December 2019

Probabilistic Framework for Behavior Characterization of Traffic Participants Enabling Long Term Prediction

Jasprit Singh Gill
Clemson University, jasprig@g.clemson.edu

PROBABILISTIC FRAMEWORK FOR BEHAVIOR CHARACTERIZATION OF TRAFFIC PARTICIPANTS ENABLING LONG TERM PREDICTION

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Automotive Engineering

by
Jasprit Singh Gill
December 2019

Accepted by:
Dr. Pierluigi Pisu, Committee Chair
Dr. Matthias J. Schmid, Co-Chair
Dr. Venkat N. Krovi
Dr. Yunyi Jia

Abstract

This research aims at developing new methods that predict the behaviors of the human driven traffic participants to enable safe operation of autonomous vehicles in complex traffic environments. Autonomous vehicles are expected to operate amongst human driven conventional vehicles in the traffic at least for the next few decades. For safe navigation they will need to infer the intents as well as the behaviors of the human traffic participants using extrinsically observable information, so that their trajectories can be predicted for a time horizon long enough to do a predictive risk analysis and gracefully avert any risky situation. This research approaches above challenge by recognizing that any maneuver performed by a human driver can be divided into four stages that depend on the surrounding context: intent determination, maneuver preparation, gap acceptance and maneuver execution. It builds on the hypothesis that for a given driver, the behavior not only spans across these four maneuver stages, but across multiple maneuvers. As a result, identifying the driver behavior in any of these stages can help characterize the nature of all the subsequent maneuvers that the driver is likely to perform, thus resulting in a more accurate prediction for a longer time horizon. To enable this, a novel probabilistic framework is proposed that couples the different maneuver stages of the observed traffic participant together and associates them to a driving style. To realize this framework two candidate Multiple Model Adaptive Estimation approaches were compared: Autonomous Multiple Model (AMM) and Interacting Multiple Model (IMM) filtering approach. The IMM approach proved superior to the AMM approach and was eventually validated using a trajectory extracted from a real world dataset for efficacy. The proposed framework was then implemented by extending the validated IMM approach with contextual information of the observed traffic participant. The classification of the driving style of the traffic participant (behavior characterization) was then demonstrated for two use case scenarios. The proposed contextual IMM (CIMM) framework also showed improvements in the performance of the behavior classification of the traffic participants compared to the IMM for the identified use case scenarios. This outcome warrants further exploration of this framework for different traffic scenarios. Further, it contributes towards the ongoing endeavors for safe deployment of autonomous vehicles on public roads.

Dedicated To My Mom, Dad and Bro

Acknowledgments

In this long journey to my doctoral degree, I am grateful to have found so many mentors, friends and families without whom my experience would not have been the same. A page is too less to merely name all the people who have influenced me in some way during this period. Nonetheless, this dissertation would not be complete without mentioning some people from the following groups.

Academic advisors: My sincerest gratitude goes to my advisor, Dr. Pierluigi Pisu, for being patient enough to let me explore different areas of the technology and allowing me to pursue the one that piqued my interest. My appreciation to my co-advisor, Dr. Matthias Schmid, is beyond words. He mentored me through the crucial phases of my doctoral research and willingly flexed his schedule in the times I needed guidance. Our lengthy and invaluable brain storming sessions in front of a white board gave me a glimpse of the things we could have achieved had we met a couple of years earlier. I am thankful for my committee members: Dr. Venkat Krovi, for supporting me at different stages of my doctoral research as well as providing me unrestricted access to the state-of-the-art robotics equipment from his lab; and Dr. Yunyi Jia, for his ever wise, insightful suggestions for this work.

I am also grateful to my advisors during the initial phases of graduate studies: Dr. Joachim Taiber, for supporting me, allowing me to explore my entrepreneurial side and coaching me into visualizing the big picture in things; Dr. David Smith for providing me with critical guidance I required during my PhD qualifiers; and Dr. Todd Hubing for guiding me in the formative years of my PhD as well as providing access to his priceless Clemson Vehicular Electronics Laboratory.

Funding agencies: Thanks are also owed to National Science Foundation (grant No. CNS-1544910), for the financial support for this work, and Clemson University Libraries, for providing me access to a treasure of literature without which my journey would have been unimaginable.

Friends and family: I have cherished the philosophical as well as entrepreneurial discussions with Prashant Salla over endless cups of *chai* and am indebted to him for jolting me out in my crucial moments of doubt by reassuring my faith in my decisions. Simone Gelmini, Marco Rossi, Roberto Merco, Andrea Gil and Adhiti Raman, the numerous get-togethers, excursions and culinary adventures we took together sharing our ups and downs will never be forgotten. Ashwini Amin, who brought colors and cheerfulness into my otherwise grayscale schedule. Angelina and Celine, my travel companions, who helped me retain my sanity through this roller-coaster journey.

I was able to survive through my time away from my family only because I found mine in Greenville. I am grateful to the Griffith, the O'Donnell and the Patrick families who opened their doors to me for their festivities and the familial moments of celebrations.

I truly admire the patience and flexibility with which Elena Gazzola has bridged the gaps for meeting my constraints during this period. Her tremendous love, support and sense of adventure have always amazed me with things I would not have known about myself otherwise.

Last but not the least, my deepest gratitude goes towards my parents, Gunwant Kaur Gill and Narendra Pal Singh Gill, as well as my brother, Harpreet Singh Gill, who have always stood by me emotionally and financially, in my moments of doubts, through my eccentric desires and kept encouraging me in achieving my goals. I am what I am because of them, and wouldn't have reached this far without their support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Advances in the field of artificial intelligence, sensing and computing in the past 30 years have opened up a lot of opportunities for the applications for autonomous vehicles. Mobile robots, which are a category of autonomous vehicles, have found many uses in factories, military reconnaissance ground vehicles, warehouses and even consumer products like robotic vacuum cleaners as well as lawn mowers [52]. Availability of excellent texts and tutorials for robotics [65, 23, 45] coupled with open source platforms like Robot Operating System (ROS)[55] have further propelled this revolution. Open communities for sharing software for autonomous vehicles have a number of reference implementations of the Simultaneous Localization and Mapping (SLAM), perception and navigation techniques made available over the past decade, some even transitioning into industrial grade off the shelf products [70, 6, 9, 2]. Due to operation in exclusive zones or in controlled conditions, the mobile industrial robots have progressed in the phase where they can function well towards their assigned goals but they need to be robust in performance [62, 21].

Another industry that is gearing up to adopt the benefits of the above developments is automotive industry. Due to a significant increase in the road fatalities in the past few decades, human safety has become one of the two key objectives of transportation, the other being fuel efficiency. Autonomous vehicle technology is seen as promising for both. Major interest in it arose after Defense Advanced Research Projects Agency (DARPA) organized two DARPA Grand Challenges and the DARPA Urban Challenge (DUC) for autonomous vehicles from 2004-2007 [3, 4, 5, 7]. Although these competitions uncovered significant challenges that needed to be addressed for getting the autonomous vehicles on public roads, the public perception towards the robotic systems as a part of automobiles was shifted. Since 2010, when Google announced it has been developing and testing a fleet of vehicles for highly autonomous driving, various other auto makers have gradually started working towards it. In 2014, Society of Automotive Engineers (SAE) released an information report J3016 [58] in which it defines 6 different levels of autonomy, with level 2 as partially automated and level 3 and above as highly automated. There have been two school of approaches for developing this technology. The newer players like Waymo (Google spinoff), Uber, Cruise automation (GM undertaking) have been working on fully automated driving (SAE Level 4+) whereas the traditional automakers have been working on incremental automation approaches. Many automakers already have an SAE Level 2 systems in the market. Some like Tesla and Mercedes Benz are working towards SAE Level 3 autonomy in their vehicles. Various independent studies have determined that by 2030, a significant portion of the vehicles being sold will have some level of autonomy [8].

1.1 Motivation

Deployment of highly autonomous vehicles on the road has significant challenges however. Unlike mobile industrial robots, which have the liberty to operate in controlled environmental conditions and designated operating areas, autonomous vehicles will have to work under changing weather and light conditions, all the while sharing the roads with other manually driven vehicles. While the changing environmental conditions pose a significant challenge in perception of the surroundings, sharing the roads with the other traffic participants pose challenges that span across perception and planning modules of the cars. Unlike industrial environments where the robots and the other human or non-human participants work towards a common overarching goal, the participants in the traffic have all independent goals. This requires the participants to interact with each other and often negotiate their intent or right of the way to successfully navigate through the traffic. MIT - Cornell collision [32] in the DUC, which occurred as a result of a failure to anticipate the intent of the other car, was one of the first instances where this challenge was uncovered. Now, almost a decade after the learnings from that incident in DUC, the obvious question that comes to ones mind is where does the technology stand?

A good way to answer this question is to go through the Disengagement Reports from the Autonomous Vehicle tests in the California Department of Motor Vehicles (CADMV) website [1]. The CADMV issues Autonomous Vehicle Testing Permits to the manufacturers developing and testing autonomous vehicles on the public roads and as per the regulations the permit holders are required to submit a disengagement report annually summarizing the cause of every disengagement during their tests. As of the date of writing of this publication, there are 62 permit holders with this program. A quick glance through their disengagement reports tells the challenges the autonomous vehicle developers are facing. The ego vehicle giving insufficient or incorrect yields, exhibiting unwanted behaviors during a maneuver, or other traffic participant driving recklessly make up a significant portion of reasons for the disengagements (Table 1.1). This justifies the need for further understanding the various interactions between the traffic participants.

Table 1.1: Automakers facing behavior related issues as per disengagement reports

Automaker	Behavior related	Total	Percentage
GM Cruise	46	82	56%
Zoox	6	8	75%
Baidu	57	88	65%
Honda	30	77	39%
Nuro	20	41	49%
Toyota	93	149	62%
Waymo	50	112	45%

1.2 Challenges

For operation in real world traffic, an autonomous vehicle will need to predict the behaviors of other traffic participants so that it can analyze the risks involved well in advance. Doing so enables graceful handling of the situations, also avoiding the ones that may lead to near term risks. Current Advanced Driver Assist Systems (ADAS) like collision avoidance, adaptive cruise control, are designed to use simplistic physics based models to predict the motions of the traffic participants to analyze a possible dangerous situation, and take necessary evasive action. The interactions between the traffic participants are typically ignored assuming they will maintain the states of their kinematic motions irrespective of what other participants in their vicinity do. However, this approach is effective only in the time horizon of 1-2 seconds i.e. a human response time [46]. For long horizon risk analysis however, the autonomous vehicle systems need to be able to predict possible maneuvers for every relevant traffic participant (other vehicle) to plan the feasible maneuvers for its own safe navigation. In a heterogeneous traffic, where both connected automated vehicles and conventional vehicles share the road, this has to be done by utilizing merely the extrinsic information of the observed traffic participants and is a challenging problem. Understanding the manner in which a human driver executes a maneuver can aid in addressing this challenge. Any maneuver performed by a traffic participant can be broken down into following four stages:

i) **Intent determination:** The intent specifies what maneuver the driver wishes to perform eventually and is typically based upon the surrounding traffic participants as well as upon the traffic situation. The reason for the initiation of most maneuvers lies usually within a mandate arising from the route (exiting a freeway, turning at an approaching intersection, merging from a ramp, etc.) or is due to a traffic obstruction (such as a stopped vehicle, construction area, end of lane, etc.). Yet, there are some maneuvers such as lane change, where the intent could be at discretion and is usually attributed to a perceived benefit in the resulting driving conditions (e.g. increase in speed, clear look ahead, etc.).

ii) **Maneuver preparation:** After self-identifying an intent for performing a maneuver, the driver progresses into the preparation stage in which the intent may be advertised using turn indicators or hand signals for instance. The driver may then accelerate, decelerate or simply maintain the speed to work out a sufficient gap between the relevant traffic participants involved in the traffic scene.

iii) **Gap acceptance:** If the spatio-temporal gap between the relevant nearby traffic participants is perceived as sufficiently safe, the driver progresses to the execution phase. If the gap is considered to be not safe, the driver falls back to the preparation phase. Although the gap acceptance is more of a decision point for transitioning from the maneuver preparation to the maneuver execution stage, it is a factor that is influenced by the driving style independently from maneuver preparation. For instance, a driver may prepare aggressively for a maneuver (e.g. by increasing the speed), but may undergo a change of mind based upon the eventual gap. This justifies to consider gap acceptance as an individual stage.

iv) **Behavior execution:** In this phase, the driver provides necessary steering, acceleration or braking inputs to perform the desired maneuver.

The key challenge arises from the altering and driver-dependent consideration of decision factors when determining intent. Apart from the mission level goals, the factors influencing such decisions may also vary based on traffic, weather, and road conditions. Furthermore, the manner of preparing for the maneuver after determining the intent differs with every driver. Depending on the traffic situation, preparation may or may not be needed. Some drivers may increase the speed (e.g. to move ahead of a vehicle in the adjoining lane when changing lanes), others may decrease the speed in the same context. The tolerances for gap acceptance (to perform the desired maneuver) also range among drivers. Thereafter, the way in which the maneuver is executed might also deviate significantly. Some drivers may turn or change a lane in a quick short way, others may do it more gradually. To predict the evolution of a traffic situation effectively, estimation of the above characteristics of the traffic participant at any given instant is important. The small time window of observation for inferring these characteristics further amplifies difficulties.

