision Prediction Algorithms, the 5th IFAC Symposium on Advances in Automatic Control, 2007.

[73] Miček J., Kapitulik J., CAR-to-CAR Communication System, the International Multiconference on Computer Science and Information Technology, Poland, 2009, pp. 627-631.

[74]http://www.prevent-ip.org/en/about_preventive_safety/introduction/ time_to_collision_02.htm, 2009.

[75] Deflt University of Technology (2008, February 27). Overtaking Assistant Could Help Prevent Many Traffic-related Deaths. ScienceDaily. Retrieved August 26, 2008 from http://www.sciencedaily.com/releases/2008/02/080226092749.htm

[76] Clarke D., Ward P.J., Jones J., Overtaking Road-Accidents: differences in maneuver as a function of driver age, Accident Anal. & Prev. Journal, Elsevier, Vol. 30, 1998, pp. 455-467.

[77] Mazzara M., Deriving Specifications of Dependable Systems: toward a Method, 12th European Workshop on Dependable Computing, EWDC, 2009.

[78] Biswas S., Tatchikou R., Dion F., Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety, IEEE Communications Magazine, 2006, pp.74 – 82.

[79] Jamson A.H., Lai F.C.H., Carsten O.M.J, Potential Benefits of an Adaptive Forward Collision Warning System, Transportation Research, Elsevier, Vol. 16, 2008, pp.471-484.

[80] Kosch T., Technical concept and prerequisites for car-to-car communication, BMW Group Research and Technology, Munich, 2006.

[81] Liu A., Salvucci D., Modeling and Prediction of Human Driver Behavior, the 9th Intl. Conference on Human-Computer Interaction, 2001.

[82] C Arsten O., Tate F., "Intelligent Speed Adaptation: The Best Collision Avoidance System?", the 17th International Technical Conference on the Enhanced Safety of Vehicles, Netherlands, 2001.

[83] Roberts G., "General state space Markov chains and MCMC algorithms", Probability Surveys, Vol.1, pp. 20-71, 2004.

[84] Webster N., Kuwahara M., Chung E., "Tactical Driver Lane Changing Heuristic Using Forward Search", Proceedings of Infrastructure Planning, Vol. 33, Japan, 2005.

[85] S.M. Bhandarkar, L. Xingzhi, "Fast and Robust Background Updating for Real-time Traffic Surveillance and Monitoring", in IEEE Computer Vision and Pattern Recognition CVPR Workshops, pp. 55 – 55, June 2005.