1.3 Research Objectives & Scope

The goal of this research is to develop new methods that enable safe operation of autonomous vehicles in complex traffic environments while sharing the roads with human participants. Specifically, this work seeks to address the above mentioned challenges by incorporating the driver type characterizing of a traffic participant. It rests on the hypothesis that for a given driver, the behavior not only spans across the four maneuver stages mentioned in the previous section, but also across multiple maneuvers. Estimating the characteristics of one or more maneuver stages for a traffic participant aids in identifying the driver type. In turn, this helps to predict the nature of subsequent maneuvers the driver is likely to perform, enabling trajectory prediction for a longer time horizon.

While the definition of human traffic participants includes pedestrians, bicyclists, motorcyclists as well as conventional vehicles, the scope of this work will be focused only on vehicle based traffic participants. Further, this work assumes that the data association for traffic participants in the environment is perfect, the information of object tracking is available and as a result the commonly chosen physical states of the vehicle as well as the other traffic participants are estimated well. It also assumes that the scenarios and situation classification is available.

Following research questions will be addressed as a part of this work:

(RQ1) How to predict the trajectories of the traffic participants for time horizons longer than the human response time (1-2 secs) using only their extrinsic information?

(RQ2) How to model interactions between the traffic participants that follow different driving policies?

The first research question aims at investigating the system frameworks that enable modeling the human characteristics influencing their driving manners. The outcome of this research would help to improve risk analysis and planning capabilities of autonomous vehicles.

The second research question aims at investigating the methods that enable operation of autonomous vehicles under varying driving circumstances (for instance peak hour traffic vs non-peak), where the human driving styles may vary even in a given geographical region. The outcome of this research question will improve robustness of operation of autonomous vehicles towards different driving situations and hence accelerate their deployment.

These research questions are stemmed in the realization that the long journey to eventually solving the challenges that autonomous vehicles will face while sharing the road with conventional vehicles will have to be in steps. Since the driver behavior can vary geographically, and the factors that influence the driver behavior are numerous and not entirely understood, model based approaches may edge out at some point. Data driven approaches have proven to outperform model based approaches in many other areas involving human behaviors like speech and text recognition. However, driving is a high level task with a lot of hidden variables and influencing parameters. In order to even explore data driven approaches, as a first step, methods for generating and automated labeling of large amount of real driver data are needed. The outcome of this research will open up new opportunities in automated labeling of driver data. Further, the research will not only help open new opportunities for understanding human driving behaviors but will also accelerate the pace with which this challenge is addressed.

1.4 Intellectual Merit / Research Contributions

This research addresses the challenges of prediction in complex traffic environments by characterizing the driver behaviors of the relevant traffic participants in the scene. Doing so, helps in estimating the policies for their driving and hence gives a better prediction for their future trajectories. Following are the contributions of this research:

- A novel probabilistic framework for prediction was introduced that uses multiple driver policies to characterize the behavior of different traffic participants and model interactions between them [35].

- The proposed framework was implemented by extending an interacting multiple model (IMM) filter with the contextual information of the observed traffic participants. The approach for characterizing the driver behavior was demonstrated for two simulated use cases. The simulated results also show an improvement in the performance for behavior classification compared to a traditional IMM.

- The work also compared two Multiple Model Adaptive Estimations (MMAE) approaches for behavior classification, Autonomous Multiple Model (AMM) approach and IMM approach. The IMM approach was validated using a trajectory extracted from a real world dataset.

1.5 Broader impact

The proposed work impacts the transportation safety and efficiency by accelerating the deployment of autonomous vehicles. When running online in an autonomous vehicle, the approach enables operation in complex traffic scenarios by determining the "active driver policy" as well as driving behaviors of the traffic participant. These assist in providing a more accurate prediction for risk analysis.

When running offline or online on a road side unit (traffic light controller etc.), it can serve two benefits: it can enable automated labeling of the driver characteristics and maneuvers of the traffic participants in the collected data, which can be used for training the machine learning based driver models offline; and it can also characterize the predominant driver models used in that geographical area, depending on the lane and traffic conditions. This can be a valuable input for autonomous vehicle operation as well as microscopic traffic analysis.

1.6 Dissertation organization

This dissertation is organized as follows. Chapter 2 discusses the needs for the prediction in complex traffic environments and gives the state-of-the-art in prediction identifying the research gaps available. Chapter 3 provides preliminaries on the motion models considered for the work. Chapter 4 proposes the probabilistic framework for representing the traffic participant as a Dynamic Hierarchical Bayesian Network. Chapters 5, 6 and 7 explore two approaches for MMAE: AMM and IMM and provide an in-depth comparison of the two for behavior identification of a traffic participant. Finally, chapter 8 implements the proposed probabilistic framework by extending the IMM approach and presents the results of evaluation for behavior identification as well as behavior characterization. The conclusion and future works are summarized in chapter 9.

Chapter 2

State Of The Art: Prediction In Complex Traffic Environment

Deployment of highly autonomous vehicles on the road requires them to be able to plan their motion in complex environment(environments with multiple maneuvering participants). Planning problem in any environment can basically be considered as an optimization problem with following steps: (i) identifying the possible states (i.e. search space) that the vehicle can attain while staying within safety constraints; (ii) performing an optimized search through this search space using an objective function (cost function) that represents the identified constraints of the motion; and (iii) providing a reference to the motion execution modules of the vehicle. Depending on the different factors considered while generating a search space, the different levels of planning are involved in navigation.

1. Path planning: A path is a collection of poses that a vehicle has followed in the past or might be following in a finite future [62]. For navigation through a ground environment, it usually consists of states like longitudinal position, lateral position and orientation in case of 2D navigation in a road environment but without any information of time. Path planning is determining the future path of the vehicle given its

present states, available map and mission level goals. States of the other participants of the traffic environment are usually not considered in this.

2. Trajectory planning: Trajectory is a collection of position, orientation, velocity and/or higher derivative states of a vehicle marked with time stamps. A path can be extracted from a trajectory, but trajectory cannot be recreated from path due to lack of time information[62]. Trajectory planning determines the future trajectory of the vehicle given its present states, map, goal and the states of the other participants of the environment.

3. Behavior planning: It can be seen in the literature [36, 73, 47, 46] that the terms maneuver and behavior are often used interchangeably. However, it can be argued that they are separate. This work differentiates between the two as follows. A maneuver is a collection of motion sequences that a driver executes in order to achieve a local or a global level mission objective and is usually described by a symbolic label like "lane change", "lane merging", "left turn", "right turn", etc. On the other hand, the policies and the physical parameters that govern the kind of inputs that the driver gives to the vehicle for performing the maneuver constitute the behavior. In other words, different behaviors can result in the different manners in which the same maneuver can be realized. Behaviors for an identified maneuver can typically be described by trajectories since they involve specifying stopping times, velocities and possibly accelerations at different spatiotemporal points in the future. Hence, behavior planning is determining the future trajectories of the vehicle given its present states, map, goals and the predicted behavior of the other traffic participants in the environment.

Navigation through a complex traffic environment requires planning to be performed at the behavior level. Given the current states of the environment and its participants, analysis of risks that account for the participant behaviors provides the necessary search space for such planners. Risk analysis can be decomposed into following 3 steps[46, 27, 47]:

i) identify states of all the traffic participants

ii) predict their future states in a time step *t+s* in the future

iii) check the possibility of a risky event (e.g. collision) at each time step

To assess the risks in the correct domain space for effective planning, we need to understand the needs of real-world driving scenarios. Collisions with other entities (pedestrian, bicyclists, other vehicles, stationary objects, etc.) are the primary risks that need to be avoided, but in the real world the drivers also avoid executing behaviors that may lead to a possible collision. Even if such behaviors don't lead to collisions, they do disrupt the smooth flow of the traffic. Hence, risks include behaviors that may lead to a risky situation. Further, the risks can also include violating a road traffic rule or a road etiquette, or even inconvenient driving conditions that may cause an injury to the occupants or the vehicle functionality in the short or long term. Accurately identifying and predicting the trajectories (behaviors) of all the relevant traffic participants becomes key to analyzing the risks and is a challenging problem.

This chapter seeks to approach the challenges of navigation through complex traffic environments by collectively looking at the needs of the operating domain as well as the state-of-the-art of the technology from a broader perspective to build the foundation for the right solution. It is organized as follows. Section 2.1 provides a fresh taxonomy of safety / risks that an autonomous vehicle needs to address while navigating through traffic. Section 2.2 focuses on the needs of prediction for a longer time horizon and then covers the state-of-the-art. The chapter concludes with identifying the research gaps.

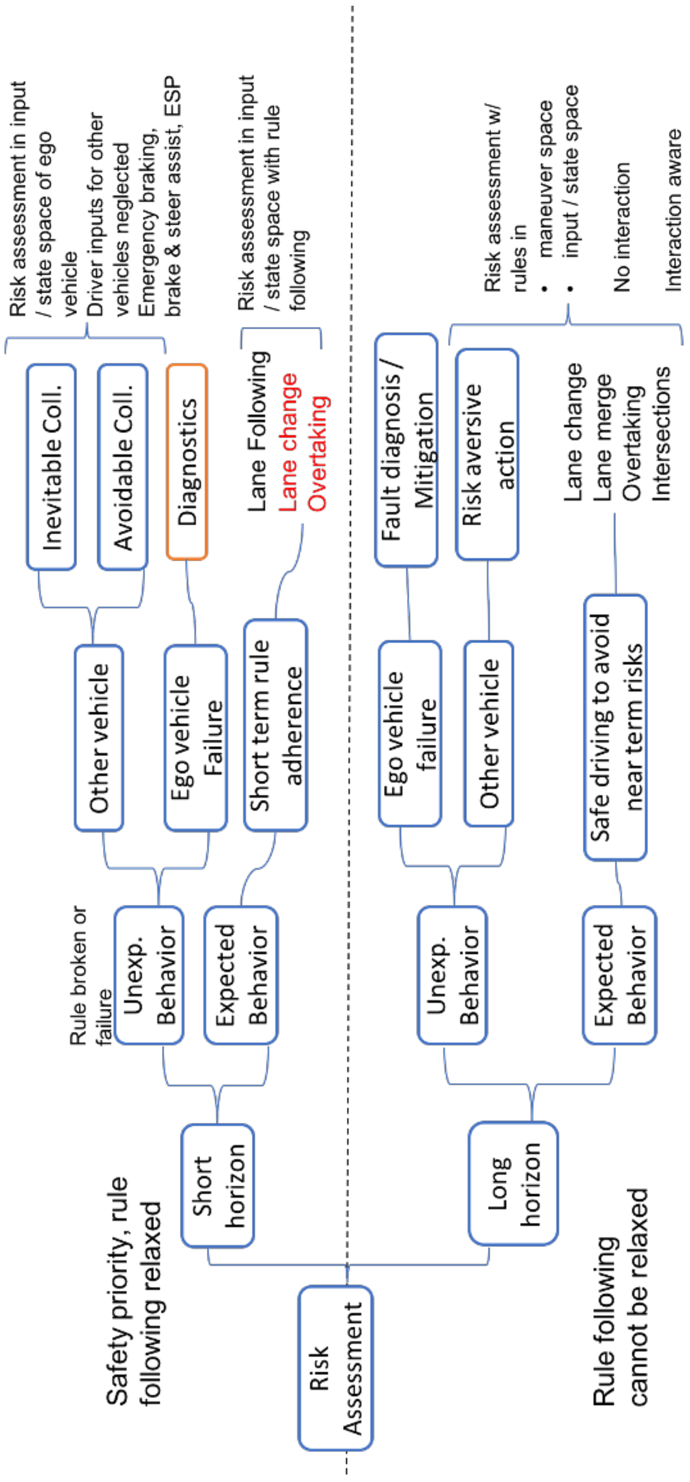


Figure 2.1: Risk assessment classification

2.1 Risk Assessment Classification

At the highest level, risk assessment can be classified as either near term (short horizon) or long horizon as illustrated in Fig. 2.1. Whenever the foreseen evaluation of the risk is within the temporal vicinity of human reaction time, it is considered as near term, otherwise it is interpreted as long horizon. This distinction has been chosen due to a key factor in determining the risk mitigation strategy (i.e. motion planning): it must be decided whether the traffic rules should be followed or not. For near term risk prevention, the dominating priority of the planners lies within risk avoidance irrespective of rule following. For long horizon risk prevention, however, the planners need to obey traffic rules. As a result, mitigation strategies for both categories of risks necessitate different policy models. For both classes, identifying whether a traffic participant performs an expected or unexpected behavior (as per traffic rules), can further assist in identifying a risk situation. One of the recent works for the identification of such unexpected behaviors can be found in [47].

2.1.1 Short horizon risk assessment

For the short horizon risk assessment, existing literature ignores the interactions between the traffic participants. Physics based models (like constant velocity, constant acceleration, constant turning rate acceleration, constant curvature, etc. [48]) are utilized for predicting the trajectories under the assumption that the participants are moving independent of each other and hence will maintain their dynamic states. The argument that the interactions between the vehicles can be ignored holds true only for the time horizon of a human reaction time, which is about 1-2 seconds.

If an unexpected behavior is observed, i.e. another traffic participant is at fault (due to a rule breaking or a failure), the risk analysis needs to determine whether the collision is inevitable or avoidable. Here, metrics such as time-to-collision, time-to-react, distance-to-collision, etc. can be utilized [46]. Example situations in this category include: a participant cutting across while the ego car drives straight; an on-coming vehicle crossing the divider and entering the ego vehicle's lane. If the collision is inevitable, the risk analysis module must be able to provide an assessment of impact, such that a behavior minimizing damage, for instance, can be planned simultaneously as the system is preparing for a collision. Most of the emergency Advanced Driver Assistance Systems (ADAS) fall into this category. If the collision is avoidable, the risk analysis module needs to provide an assessment such that an evasive behavior (braking only, steering only, or a combination of braking and steering) with least possible consequences can be planned. Whether the collision is avoidable or inevitable, planning for this situation is performed with low priority attributed to rule following and higher priority attributed to realizing a maneuver with least possible consequences.

Unexpected behavior can also be observed for the ego vehicle likely due to a failure in some vehicle module. In such situations, the fault diagnosis and fault mitigation routines need to take over as typical models may fail. This research area is out of scope of the presented work.

For the case in which all participants behave as per the traffic rules, the main objective of the risk assessment module amounts to aiding short term rule adherence for the ego vehicle and analyzing trajectories that may lead to dangerous situations. Driver assistance systems such as lane keeping and adaptive cruise control fall under this category.

Most of the SAE Level 1 and 2 ADAS systems in the market address risks in this category. Lane merge assist systems that are on the horizon also address such risks, however a lane merge maneuver isn't always a short term maneuver. Collision-based risk prediction has been an active topic for researchers for quite some time now with a large body of literature available [17, 41, 69, 33, 53, 16, 64, 40, 19, 44, 13, 27, 31, 12, 68]. The popular approach in such cases is to assume no inputs from the driver, make assumptions about constant velocity, acceleration, steering angle or steering rate, predict the trajectories of the traffic participants and then check for an event like time-to-collision, distance-to-collision[46], time to closest approach, etc. to determine possible conflicts. Some of the notable works that implement computationally effective methods to determine possible conflicts are discussed here. Campos et. al. [27] provide the 3 step procedure for the threat assessment for collision avoidance at intersections. The three step procedure is: (i) use unscented transform to predict the trajectories using constant turn rate acceleration (CTRA) model; (ii) define some geometric areas (collision zones approximated as rectangles) on the vehicles for calculating the time-to-collision (TTC) and distance-to-collision (DTC) and then use bivariate normal distribution integral approximation (Drezner, 1978[29]) to find the probability of collision. (iii) employ reachability analysis for assessment of threat. This publication also gives the values of covariance parameters from experiments, which are typically challenging to determine. Batz et al [16] use unscented transform for predicting trajectories. Here, covariance ellipses are utilized in calculating the area representing the position of the vehicles, adding uncertainty of orientation and then applying Minkowski's sum operator to it for every vehicle. Then for every vehicle pair, the approach calculates the minimum distance in the short time in the future, if this distance falls below certain threshold it flags a dangerous situation. However, these methods are usable only for short term prediction as they ignore driver intent, which is the key uncertainty for long horizon risk assessment.

2.1.2 Long horizon risk assessment

This category addresses analyzed risks that are spatially or temporally distant enough to effectively take risk averse actions in order to gracefully handle the situation. As such, the assessment rather analyzes for possible near term risky situations than events, incorporating the notion that risky events will be avoided implicitly if risky situations can be avoided. Here, models assume the traffic participants to be either interaction-aware or non-interactive. For complete risk analysis, however, the first case needs to be considered. Irrespective of whether an unexpected behavior has been observed or not, rule adherence in the planning must be ensured.

If an unexpected behavior on the part of another participant is observed, the risk assessment module analyzes different strategies to aid in planning a risk-averse maneuver gracefully handling the situation. The strategies can either be based on the possible maneuvers available to the ego vehicle or on the input space of the vehicle [26]. An example scenario would be when the ego vehicle comes across a vehicle blocking an intersection. The graceful behavior in such case would be either to brake to a complete halt or change lanes in order to drive around the slower vehicle.

If no unexpected behaviors are observed for other traffic participants, then, the risk assessment consists of safety and efficiency analysis for driving maneuvers such that potential near-term risks can be avoided. Examples include a car parked on the shoulder: here, typical behavior of the vehicles driving in the adjoining lane is to either slow down or change lanes. Another example could be a multiple lane intersection with merge-ins as vehicles in an adjoining lane can change lanes to avoid any possible conflicts with merging vehicles.

For long term risks, necessary steps for risk assessment are:

1. Model and predict possible/expected behaviors of other traffic participants

2. Analyze the potential risk for every possible behavior of the ego vehicle against the predicted behaviors of other traffic participants

3. Identify the realized behavior as soon as the other traffic participant executes a maneuver

4. Differentiate an unexpected maneuver from the expected ones if such a situation oc-

curs

Clearly, the most challenging part in the above steps is predicting the behavior (and implicitly the trajectories) of the traffic participants accurately for a long time horizon. This is due to three kinds of uncertainties that need to be factored into the prediction [71]:

i) The uncertainty in the maneuver that may be executed, which depends on the situation and may vary with the driver for a given situation

ii) The uncertainty in the manner in which the maneuver is executed, which depends on the driver

iii) The uncertainty in the state estimates of the observed participant due to sensor noise.

The prediction is also the most crucial step, since the risk analysis will be as effective as the accuracy of the predicted behaviors. Studies addressing the challenges in this category are relatively recent. Lefevre et al. [47] presented a framework for reasoning for collision risk at a semantic level by differentiating expected behaviors of the drivers from the unexpected behaviors. Their approach rests on the assumption that participants that execute unexpected behaviors invoke dangerous situations. The framework was demonstrated by applying to the road intersections with interactions. Among the most recent works, Damerow [26] presents a situation-based risk evaluation and behavior planning framework for highly automated driving. Analysis of risks is performed using prototypical predicted trajectories of the traffic participants and the result of the analysis is a proposed risk map that is basically a search space for the behavior planner. The framework is demonstrated for parallel driving and intersection traffic scenes. Prototypical trajectories based approaches however implicitly assume the availability of the high definition digital maps for effective operation. Further, the framework is decoupled from the environment perception module and doesn't reuse the information about uncertainty already available from it.

The focus of this work is to enable prediction for a longer time horizon and it also subsumes behavior identification. The next section covers the related work in prediction.

2.2 Prediction for long time horizon

Accurate prediction of trajectories for long time horizon requires the interactions between the traffic participants to be considered. In heterogeneous traffic, where both connected automated vehicles and conventional vehicles share the road, prediction has to rely entirely on observing the traffic participant extrinsically and is a challenging problem. Understanding of the manner in which a human driver executes a maneuver can help in addressing this challenge. Any maneuver performed by a traffic participant can be broken down into following stages [56], and can be collectively utilized to infer the driver behavior:

i) **Intent determination:** The intent specifies what maneuver the driver wishes to perform eventually, and is typically based on the surrounding traffic participants as well as the traffic situation. The cause for most of the maneuvers is usually either a mandate due to the route (exiting a free way, turning at an approaching intersection, merging in from the ramp, etc.) or due to a traffic obstruction (like a stopped vehicle, construction, end of lane, etc.). Yet, there are some maneuvers like lane change, where the intent could be at discretion and is usually attributed to a perceived benefit in the resulting driving conditions (e.g. increase in speed, clear look ahead, etc.).

ii) **Maneuver preparation:** After self-identifying an intent for performing a maneuver, the driver progresses into the preparation stage where the intent may be advertised using turn indicators or hand signals for instance. She may then accelerate, decelerate or simply maintain the speed to work out a sufficient gap between the relevant traffic participants.

iii) **Gap acceptance:** When the spatiotemporal gap between the relevant nearby traffic participants is perceived to be safe enough, the driver progresses to the execution phase. If the gap is not perceived to be safe, the driver falls back to the preparation phase. Although the gap acceptance is more of a decision point for transitioning from maneuver preparation to maneuver execution stages, it is a factor that is influenced by the driving style independently from maneuver preparation. For instance, a driver may prepare aggressively for a maneuver by increasing the speed, but may change the mind given the eventual gap. his justifies to consider it as an independent stage.

iv) **Maneuver execution:** In this phase, the driver gives the necessary steering, acceleration or braking input to perform the desired maneuver, e.g. lane change.

As long as the traffic participant communicates the intent explicitly (using turn indicators for instance), the prediction can be straight forward. However, studies show that in many cases (55% [56]), the maneuvers are performed either without communicating the intent explicitly, or communicated after / just before the maneuver has been initiated. In such cases the intent must be inferred from the behavior of the participant. A popular way of inferring is by running a mathematical model, associated with every relevant traffic participant, that considers their surrounding information (also called as context [36]) and then estimates their maneuver intent. Such approaches can be divided into model based approach and data driven approaches. In the model based approach, the mathematical model is a driver model that considers the context of the subject vehicle, i.e. its spatiotermporal relations with the surrounding traffic and estimates the driver intent. For the data driven approach, machine learning based approaches are used to train the mathematical model offline using data collected from different driving scenarios based on the identified parameters with the surrounding traffic. Finally, the trained model is used on-line to infer the intent of the observed vehicle. The challenge is that the factors that are considered in determining the intent vary with every driver. The mission level goals like approaching a turn or an exit / ramp that the driver wants to take also influence such decisions. And even with a given driver, these factors may vary based on the circumstances like traffic conditions, weather conditions and road conditions.

Once a driver identifies an intent for performing a maneuver, the driver accelerates or decelerates to prepare for the maneuver. The driver maintains this stage until gap acceptance, where the driver determines that the gap between the traffic participants is sufficient / safe enough to perform the maneuver. Finally, the driver proceeds to the execution of the maneuver. Depending on the traffic situations the the preparation may or may not be needed. However there can be a variation in the way different drivers prepare for a maneuver. Some may increase the speed to move ahead of a vehicle on the side for a lane change for instance, others may decrease the speed. Further the tolerances for gap acceptances also varies with the driver. Finally, the way in which the maneuver is executed also deviates. Some may take a quick short turn or lane change, others may perform it more gradually. preparation and execution stages are basically motions based on the decision on an accepted threshold of the gap. The maneuver execution can be represented as motion models for the maneuvers, with the variability expressed in the form of parameters.

The approach to model the driver intents using contextual information has been utilized in the microscopic traffic analysis for quite some time. Gipp's car following [37] and Intelligent Driver Model (IDM) [67] are two of the most popular car following models. One of the recent additions to this is a Foresighted Driver Model (FDM) [30]. Likewise, for lane change intents, Gipp's lane change [38] and Minimum Overall Braking Decelerations Induced by Lane changes (MOBIL) [42] are the popular ones. Toledo et al. were one of the first ones to propose an integrated lane change and car following model [66]. For a comprehensive review on the lane change models, the reader is referred to Rahman et al. [56]. Apart from the microscopic traffic analysis models, Dou et al. [28] presented their approach for predicting lane changes at higway lane change drops using SVM, Artificial Neural Network (ANN) and a combined classifier. Contextual information like relative speed, difference and distance with the nearby vehicles are given as inputs to these classifiers, and the output is a vote suggesting merge or not to merge. The approach is evaluated on the I80 and US 101 datasets in the publicly available NGSIM database. All the above models focus on intent determination as well as the gap acceptance aspects of the driver behavior.

Amongst the recent works, Rehder et al. [57] use logistic regressions to identify lane change intentions. The model was trained with data from humans driving on a simulator. Influences like traffic rules, road structure, individual driving style, environmental conditions as well as influence from co-drivers was considered to model the interactions. Aoude et al. presented a machine learning base model for driver intents at intersections [14]. They used Support Vector Machines with Bayesian Filter (SVM-BF) and Hidden Markov Model (HMM) based models to classify the behaviors of the traffic participants as "compliant" or "violators". The model was evaluated with the real world data collected in Christiansburg, VA, through the US Department of Transportation Cooperative Intersection Collision Avoidance System for Violations (CICAS-V) initiative. Social Force Model (SFM) [39] is a popular approach for modeling interactions with the pedestrians. Recently, several studies have utilized SFM to model vehicle to vehicle interactions. One such study is by Yoon et al. [74], where SFM is utilized on a multilane highway to model the tendency of the drivers to keep safe distances from the object vehicles, to get to the maximum allowable speed on the route and to stay at the center of the lane. The works in this paragraph focus only on the intent determination stage of the driver behavior.

The works that identify driver behaviors based on the maneuver intent and maneuver execution are [26, 71, 72, 11, 15, 36, 10, 20, 47] . Damerow [26] utilizes the FDM to model the driver intent and trajectory matching with prototype trajectories to predict the maneuver. Wissing et al. [71] use IDM and MOBIL for intent determination and Monte Carlo approach for execution. They demonstrate their approach for lane keeping, left and right lane change maneuvers. In another work by them [72], contextual information, such as relative distances and relative velocities between the observed vehicle and its nearby vehicles, is utilized along with the spatiotemporal information of the observed vehicle, such as lateral distance to lane center, lateral velocity, longitudinal velocity, as features for training an SVM model that classifies the maneuver and predicts the time to lane change. This information is used to model the predicted trajectories as cubic splines with constant velocity assumption. Alin et al. [11] describe the use of attractor functions to determine the context of the system and predict the potential target locations for the traffic participants. They demonstrate this on a lane following as well as double lane scenarios and use grid-based Bayesian filters for modeling the maneuver execution. The approach was evaluated using TORCS simulator. Bahram [15] uses game theory for modeling intent and Naive Bayes for classifying the maneuver based on the longitudinal and lateral motion. The approach is demonstrated for lane keeping, left and right lane change maneuvers. An approach that stands out from above works in this paragraph is by Gindele et al. [36]. This work was the first comprehensive framework that integrates the interaction-aware driver intent determination and maneuver execution stages together in a probabilistic filtering framework. The work represents the traffic scenario as a Dynamic Bayesian Network and neatly ties together the context, maneuver, predicted trajectories and the base states (continuous physical states) of the traffic participants. The approach is then demonstrated on a two lane highway scenario for an overtaking situation using particle filters. This framework is also adapted by a few other other works. Agamennoni et al. [10] adopted this framework and utilize feature functions to extract the high level information from low level contextual information. The inference in their implementation is deterministic and is implemented for mining safety on haul trucks. Brechtel et al. [20] adopted the same framework in their work for behavior planning for autonomous vehicles. Their framework is implemented with Partially Observable Markov Decision Process (POMDP) and is evaluated at runtime on a simulator for overtaking situation. Due to the intuitive design of the framework, our approach also extends the framework by Gindele et al. [36]. Further, just like Agamennoni et al. [10] we utilize our own feature functions. The details for this can be found in section 4.4.

The works that focus on maneuver execution phase of the driver behavior are [22, 51, 59, 50, 60, 73]. Carvalho et al. [22] use Interactive Multi Model based Kalman Filtering (IMM-KF) for modeling left lane change ,lane keeping and right lane change maneuvers. The longitudinal motion is modeled with constant acceleration and constant velocity models, where as the IMM-KF is utilized for lateral motion. The implementation uses a bank of 9 elemental filters each running a second order motion model. Liu et al. [51] use cascaded HMM classifiers to identify the behaviors based on maneuver execution. In the first stage, an HMM for a traffic participant classifies the maneuver type as lane change left, right or lane keeping based on its base states. In the second stage two HMMs are modeled, one for an normal driver and the other for a dangerous driver. Given the maneuver, this state maps the base states of the participant to a symbolic driver behavior. In the final stage, the likelihood of the models is used to give the decisive classification. Schreier et al. [59] use Bayesian inference with multiple motion models to identify dangerous drivers. They demonstrate their work for lane following, lane changes, turns and for a few trash maneuvers. Among the most recent works, Li et al. [50] is Gaussian Mixture HMM (GMHMM) for maneuver based threat assessment in lane keeping, left and right lane change maneuvers. Their observation vector consists of lateral offset, lateral velocity and lateral acceleration. Constant acceleration model is used for lane keeping motions and sinusoidal half lane change with constant velocity for lane changes. Schulz et al. [60] use Autonomous Multiple Model (AMM) [49], also known as classical Multi Model Adaptive Estimation (MMAE) [25] filters with Unscented Kalman Filters (UKF) for an intersection scenario. The number of modes in their model depends on a possible combination of route and maneuver intentions. They evaluated their model with 6 modes for upto 15 vehicles, and concluded that their approach outperforms a particle filter with upto 50,000 particles. Further, the runtime of their cycle was 0.5 secs as against 7 secs for the particle filter. Xie et al. [73] utilize Interactive Multiple Model (IMM) to integrate maneuver based and physics model model for trajectory prediction. Constant Turn Rate Acceleration (CTRA) [48] model was utilized for the physics based mode and sinusoidal lane change was utilized for the maneuver based model. The model was evaluated on naturalistic data collected and was validated to be superior than the single model approaches.

To the best of my knowledge, the only available framework that models maneuver intent, gap acceptance and maneuver execution together is by Lefevre et al. [47]. Their framework extends the framework by Gindele et al. [36] for reasoning of collision risk at a semantic level by differentiating expected behaviors of the drivers from the unexpected behaviors. The dangerous situations are identified by comparing what drivers intend to do with what they are expected to do according to the traffic rules. The authors propose a Markov State Space model (MSSM) based graphical representation where the previous intentions of the other vehicles in a multiple vehicle situation influences the expected behaviors of the vehicles. A hierarchical framework is presented in which the highest level corresponds to the expected and the intended maneuver performed, next level corresponds to the physical state of the vehicle and the lowest level the measurements. The framework is applied to road intersection scenario and uses a feature function for gap acceptance to decide on the necessity to stop. The risk estimation is performed using a bootstrap filter for approximate inference.

Table 2.1: Comparative review of works on driver behavior identification

			Driver Behavior				
Study	Remark	Behaviors	Intent determination	Maneuver preparation	Gap acceptance	Maneuver execution	Driver update
Gipp's 1981 [37] IDM - Helbing 2000 [67] FDM - Eggert 2015 [30]	Rule based	Car following	✓		✓		
MOBIL - Helbing 2007 [42] Gipp's 1986 [38]	Rule based	Lane change	✓		✓		
Toledo 2007 [66]	Rule based	Integrated lane change	✓		✓		
Dou 2016 [28]	SVM, ANN	Lane merge	✓		✓		
Rehder 2016 [57]	Logistic regression	Lane change	✓				
Aoude 2011 [14]	SVM-BF, HMM	Intersection	✓				
SFM - Helbing 1995 [39]	Rule based	Intersections with pedestrians	✓				
Yoon 2018 [74]	SFM with NMPC	Multi lane driving	✓				
Damerow 2018 [26]	FDM, prototype trajectories	Intersection, lane following	✓			✓	
Wissing 2018 [71]	IDM, MOBIL with Monte Carlo	Lane merge	✓			✓	
Wissing 2018 [72]	SVM, cubic splines	Lane change	✓			✓	
Alin 2012 [11]	Attractor functions, grid-based BF	Lane keeping & changing	✓			✓	
Bahram 2017 [15]	Game theory, Naive Bayes	Lane keeping & changing	✓			✓	
Gindele 2010 [36]	DBN, particle filter	Overtaking	✓			✓	
Agamennoni 2012 [10]	DBN, feature functions	Intersection	✓			✓	
Brechtel 2011 [20]	POMDP	Overtaking	✓			✓	
Carvalho 2014 [22]	IMM-KF	Lane keeping & changing				✓	
Liu 2015 [51]	Cascaded HMM	Lane keeping & changing				✓	
Schreier 2016 [59]	BF	Lane keeping & changing				✓	
Li 2018 [50]	GMHMM	Lane keeping & changing				✓	
Schulz 2018 [60]	AMM	Intersection				✓	
Xie 2018 [73]	IMM	Lane change				✓	
Gill 2019 [34]	AMM	Lane keeping & changing				✓	
Lefevre 2015 [47]	DBN, feature functions, boot strap filters	Intersections	✓		✓	✓	
Zhou 2018 [75]	POMDP	Intersection	✓	✓		✓	
Gill 2019 (Proposed) [35]	DBN	Lane keeping & changing	✓		✓	✓	✓
??	??	??	✓	✓	✓	✓	✓

2.3 Research Gaps

Table 2.1 summarizes the presented works mapping them against the stages of driver behaviors for a maneuver. To the best of my knowledge, following gaps exist in the literature:

1. There are no probabilistic filtering frameworks that couple together the four stages of the driver maneuver discussed in the section above.

2. There are no filtering frameworks that characterize the driver style and update it on the fly based on the behavior identified from one of the maneuver stages to influence on the other stage. Driver behaviors have a correlation between the maneuver stages. Characterizing the driver early on helps predict their intents, and hence the trajectories for long horizon more accurately. For instance, identifying the behavior of rash driver in the maneuver execution stage reveals additional information about the intent determination nature of the driver. Without associating these stages to a driving style and coupling them together, crucial correlation between these stages is lost.

3. None of the works utilize the maneuver preparation stage of the driver for characterizing the behavior. Utilizing this information can not only help identify the intent of the driver, but also identify the driving style such as "cooperative", "opportunistic", "aggressive" even before the initiation of the maneuver.

Chapter 3

Preliminaries: Motion Models

3.1 Motion models explored in this study

Prediction and tracking share similar foundations when it comes to functioning but with some significant differences. As against tracking, where the propagation horizon is relatively short, is independent of the surroundings of the participants, and is typically utilized in missed measurements or occlusions, the propagation horizon for prediction is longer as it is utilized for planning. As a result, the error in prediction can differ by a large margin if the motion model is incorrect, leading to sub optimal planning. The key to prediction is identifying the right motion model that the target is believed to be following. The more accurately the model represents the target, the longer the horizon of accurate prediction.

Ground vehicle motions are typically represented as 2D motions in the longitudinal and lateral directions. The resulting motion models can be divided into symmetric and asymmetric. Symmetric motion models have the same propagation equations representing the longitudinal and lateral motions, whereas asymmetric ones have different propagation equations for longitudinal and lateral motions. The effectiveness of the motion model depends on the operation domain of the target. For an unstructured environment where the motion can be expected in both lateral as well as longitudinal direction symmetric models work well. Constant acceleration (CA), constant velocity (CV) are examples of symmetric motion models. For a road traffic environment, which is structured in terms of rules and motions, and where larger motion is expected in the longitudinal direction than lateral, and that certain predefined distributions of movements are performed for a typical maneuver, asymmetric models are more effective. The continuous time state-space models for such mathematical models are given by:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t), \mathbf{x}(t_0) = x_0 \tag{3.1}$$

$$\dot{\mathbf{z}}(t) = h(\mathbf{x}(t), \mathbf{u}(t), t) + v(t) \tag{3.2}$$

The linearized, discrete time equivalent of the above model are of the form:

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k) + B(\mathbf{u}_k) \tag{3.3}$$

$$\mathbf{z}_k = H(\mathbf{x}_k) + D(\mathbf{u}_k) \tag{3.4}$$

where F and H are the Jacobian matrices of the state and measurement equations respectively, if they are non-linear. Following are some of the motion models that were explored in this work. Many of these come from the field of tracking. For ground vehicles, the motions are typically considered in x-y plane.

3.1.1 Constant Velocity Motion Model

Constant Velocity (CV) motion model represents the linear motions of a vehicle and can be used to represent straight, free flow highway like driving conditions where the vehicle speeds are nearly constant over a period of time and the roads are nearly straight.

The state vector comprises of,

$$\dot{\mathbf{x}} = \begin{bmatrix} x & v_x & y & v_y \end{bmatrix}' \tag{3.5}$$

and the state equations are given as:

$$\dot{x} = v_x \tag{3.6}$$

$$\dot{v_x} = 0 \tag{3.7}$$

$$\dot{y} = v_y \tag{3.8}$$

$$\dot{v_y} = 0 \tag{3.9}$$

The discrete time counter part of the above equation with the noise is:

$$\mathbf{x}_{k+1} = F_{cv} \cdot \mathbf{x}_k + \zeta_k \tag{3.10}$$

where F_{cv} is the state transition matrix given by,

$$F_{cv} = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

ζ_{cv} is the process noise with statistics as,

$$\zeta_k \sim \mathcal{N}(0, Q_k) \tag{3.12}$$

Q_k is the process noise covariance matrix, and is usually assumed as diagonal for simplicity.

$$Q_k = \begin{bmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_{vx} & 0 & 0 \\ 0 & 0 & \sigma_y & 0 \\ 0 & 0 & 0 & \sigma_{vy} \end{bmatrix} \tag{3.13}$$

Remark. *The noise in constant velocity is correlated due to acceleration, and is given as following based on the sample time:*

$$\zeta = \begin{bmatrix} T_s^2/2 & 0 \\ T_s & 0 \\ 0 & T_s^2/2 \\ 0 & T_s \end{bmatrix} \cdot \begin{bmatrix} \zeta_x \\ \zeta_y \end{bmatrix}$$

and the covariance of the noise is given by:

$$Q = \begin{bmatrix} \sigma_x \cdot T_s^4/3 & \sigma_x \cdot T_s^3/2 & 0 & 0 \\ \sigma_x \cdot T_s^3/2 & \sigma_x \cdot T_s^2 & 0 & 0 \\ 0 & 0 & \sigma_y \cdot T_s^4/3 & \sigma_y \cdot T_s^3/2 \\ 0 & 0 & \sigma_y \cdot T_s^3/2 & \sigma_y \cdot T_s^2 \end{bmatrix} \tag{3.14}$$

3.1.2 Constant Acceleration

Constant Acceleration (CA) motion model is good at representing accelerating as well as decelerating targets and is very effective in the nearly straight urban driving conditions where the vehicles undergo a lot of changes in speeds. It assumes a nearly constant acceleration. The state vector comprises of $\dot{\mathbf{x}} = \begin{bmatrix} x & v_x & a_x & y & v_y & a_y \end{bmatrix}'$

The state equations are:

$$\dot{x} = v_x + a_x\,t \tag{3.15}$$

$$\dot{v}_x = a_x \tag{3.16}$$

$$\dot{a}_x = 0 \tag{3.17}$$

$$\dot{y} = v_y + a_y\,t \tag{3.18}$$

$$\dot{v}_y = a_y \tag{3.19}$$

$$\dot{a}_y = 0 \tag{3.20}$$

The discrete time counter part of the above equation with the noise is:

$$\mathbf{x}_{k+1} = F_{ca} \cdot \mathbf{x}_k + \zeta_k \tag{3.21}$$

where F_{ca} is the state transition matrix given by,

$$F_{cv} = \begin{bmatrix} 1 & T_s & T_s^2/2 & 0 & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & T_s^2/2 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

ζ_k is the process noise with statistics as,

$$\zeta_k \sim \mathcal{N}(0,\, Q_k) \tag{3.23}$$

Q_k is the process noise covariance matrix, and is usually assumed as diagonal for simplicity.

$$Q_k = \begin{bmatrix} \sigma_x & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{vx} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{ax} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_y & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{vy} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{ay} \end{bmatrix} \tag{3.24}$$

3.1.3 Constant Turn Rate and Velocity

Constant Turn Rate and Velocity (CTRV) is employed in representing a target that is turning and has nearly constant velocity as well as yaw rate. The state vector comprises of

$$\mathbf{x}_k = \begin{bmatrix} x & y & \psi & v & \omega \end{bmatrix}' \tag{3.25}$$

where v is the velocity tangential to the direction of motion, ψ is the yaw angle and ω is the yaw rate of the target.

State transition equations are given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \frac{v}{\omega} \cdot sin\left(\omega \cdot T_s + \psi\right) - \frac{v}{w} \cdot sin(\psi) + x_k \\ -\frac{v}{w} \cdot cos\left(\omega \cdot T_s + \psi\right) + \frac{v}{w} \cdot cos(\psi) + y_k \\ v \\ \omega \cdot T_s + \psi \\ \omega \end{bmatrix} \tag{3.26}$$

3.1.4 Constant Turn Rate and Acceleration

The Constant Turn Rate and Acceleration (CTRA) motion model is employed in representing turning motions where the targets are also accelerating. The state vector comprises of

$$\mathbf{x}_k = \begin{bmatrix} x & y & \psi & v & a & \omega \end{bmatrix}' \tag{3.27}$$

where in addition to Eqn. (3.25), a is acceleration in the tangential direction of motion of the target.

The state transition equation is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \begin{bmatrix} \Delta x(T_s) \\ \Delta y(T_s) \\ \omega \cdot T_s \\ a \cdot T_s \\ 0 \\ 0 \end{bmatrix} \tag{3.28}$$

where

$$\Delta x(T_s) = \frac{1}{w^2}\left[(v_k\omega + a\omega T_s) \cdot sin(\theta_k + \omega T_s) + a \cdot cos(\theta_k + \omega T_s) - v_k\omega \cdot sin(\theta_k) - a \cdot cos(\theta_k)\right] \tag{3.29}$$

and

$$\Delta y(T_s) = \frac{1}{w^2}\left[(-v_k\omega - a\omega T_s) \cdot cos(\theta_k + \omega T_s) + a \cdot sin(\theta_k + \omega T_s) + v_k\omega \cdot cos(\theta_k) - a \cdot sin(\theta_k)\right] \tag{3.30}$$

3.1.5 Sinusoidal Lane Change Maneuver Motion Models

As against the straight and turning motion, lane change is a maneuver that requires a specific sequence of inputs and cannot be represented with any of the above motion models. Popular approaches for representing the lane change is using a sinusoidal function, splines or bezier curves. We chose the sinusoidal motion model for lateral motion due to its simplicity and the fact that it can be represented using only two parameters: maneuver initiation point and maneuver length. Also the input that the driver gives to make a lane change results in a nearly sinusoidal steering input. It should be noted however that although the steering input to the vehicle can be sinusoidal for a lane change, the resulting trajectory of a realistic vehicle is not exactly a sinusoidal wave. The key parameters that have uncertainty associated with them are the maneuver length and the maneuver initiation point. Based on these, following three configurations of the sinusoidal motion model was explored. For longitudinal motion, a constant velocity motion model was assumed. However the equations can be adapted to any other longitudinal motion model.

3.1.5.1 Maneuver initiation point (X_{mip}) and maneuver length as parameters

The states are comprised of

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y \end{bmatrix}' \tag{3.31}$$

The state transition equations for the left lane change maneuver are given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + v_{xk} \cdot T_s \\ v_{xk} \\ -A\ cos(\omega\ \Delta x_k) + A + y_L \\ A\omega v_{xk} sin(\omega \Delta x_k) \end{bmatrix} \tag{3.32}$$

with,

$$A = \frac{w_L}{2} \tag{3.33}$$

$$\omega = \frac{\pi}{L} \tag{3.34}$$

$$\Delta x = x_k - x_{mip} \tag{3.35}$$

and w_L, L, y_L and x_{mip} are the lane width, maneuver length, lateral position from where the maneuver will be initiated (typically around lane center) and the maneuver initiation point.

The equations for the right lane change can be represented in a similar manner as:

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + v_{xk} \cdot T_s \\ v_{xk} \\ A\ cos(\omega\ \Delta x_k) - A + y_L \\ -A\omega v_{xk} sin(\omega \Delta x_k) \end{bmatrix} \tag{3.36}$$

3.1.5.2 Maneuver initiation point as parameter and maneuver length estimated

The states include maneuver length in addition to the ones in Eqn. (3.31).

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y & L \end{bmatrix}' \tag{3.37}$$

The state transition equations for the left lane change maneuver are given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + v_{xk} \cdot T_s \\ v_{xk} \\ -A\ cos\left(\frac{\pi}{L_k}\left(x_k - x_{mip}\right)\right) + A + y_L \\ A\ \frac{\pi}{L_k}\ v_{xk}\ sin\left(\frac{\pi}{L_k}\left(x_k - x_{mip}\right)\right) \\ L_k \end{bmatrix} \tag{3.38}$$

Equations for the right lane change can be extended similarly from Eq. (3.36)

3.1.5.3 Maneuver initiation point estimated and maneuver length as a parameter

The states include maneuver initiation point in addition to the ones in Eqn. (3.31).

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y & x_{mip} \end{bmatrix}' \tag{3.39}$$

The state transition equations for the left lane change maneuver are given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + v_{xk} \cdot T_s \\ v_{xk} \\ -A\ cos\left(\omega\left(x_k - x_{mip}\right)\right) + A + y_L \\ A\ \omega\ v_{xk}\ sin\left(\omega\left(x_k - x_{mip}\right)\right) \\ x_{mip,k} \end{bmatrix} \tag{3.40}$$

Chapter 4

System Architecture and Probabilistic Framework

This chapter seeks to address two challenges as part of the research work. First, how to utilize the different stages of maneuvers in characterizing the behaviors of the traffic participants? Second, how to model the interactions between different traffic participants in a traffic scene to enable long term prediction? The chapter starts with an overview of an autonomous vehicle block diagram explaining as to which block this research is being applied for. It also explains the various dependencies between different modules involved. This viewpoint helps understand the assumptions that can be made for our research. Thereafter, it proposes a probabilistic framework for modeling the traffic participants as hybrid systems using Dynamic Hierarchical Bayesian Network describing the approach for addressing the above two challenges.

4.1 System architecture for highly autonomous driving

Figure 4.1 provides the overall software architecture envisioned for a highly autonomous driving vehicle. The different modules in it are described below.

4.1.1 Localization

This module utilizes the information from the proprioceptive and exteroceptive sensors to estimate the detailed pose (position and orientation) of the ego-vehicle, including the lane and offset from the center lane, in the operating environment. Typical proprioceptive sensors for a vehicle are inertia measurement unit (IMU) and wheel speed sensors, whereas commonly used exteroceptive sensors are camera, LiDAR, radar and GPS.

4.1.2 Environment perception

This module utilizes the information from exteroceptive sensors like cameras, lidars and radars to detect stationary and dynamic obstacles in the vicinity of the ego vehicle, classifies them and tracks them to estimate their kinematic states. It also separates quasi-stationary features in the environment (e.g., parked vehicles, temporary obstacles, etc.) from permanent features that define the environment (for e.g., landmarks, lane information, traffic signs, etc.).

4.1.3 Mission planning

This module processes mission level parameters such as current position, goal position, and digital map information to plan the most optimal route for the vehicle. The resulting route does not consider traffic or lane level information in its solution. Furthermore, it is only specified in the form of way point or path data and does not contain trajectory

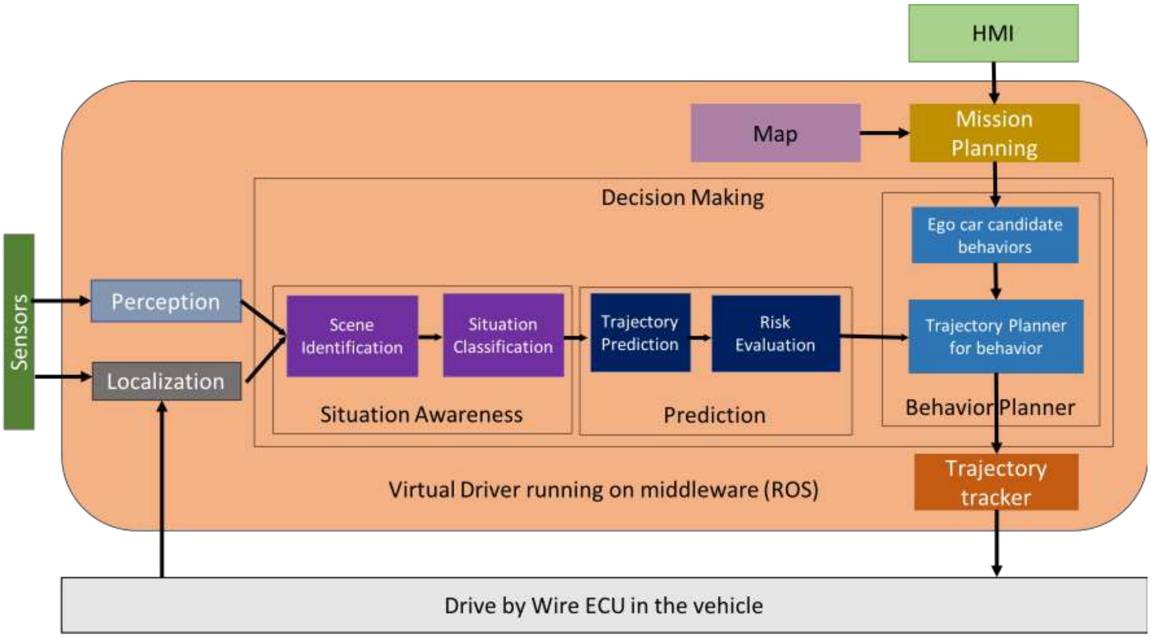


Figure 4.1: Autonomous vehicle system architecture

information. The mission planning module is also responsible in ascertaining whether the assigned mission has been accomplished.

4.1.4 Decision maker

The information about the ego-vehicle pose, the states of the other traffic participants and traffic conditions (signals, road geometry) determined from the localization and perception module are provided to the decision-making module. This module can be further divided into situational awareness (SA) and prediction sub-modules.

4.1.4.1 Situational awareness sub-module

This module uses the information from perception module as well as digital maps to identify traffic scenarios. Traffic scenarios can be thought of as a high level semantic representation of the traffic environment, like signalized intersection, non-signalized intersection, lane following, round about, etc. Each traffic scenario can be further divided into different situations depending on the overarching objective of the ego vehicle and the current states of the other traffic participants. For e.g., in an intersection where the ego vehicle intends to turn left, the situation can vary depending on whether there is another participant present in an oncoming lane that is going straight. This identification is important as it defines the traffic policies (like right of the way) that apply to all the traffic participants in the situation. After identifying the traffic scene, based on the current states of all the traffic participants, this sub-module is also responsible for identifying the current situation of the scenario.

4.1.4.2 Prediction sub-module

Based on the current situation, the prediction stage determines the possible evolution of the situation by taking into account all possible behaviors that may be executed. For each situation, trajectories for a predefined time horizon are then predicted by forward simulation utilizing driving input assumptions about traffic participants. These predicted trajectories can then be employed to assess the associated safety/risk by analyzing the traffic policies in place against all possible maneuvers available to the ego-vehicle. Metrics for this conflict can be based upon lane occupancy, potential collisions, time to collision, distance to collision, etc. The outcome of this risk analysis defines a search space utilized by the planning modules to determine the future trajectory of the ego vehicle.

4.1.4.3 Behavior planning

Here, data from prediction is connected with mission planning requirement. The route information from the mission planning module and the risk/safety assessment from prediction are combined to identify the safest behavior that the ego-vehicle should perform and to specify the trajectory realizing this behavior. The generated reference trajectory is provided to the trajectory tracker module for further processing.

4.1.5 Trajectory tracker

Based on the input from the behavior planner, the trajectory tracker represents the controller that realizes the reference trajectory in the smoothest possible way incorporating the the current state of the vehicle.

4.2 Decision making in complex traffic environments

For safe navigation in complex traffic environments, the situational awareness, prediction and behavior stages together form the decision module of the vehicle. The success of the behavior planner in navigating safely through a dynamic environment depends on the risk identification capability of the prediction stage. The prediction stage performance in turn depends on the quality of the situational awareness module in identifying the situation and applicable traffic policy. As a result, a comprehensive analysis of long-term risks necessitates a significant information flow and interaction between the three stages, and it requires a probabilistic framework capable of incorporating these interdependencies. This is in contrast to near term risk planning where these stages are decoupled. Currently, the most comprehensive framework available for prediction is provided by Gindele et. al. [36], which utilizes the concept of context and establishes the relationships between context, situations and maneuvers. This research work builds on top of their framework introducing additional states to fully characterize the driver behavior. This work belongs to the trajectory prediction sub-module of the prediction module in the architecture described above. The next section will formulate the problem of driver behavior identification mathematically, and the following section will describe the probabilistic framework.

4.3 Problem Formulation

The notation convention applied throughout this paper is as follows: capital letters represent sets of random variables, with script and regular face-types signifying discrete and continuous quantities, respectively. Lower case bold letters represent vectors while lower case regular letters stand for scalars. Further, **p(x)** represents the probability (density) distribution (pdf) of x, whereas $\mathbb{P}(x)$ represents the probability measure of x.

Let $j \in [0, J-1]$ be an index distinguishing each traffic participant in a traffic scenario, with j = 0 signifying the ego vehicle, and let $k \in [0, 1, ...\infty[$ denote a discrete time step. Then, $\mathscr{M}$ is the set of maneuvers that can be executed by a traffic participant, i.e.

$$\mathscr{M} = [m_m]_{m \in [1,...,M]}$$

with M being the total number of executable maneuvers and $m \in [1, ..., M]$ being the associated maneuver index. Examples of maneuvers are 'left lane change', 'right lane change', 'left turn', 'right turn', and 'straight' (maintain lane).

Let $\mathscr{D}$ be a set of different driver types that can be associated with a traffic participant, i.e.

$$\mathscr{D} = [d_i]_{i \in [1,...,I]}$$

where I is the total size of the set $\mathscr{D}$ and $i \in [1, ..., I]$ is the driver type index. Driver types could for instance be characterized as 'aggressive driver', 'passive driver', 'cooperative driver', etc.

Then, the possible behaviors of a traffic participant can be fully described by the potential maneuvers that it may perform and their associate driver types. For each combination of a driver type and a maneuver an identified list of parameters $\Theta_{i,m}$ can be predetermined

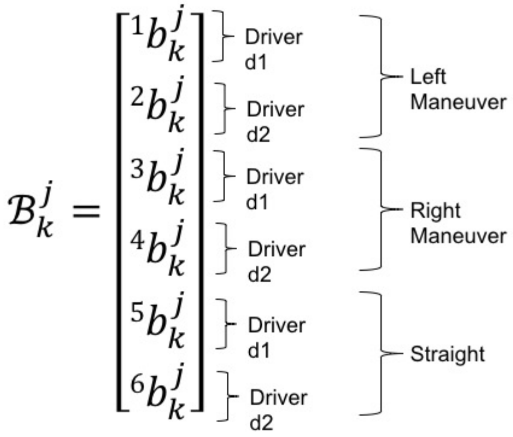

$\mathcal{B}_k^j = \begin{bmatrix} {}^1b_k^j \\ {}^2b_k^j \\ {}^3b_k^j \\ {}^4b_k^j \\ {}^5b_k^j \\ {}^6b_k^j \end{bmatrix}$
Driver d1
Driver d2
Driver d1
Driver d2
Driver d1
Driver d2
Left Maneuver
Right Maneuver
Straight

Figure 4.2: Representation of the behavior set

offline by observations from existing traffic data.

$$\Theta_{i,m} = [\theta_1, \theta_2, ...\theta_{L_m}]$$

where L_m is the total number of parameters for the m^{th} symbolic maneuver in $\mathscr{M}$. Example of maneuver parameters are maneuver length, lateral acceleration, lateral velocity, etc. These can be collectively represented by a set $\mathscr{B}$ which is defined as follows for each traffic participant:

Let $\mathscr{B}_k^j$ be a random variable representing the set of behaviors with the sample space consisting of the ${}^q b_k^j$ elementary elements (Fig. 4.2), where $q \in [1, ..., M \cdot I]$ and k is the time step. Each elementary element consists of the parameter vector $\Theta_{i,m}$. Hence, $\mathscr{B}_k^j$ has three dimensions: driver types, maneuver types and corresponding parameters. If L_m is same for all the maneuvers, $\mathscr{B}$ could be represented by a 3D array. However, for simplicity, it is being structured as a set of $M \cdot I$ vectors as shown in Fig. 4.2. Parameters $\Theta_{i,m}$ are arguments for the state trajectories given by

$$\hat{\mathbf{x}}_{k+1}^j = \mathbf{f}(\mathbf{x}_k^j, b_k^j, t)\Delta t + \mathbf{x}_k^j + \omega_k \tag{4.1}$$

where $X_k = \begin{bmatrix} \mathbf{x}_k^0 & \mathbf{x}_k^1 & ... & \mathbf{x}_k^{J-1} \end{bmatrix}^T$ represent the base (continuous) states for the traffic scenario with $\mathbf{x}_k^j$ representing the base state of the j^{th} traffic participant at k^{th} time step.

Now, the probability distribution of the behavior set for a traffic participant can be given as:

$$\mathbf{p}(\mathscr{B}_k^j) = \begin{bmatrix} \mathbb{P}({}^1 b_k^j) \\ . \\ . \\ . \\ \mathbb{P}({}^{M \cdot I} b_k^j) \end{bmatrix} \tag{4.2}$$

The future behavior of a traffic participant is then completely described by the current associated base states, the maneuver likely to be performed, and the trajectory to be likely executed based upon the driver characteristics. Hence, the core challenge for risk analysis based on behavior prediction for a traffic participant consists of two stages, i.e. for every traffic participant j and driving style i:

1. determine $\mathbb{P}({}^q b_k^j)$, where $q = i + I \cdot (i - 1)$.

2. determine the trajectory $\hat{\mathbf{x}}_{k+1}^{j,i}$.

4.4 Probabilistic framework

The solution to the above challenge can be addressed by utilizing a Dynamic Hierarchical Bayesian Network in the following way. Let $\mathbf{x}_k^j = [x_k^j \ y_k^j \ v_{x,k}^j \ v_{y,k}^j \ \psi_k^j]'$ represent the base states for the j^{th} traffic participant at the k^{th} time step, i.e. longitudinal and lateral positions and velocities as well as the yaw angle. The quantity $\mathbf{z}_k^j = [\tilde{x}_k^j \ \tilde{y}_k^j \ \tilde{v}_{x,k}^j \ \tilde{v}_{y,k}^j \ \tilde{\psi}_k^j]'$ represents the observed states for the j^{th} traffic participant at the k^{th} time step. Note that measurement intervals need not necessarily coincide with integration steps.

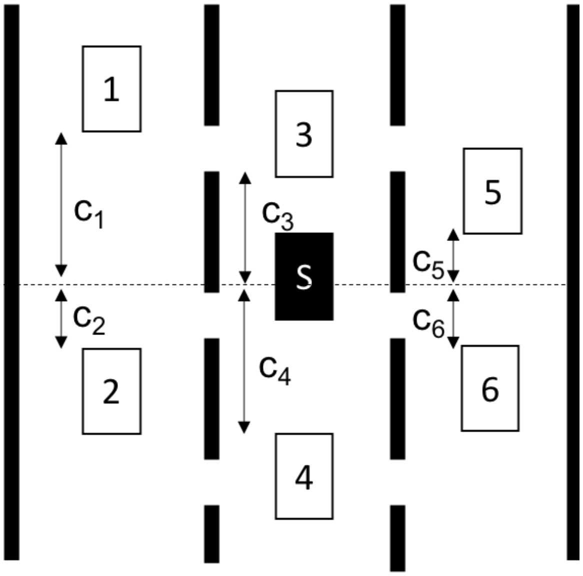


Figure 4.3: Traffic scenario as an example. *S* is the subject vehicle being observed, *1 - 6* are the relevant traffic participants for S, with c_1 - c_6 forming the spatio-temporal relations for the context

To predict the behavior of the traffic participant, we employ the concept of context, as motivated by Gindele et al. [36]. The context vector establishes the necessary spatio-temporal relationships between a subject vehicle and the relevant traffic participants around it for every traffic participant, i.e. C_k^j for the j^{th} traffic participant at the k^{th} time step. Furthermore, let $\mathscr{G}_k^j$ be a binary state at time k indicating that the j^{th} traffic participant has accepted a gap in a traffic scenario for maneuver execution.

Inspired by Gindele et al. [36], we assume certain conditional independence between quantities. The joint probability density function for the traffic scenario can then be decomposed as follows:

$$\begin{aligned}
&\mathbb{P}\big(\hat{X}_{k+1}, \hat{X}_k, C_{k+1}, \mathscr{G}_{k+1}, \mathscr{G}_k, \mathscr{M}_{k+1}, \mathscr{M}_k,\\
&\quad \mathscr{D}_{k+1}, \mathscr{D}_k, \mathscr{B}_{k+1}, \mathscr{B}_k, Z_{k+1}\big)\\
&\quad = \mathbb{P}(\hat{X}_k)\, \mathbb{P}(\mathscr{G}_k)\, \mathbb{P}(\mathscr{M}_k)\, \mathbb{P}(\mathscr{D}_k)\, \mathbb{P}(\mathscr{B}_k)\\
&\qquad \mathbb{P}(\hat{X}_{k+1}|\hat{X}_k, \mathscr{B}_k)\, \mathbb{P}(C_{k+1}|X_{k+1})\\
&\qquad \mathbb{P}(\mathscr{G}_{k+1}|\mathscr{G}_k, C_{k+1}, \mathscr{D}_k)\, \mathbb{P}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1})\\
&\qquad \mathbb{P}(\mathscr{B}_{k+1}|\mathscr{M}_{k+1}, \mathscr{D}_k, \mathscr{B}_k, \mathscr{G}_{k+1})\\
&\qquad \mathbb{P}(Z_{k+1}|\hat{X}_{k+1})\, \mathbb{P}(Z_{k+1}|\mathscr{D}_{k+1}) \qquad (4.3)
\end{aligned}$$

The application of the introduced abstract framework above to a particular traffic scenario will be illustrated via the example of multi-lane driving on a highway. The presented framework and its employed terms, however, are of such a generic nature that they can be easily adapted to other urban as well as highway driving scenarios. For the exemplary application, we assume three maneuvers (left lane change, LL, right lane change, LR, and straight driving, S), two driver types (aggressive, d_1, and passive, d_2) and a context vector based upon the relative longitudinal distances shown in Fig. 4.3. Hence, the introduced sets yield

$$\begin{aligned}
\mathscr{M} &:= [m_{LL}, m_{LR}, m_S], \quad \mathscr{D} := [d_1, d_2]\\
C_k^j &:= [c_1^j\ c_2^j\ c_3^j\ c_4^j\ c_5^j\ c_6^j]
\end{aligned}$$

with the sets $\mathscr{B}_k^j$ each containing 6 behaviors ($M = 3, I = 2$). Further exemplary implementations of the employed terms in the framework are detailed in the following explanatory sections:

Motion prediction, $\mathbf{p}(\hat{X}_{k+1}|\hat{X}_k, \mathscr{B}_k)$: The predicted motion of a traffic participant depends on its current states, the driver type, and the maneuver that the driver intends to perform. The driver type combined with the intended maneuver determine the corresponding parameters for the trajectory generation of the participant as contained in the behavior set $\boldsymbol{B}_k^j$. The identified trajectory can be used as the underlying motion model for propagating the states of every traffic participant.

Context, $\mathbf{p}(C_{k+1}|\hat{X}_{k+1})$: The context experienced by each traffic participant aids in establishing the local traffic situation. In its simplest case, it can be derived directly from the base states by creating a vector reflection the spatio-temporal relations to the neighboring/near-by other traffic participants, for instance distances, velocity differences, etc. Although the context in our example below arises from a linear combination of distances, it can be also include velocity or to any general state relationship that can aid in predicting a driver's intent, without the loss of generality of the probabilistic framework. Since we assume our base states to be Gaussian perturbed, our context vector represents a mixture of Gaussian terms.

Gap acceptance policy, $\mathbf{p}(\mathscr{G}_{k+1}|\mathscr{G}_k, C_{k+1}, \mathscr{D}_k)$: Given the current context around a vehicle, the decision for gap acceptance varies with the driver. In addition to the conditioning on context and driver, we also include a dependence of gap acceptance on the decision in the previous step to include memory. In an exemplary realization, the gap acceptance pdf could be defined as

$$\mathbf{p}(\mathscr{G}_{k+1}^j \mid \mathscr{G}_k^j, \boldsymbol{C}_{k+1}^j, \mathscr{D}_k^j) = \begin{bmatrix} \mathbb{P}(\mathscr{G}_{k+1}^j = 0 \mid \mathscr{G}_k^j, \boldsymbol{C}_{k+1}^j, \mathscr{D}_k^j) \\ \mathbb{P}(\mathscr{G}_{k+1}^j = 1 \mid \mathscr{G}_k^j, \boldsymbol{C}_{k+1}^j, \mathscr{D}_k^j) \end{bmatrix} \tag{4.4}$$

where $\mathscr{G}_k^j = 0$ denotes gap rejection for participant j while $\mathscr{G}_k^j = 1$ stands for gap acceptance. The terms on the right hand side of Eq. (4.4) can be further augmented as

$$
\begin{aligned}
&\mathbf{p}(\mathscr{G}^j_{k+1} \mid \mathscr{G}^j_k, \boldsymbol{C}^j_{k+1}, \mathscr{D}^j_k) \\
&\quad\propto \Big[\mathbb{P}(\mathscr{D}^j_k = d_1) \cdot \mathbf{p}\big(\mathscr{G}^j_{k+1} | \mathscr{G}^j_k, \boldsymbol{C}^j_{k+1}, \mathscr{D}^j_k = d_1\big) \\
&\qquad + \mathbb{P}(\mathscr{D}^j_k = d_2) \cdot \mathbf{p}\big(\mathscr{G}^j_{k+1} | \mathscr{G}^j_k, \boldsymbol{C}^j_{k+1}, \mathscr{D}^j_k = d_2\big)\Big]
\end{aligned}
\tag{4.5}
$$

The first conditional probability on the right hand side of Eq. (8.7) can be expanded as

$$
\mathbf{p}\big(\mathscr{G}^j_{k+1} | \mathscr{G}^j_k, \boldsymbol{C}^j_{k+1}, \mathscr{D}^j_k = d_1\big) = \begin{bmatrix} 1 - g_{ag}(\boldsymbol{C}^j_{k+1}) \\ g_{ag}(\boldsymbol{C}^j_{k+1}) \end{bmatrix}
\tag{4.6}
$$

where $g_{ag}(\cdot)$ is a gap function associated with an aggressive driver. Here, the context for the specific traffic participant is accepted as an input, returning an associated probability of gap acceptance. This could, for instance, be realized via a step function, i.e.

$$
g_{ag}(\boldsymbol{C}^j_{k+1}) = \begin{cases} 0 & (c_1 + c_2) < \tau_{ag} \wedge \; (c_5 + c_6) < \tau_{ag} \\ 1 & \text{otherwise} \end{cases}
\tag{4.7}
$$

Here, τ_{ag} is a distance threshold for an aggressive driver for gap acceptance. To allow for a more gradual decision transition, a logistic function could be utilized alternatively, yielding

$$
g_{ag}(\boldsymbol{C}^j_{k+1}) = \frac{1}{1 + e^{-\zeta(gap - \tau_{ag})}}
$$

with ζ being a tuning factor and with gap being a placeholder for any of the relevant observed gaps. The second conditional probability on the right hand side of Eq. (4.5) can be treated in a similar fashion by using the corresponding gap functions for a passive driver. It should be noted that these are only exemplary realization of the introduced gap functions only utilizing distance relations in the context. A more sophisticated gap processing based upon relative velocities (or other relations) could certainly be employed without loss of generality. The context vector would require modifications according to this case.

Maneuver policy, $\mathbf{p}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1})$: The maneuver policy can formally be expanded similar to the gap acceptance policy, i.e.

$$\begin{aligned}&\mathbf{p}\big(\mathscr{M}^j_{k+1} \mid \mathscr{M}^j_k, \mathscr{G}^j_{k+1}, C^j_{k+1}\big)\\ &\quad\propto \Big[\mathbb{P}(\mathscr{G}^j_{k+1} = 0) \cdot \mathbf{p}\big(\mathscr{M}^j_{k+1}|\mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 0\big)\\ &\qquad+\mathbb{P}(\mathscr{G}^j_{k+1} = 1) \cdot \mathbf{p}\big(\mathscr{M}^j_{k+1}|\mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 1\big)\Big]\end{aligned} \tag{4.8}$$

with $\mathscr{G}^j_{k+1} = 0$ signifying that the driver did not accept the gap. Here, no maneuver other than a nominal driving maneuver (for instance defined as straight) will be executed, irrespective of the driver type. Hence, the conditional probability for this case in Eq. (4.8) reduces to

$$\mathbf{p}\Big(\mathscr{M}^j_{k+1} \mid \mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 0\Big) = \begin{bmatrix}\mathbb{P}(m_{LL})\\ \mathbb{P}(m_{LR})\\ \mathbb{P}(m_S)\end{bmatrix} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

Similarly, the gap acceptance term in Eq. (4.8) can be realized as

$$\mathbf{p}\Big(\mathscr{M}^j_{k+1} \mid \mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 1\Big) = \begin{bmatrix}f_{LL}(C^j_{k+1})\\ f_{LR}(C^j_{k+1})\\ 0\end{bmatrix}$$

where $f_{(.)}(\cdot)$ is a maneuver function establishing a relationship between the context and the left and right lane maneuver, respectively. A simple exemplary implementation of such a function yields

$$f_{LL}(C_{k+1}) = \frac{c_1 + c_2}{(c_1 + c_2) + (c_5 + c_6)} \tag{4.9}$$

For countries with asymmetric lane changes, another parameter could be included in the maneuver function reflecting different weights for left and right. To ensure that the model does not switch instantaneously from one maneuver to another during execution, a history factor can be included into the determination of the maneuver policy via weights given as

$$\begin{aligned} w_1 &= 1 - \alpha e^{-\beta \Delta t} \\ w_2 &= \alpha e^{-\beta \Delta t} \end{aligned} \tag{4.10}$$

where α and β are the tuning parameters, and Δt is the sampling interval. The maneuver policy now takes the weighted average of past decisions into account, i.e.

$$\begin{aligned} &\mathbf{p}\big(\mathscr{M}_{k+1} |\, \mathscr{M}_k^j, C_{k+1}^j, \mathscr{G}_{k+1}^j\big) \\ &\propto w_1 \begin{bmatrix} \mathbb{P}(m_{LL(k+1)}|...) \\ \mathbb{P}(m_{RL(k+1)}|...) \\ \mathbb{P}(m_{S(k+1)}|...) \end{bmatrix} + w_2 \begin{bmatrix} \mathbb{P}(m_{LL(k)}|...) \\ \mathbb{P}(m_{RL(k)}|...) \\ \mathbb{P}(m_{S(k)}|...) \end{bmatrix} \end{aligned} \tag{4.11}$$

As the framework already factors the driver influence into the gap acceptance policy, the maneuver policy does not depend on the driver directly. Yet, this does not prohibit the consideration of the driver's influence in a maneuver decision explicitly if such an extension of the policy is desired in the future.

Behavior, $\mathbf{p}(\mathscr{B}_{k+1}|\mathscr{M}_{k+1}, \mathscr{D}_k, \mathscr{B}_k)$: As we assume that the maneuver and driver types can evolve independently for a specific traffic participant, they are addressed separately instead of jointly in the behavior space. This allows for considering their probability transitions separately in update and propagation, yielding

$$\mathbb{P}({}^{q}b^{j}_{k+1}) = \frac{\mathbb{P}({}^{i}d^{j}_{k}) \cdot \mathbb{P}({}^{m}m^{j}_{k+1})}{\sum_{q=1}^{M \cdot I} \mathbb{P}({}^{i}d^{j}_{k}) \cdot \mathbb{P}({}^{m}m^{j}_{k+1})} \tag{4.12}$$

where $i = (q \bmod I + 1)$ indicates the driver index and $m = \left\lceil \frac{q}{I} \right\rceil$ the maneuver index in the set $\mathscr{B}^{j}_{k+1}$.

Likelihood, $\mathbf{p}(Z_{k+1}|\hat{X}_{k+1})$: For instance, standard Kalman Filter update expressions can be utilized to update the estimates once the measurements are available. However, any other sophisticated update relation is possible.

Driver policy update, $\mathbf{p}(Z_{k+1}|\mathscr{D}_{k+1})$: In [34], we have presented an approach for maneuver identification of a traffic participant using Multiple Model Adaptive Estimation (MMAE). A bank of Extended Kalman Filters (EKFs) has been employed in a classic MMAE configuration to differentiate between left lane change, right lane change, and straight maneuver. The very same approach can be directly adapted for utilization in the framework of this study. For every traffic participant, a bank of $M \cdot I$ EKFs can be employed in either the classic MMAE or the Interactive Multiple Model (IMM) filter setting to propagate trajectories and to estimate the likelihood of a particular behavior. The probabilistic driver type itself is a parameter in the propagation in between measurement intervals. It will be updated based upon the likelihood of the measurements for the employed behavior in the individual single model filters. The driver model probability then yields

$$\mathbb{P}(d_1) = \frac{\mathbb{P}({}^{1}b^{j}_{k}) + \mathbb{P}({}^{3}b^{j}_{k}) + \mathbb{P}({}^{5}b^{j}_{k})}{\sum_{q=1}^{M \cdot I} \mathbb{P}({}^{q}b^{j}_{k})} \tag{4.13}$$

4.5 Filter equation

The entire filter equation for the presented approach becomes:

$$\begin{aligned}&\mathbb{P}(\hat{X}_{k+1}, C_{k+1} | Z_{k+1})\\ &\quad = \mathbb{P}(Z_{k+1}|\hat{X}_{k+1})\mathbb{P}(Z_{k+1}|\mathscr{D}_{k+1}) \int_{\mathscr{G}_k,\mathscr{M}_k,\mathscr{D}_k,\mathscr{B}_k} \mathbb{P}(\hat{X}_{k+1}, C_{k+1}, \mathscr{G}_{k+1}, \mathscr{M}_{k+1}, \mathscr{D}_{k+1}\\ &\qquad\qquad , \mathscr{B}_{k+1}|\mathscr{G}_k, \mathscr{M}_k, \mathscr{D}_k, \mathscr{B}_k, \hat{X}_k)\mathbb{P}(\mathscr{G}_k, \mathscr{M}_k, \mathscr{D}_k, \mathscr{B}_k, \hat{X}_k) \qquad (4.14)\end{aligned}$$

While various approaches based on Monte Carlo simulation can be employed for inference, their limitation however is that the resulting posterior for the states is not Gaussian. Our approach with Multi Model Adaptive Estimation results in a posterior that is approximated with a mixture of Gaussians and hence, can be represented with only two parameters - the mean and the variance.

If we fit this Bayesian structure to the system architecture shown in Figure 4.1, the probability density function (pdf) of the prior positions, $\mathbf{p}(\hat{X}_k)$, is provided collectively by the perception and the localization modules. The scene identification sub-module in this block uses the available digital maps and/or the prior information to identify the traffic scene. The situation classification then utilizes this information for the derivation of the context information using helper functions (for details about helper functions refer to [36, 10]) and in turn employs the context, $\mathbf{p}(C_{k+1}|X_{k+1})$, to classify the situation. Based on the traffic policies defined within it, the trajectory prediction sub-module, supplies the conditional pdf of the gap acceptance, $\mathbf{p}(\mathscr{G}_{k+1}|\mathscr{G}_k, C_{k+1}, \mathscr{D}_k)$, maneuver intent, $\mathbf{p}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1})$, and the possible behaviors for each traffic participant, $\mathbf{p}(\mathscr{B}_{k+1}|\mathscr{M}_{k+1}, \mathscr{D}_k, \mathscr{B}_k)$. It then identifies likely trajectories to be executed in order to realize the behaviors, $\mathbf{p}(\hat{X}_{k+1}|\hat{X}_k, \mathscr{B}_k)$. The risk evaluation sub-module then incorporates these predicted trajectories for all traffic participants to predict possible risks as per the identified metrics.

4.6 Conclusion

When sharing a traffic environment with human drivers, autonomous vehicles must incorporate driving style identification for traffic participant similar to human performance. Differentiating between aggressive, passive, cooperative drivers, for instance, aids in predicting the behavior of other traffic participants for a longer time horizon, and supports the planning and decision making for autonomous cars. In this chapter, a probabilistic framework incorporating driving style characterization for an observed traffic participant is presented. The framework takes four stages into account usually observed in human maneuver performance: intent determination, maneuver preparation, gap acceptance, and maneuver execution. In addition to constructing an abstract framework, this study includes basic suggestions for implementation of intent determination, gap acceptance and maneuver execution stages. Yet, the modularity of the framework allows for an effortless expandability. The remaining chapters will focus on implementing this framework using Multiple Model Adaptive Estimation based approaches.

Chapter 5

Maneuver Identification and Prediction for Traffic Participants

Previous chapter described an approach to model the traffic participants using a Dynamic Hierarchical Bayesian Network. This chapter seeks to address one of the key challenges to build the framework: identifying an approach to infer the higher level behaviors from the available low level measurements from the traffic participant. It summarizes a Multiple Model Adaptive Estimation (MMAE) based approach called Autonomous Multiple Model (AMM) filtering and presents the results of process and measurement noise analysis on inferring the behaviors from the observations. Since only one behavior for a maneuver is considered in this evaluation, this chapter uses the terms behavior and maneuvers interchangeably.

5.1 Maneuver identification model

5.1.1 Autonomous Multiple Model estimation approach

The presence of both discrete as well as continuous states in the autonomous vehicle framework described above yields a hybrid system. MMAE methods are a popular technique to cope with hybrid state estimation. For a comprehensive review of MMAE based approaches, the reader is referred to[49, 24]. Here, MMAE is also categorized into 3 generations: the Autonomous Multiple Model (AMM, also called as the classic MMAE in literature), the Interacting Multiple Model (IMM) approach, and variable structure MMAE . The IMM technique is adopted the most [49] for a variety of applications and as it exhibits a simpler design in comparison to variable structure approaches. Both AMM and IMM employ a bank of Kalman filters, each reflecting a different motion and/or measurement model. Then, a combined state estimate is calculated as a weighted sum of every individual filter result. IMM estimation, however, includes an additional stage called interaction in which the most recent estimates from all individual filters are mixed according to their predicted probabilities and then set as initial values for the next cycle. It is this interaction stage that renders the execution of IMM filter banks dependent on each other in between every cycle. As a result, the models in IMM can only run in parallel within a prediction step, but not across multiple prediction cycles. This is in contrast to AMM where all single model filters run independent across all cycles. This enables the full utilization of recent advances in parallel computing technology.

One recent study by Xie et al. also utilizes a multi model approach for prediction. Here, the authors employ an IMM to differentiate between long and near term trajectory prediction for the same maneuver. Our study differs in two ways: first, we employ the Autonomous Multiple Model approach (AMM), also called as a classic MMAE [25], for trajectory prediction; second, each single-model filter corresponds to a different maneuver.

With the recent advances in computing and sensor technologies, and due to a different set of requirements for prediction applications in comparison to tracking applications, a careful re-evaluation of MMAE is necessary. For the scope of this study, the MMAE approach is applied to only one observed vehicle. Expanding to multiple vehicle will be addressed in future work. Classic MMAE, or AMM, consists of three stages [24]:

1. *Model specific filtering:* The expressions for prediction and update for each single-model filter resemble the ones employed in the Extended Kalman filter (EKF) [24]. Different model assumptions for each filter will be discussed in the next section.

2. *Model probability update:* This stage computes the likelihood of the measurements for every single-model filter estimate, and subsequently determines the associated weights for each filter. For M filters, the initial weight for every filter is uniform distributed, i.e. $w_0^q = 1/M$ for q = 1, 2, ... M. The update relations then yield

$$w_k^q = w_{k-1}^q \mathbf{p}(\mathbf{z}_k|\mathbf{x}_k^{-q}) \tag{5.1}$$

$$w_k^q \leftarrow \frac{w_k^q}{\sum_{q=1}^{M} w_k^q} \tag{5.2}$$

$$p(\mathbf{z}_k|\mathbf{x}_k^{-q}) = \frac{1}{\left|2\pi E_k^{-q}\right|^{1/2}} e^{\left\{-\frac{1}{2}\mathbf{e}_k^{-qT}(E_k^{-q})^{-1}\mathbf{e}_k^{-q}\right\}} \tag{5.3}$$

$$E_k^{-q} = E\left\{e_k^{-q}e_k^{-qT}\right\} = H_k^q P_k^{-q} H_k^{qT} + R_k^q \tag{5.4}$$

3. *Combination:* The estimates and covariances from all single-model filters are fused utilizing their associated weights to provide a combined estimate, i.e.

$$\hat{\mathbf{x}}_k^+ = \sum_{q=1}^{M} w_k^q \hat{\mathbf{x}}_k^{+q} \tag{5.5}$$

$$P_k^+ = \sum_{q=1}^{M} w_k^q \left[\left(\hat{\mathbf{x}}_k^{+q} - \hat{\mathbf{x}}_k^+\right)\left(\hat{\mathbf{x}}_k^{+q} - \hat{\mathbf{x}}_k^+\right)^T + P_k^{+q}\right] . \tag{5.6}$$

The AMM process is shown in Fig. 5.1, which is inspired from [25]. In the figure, $\mathbf{e}_k^{-q}$ is the vector of measurement residuals, $\mathbf{z}_k$ is the measurement vector and $\hat{\mathbf{x}}_k^{+q}$ is the estimated state vector for the q^{th} elemental filter, where $q \in [1, M.I]$.

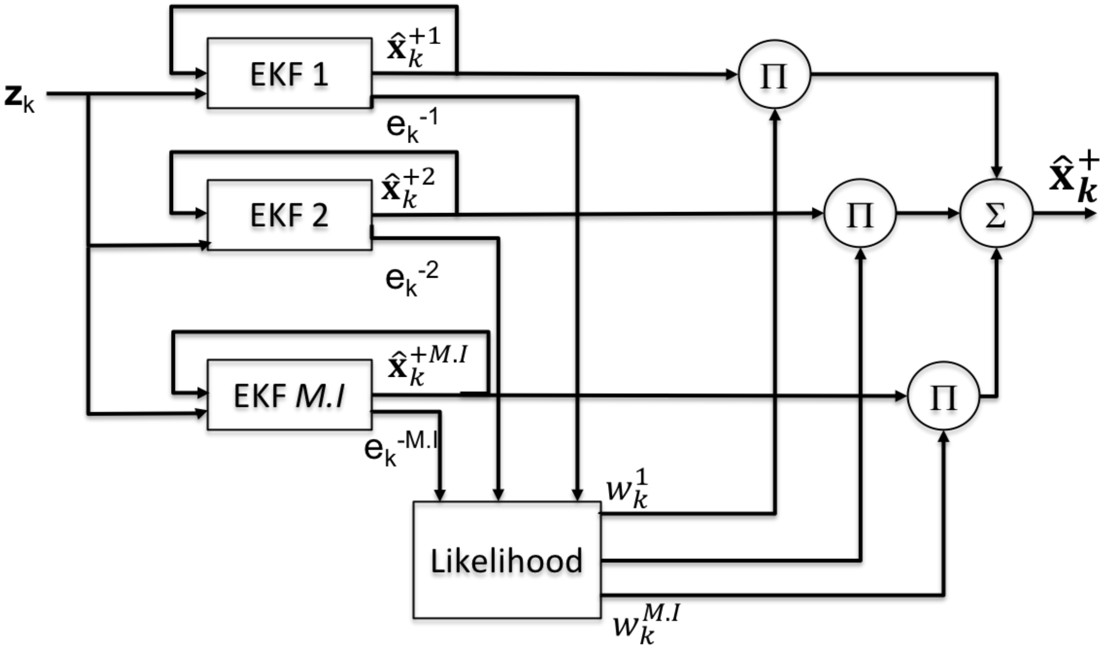


Figure 5.1: Autonomous Multiple Model filter block diagram

5.1.2 Motion models and measurement model

In the presented study, MMAE is applied to distinguish three maneuvers: straight motion, left lane change and right lane change. Straight motion assumes a constant velocity model (section 3.1.1) whereas the lane changes are modeled as sinusoidal trajectories with the maneuver initiation point and maneuver length as parameters (as per section 3.1.5.1). A bank of three filters runs in parallel: one linear Kalman filter for straight motion and two EKFs for lane changes. The states of a traffic participant are represented by $\mathbf{x} = [x, v_x, y, v_y]'$ with x and y being the longitudinal and lateral position of the vehicle in the road coordinate frame, and v_x and v_y being the respective velocities.

Model equations for straight motion: For the straight motion, the discrete-time state transition for a single step $k+1$ with sampling time T_s results as

$$\begin{bmatrix} x_{k+1} \\ v_{x(k+1)} \\ y_{k+1} \\ v_{y(k+1)} \end{bmatrix} = \begin{bmatrix} 1 & Ts & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Ts \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_{x(k)} \\ y_k \\ v_{y(k)} \end{bmatrix} \tag{5.7}$$

Model equations for lane change maneuvers: For a right lane change with the lane width w_L, length of the maneuver L and longitudinal distance of the vehicle from the starting point of the maneuver Δx (Fig. 5.2), the employed sinusoidal motion yields

$$y(\Delta x) = \frac{w_L}{2} cos\left(\frac{\pi}{L}\Delta x\right) \tag{5.8}$$

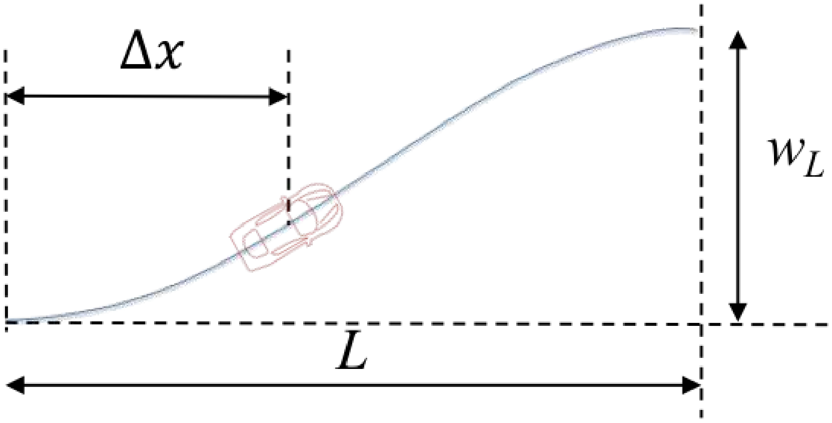


Figure 5.2: Sinusoidal lane change parameters

For simplicity, we assume that the vehicle starts one particular maneuver with a constant velocity at the initial position. The discrete-time equations of motion can then be approximated as

$$x_{k+1} = x_k + v_{xk} T_s \tag{5.9}$$

$$v_{x(k+1)} = v_{xk} \tag{5.10}$$

$$y_{k+1} = y_k + v_{yk} T_s \tag{5.11}$$

$$v_{y(k+1)} = -\frac{w_L \pi v_x}{2L} sin\left(\frac{\pi}{L} x_{k+1}\right) \ . \tag{5.12}$$

The Jacobian matrix for the right lane change then yields

$$\Phi = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Ts \\ -\frac{w_L}{2}\left(\frac{\pi}{L}\right)^2 v_{xk} cos\left(\frac{\pi}{L} x_k\right) & -\frac{w_L \pi}{2L} cos\left(\frac{\pi}{L} x_k\right) & 0 & 0 \end{bmatrix} \tag{5.13}$$

Consequently, the equations for the left lane change can be described by a phase shifted right lane maneuver

$$y\left(\Delta x\right) = \frac{w_L}{2} cos\left(\frac{\pi}{L}\Delta x - \pi\right) \ . \tag{5.14}$$

Measurement model: Assuming a hybrid sensor fusion architecture[54], the measurement model is chosen as linear, with vehicle positions directly observable, i.e.

$$\begin{bmatrix} z_x \\ z_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ v_{xk} \\ y_k \\ v_{yk} \end{bmatrix} \tag{5.15}$$

5.2 Vehicle dynamics model

A single track nonlinear vehicle dynamics model is employed to evaluate the behavior identification model. This model approximates the vehicle loads by considering only a lumped front and rear axle, ignoring roll and pitch motions. The resulting two translational and one rotational degree of freedoms are given by

$$\dot{v}_x - v_y\dot{\psi} = \frac{1}{m}\left(F_{x,f}cos(\delta) + F_{x,r} - F_{y,f}sin(\delta)\right) = \frac{F_X}{m} \tag{5.16}$$

$$\dot{v}_y + v_x\dot{\psi} = \frac{1}{m}\left(F_{y,f}cos(\delta) + F_{y,r} - F_{x,f}sin(\delta)\right) = \frac{F_Y}{m} \tag{5.17}$$

$$I_{zz}\ddot{\psi} = (l_f F_{y,f}cos(\delta) - l_r F_{y,r} + F_{x,f}sin(\delta) = M_z \tag{5.18}$$

where m is the mass of the vehicle, v_x and v_y are the longitudinal and lateral velocities, ψ is the yaw, and δ the steering angle of the vehicle. $F_{x,i}$ and $F_{y,i}$, with $i = [f, r]$, are the net longitudinal and lateral forces acting on the front and rear axle whereas l_f and l_r correspond to the distances of the axles from the center of gravity, and I_z is the vehicle inertia. For more details on the parameter values selected, the reader is referred to [18]. Measurements fed to the behavior model were synthetically generated by adding delta-correlated (white) noise to the values resulting from the vehicle model.

5.3 Tuning and evaluation of the behavior identification model

The simulation part of this study consists of two stages. At first, different cases of process and measurement noises are considered, for the ground truth generated from the motion models, to understand the variations that the approach can tolerate (Table 5.1) and to support filter tuning. Next, the approach is evaluated with the single track vehicle model. Disturbances were modeled as additive white noise.

5.3.1 Process and measurement noise analysis

In this stage of simulations, the measurements were generated from the respective motion model of the maneuver. The initial conditions for both measurements and filter models were set identical to $x = 0$ m, $y = 0$ m, $v_x = 10$ m/s, $v_y = 0$ m/s with initial covariances of $P_0 = 0.0025$, thus, high confidence in the initial estimates. The maneuver

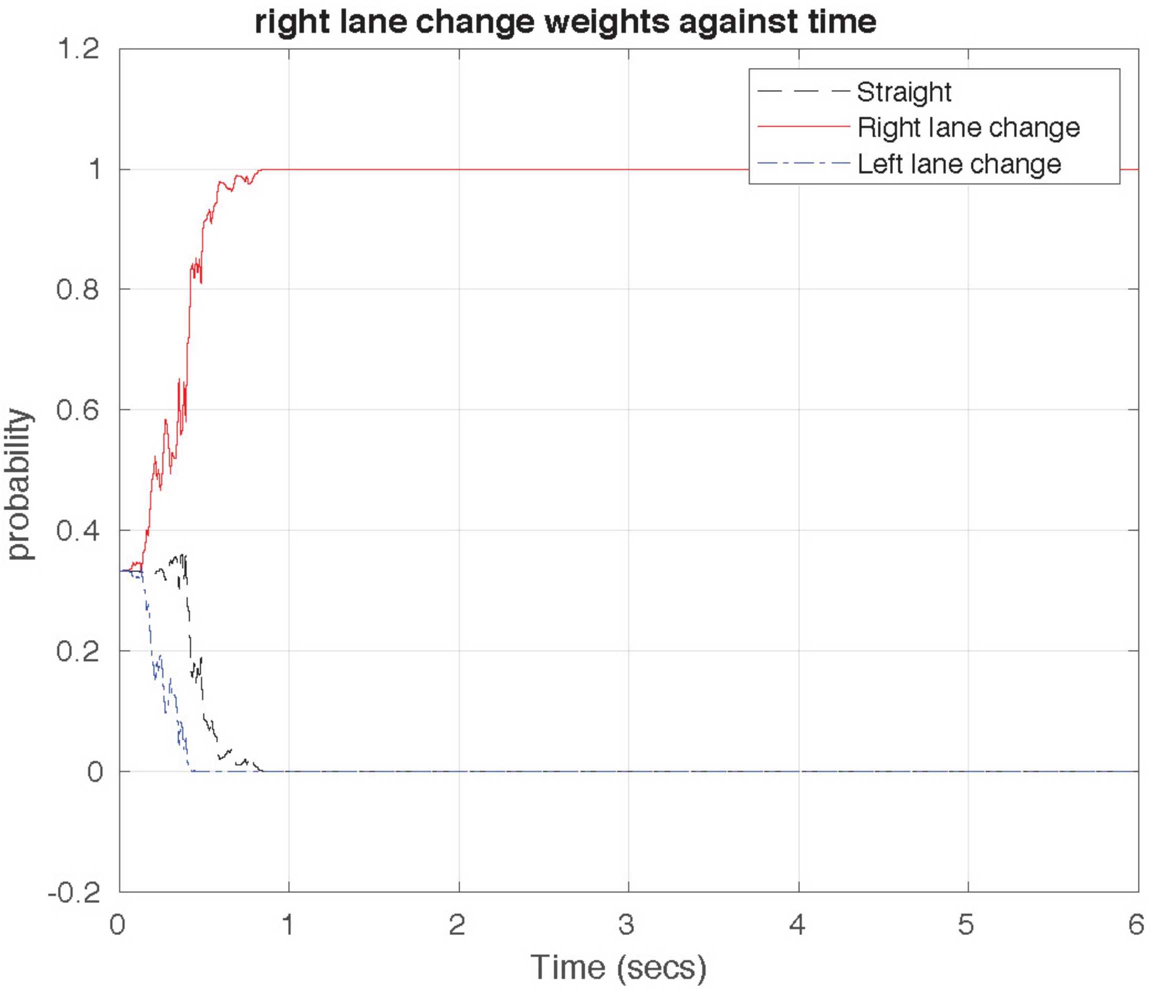


Figure 5.3: Right lane change weights for case 1

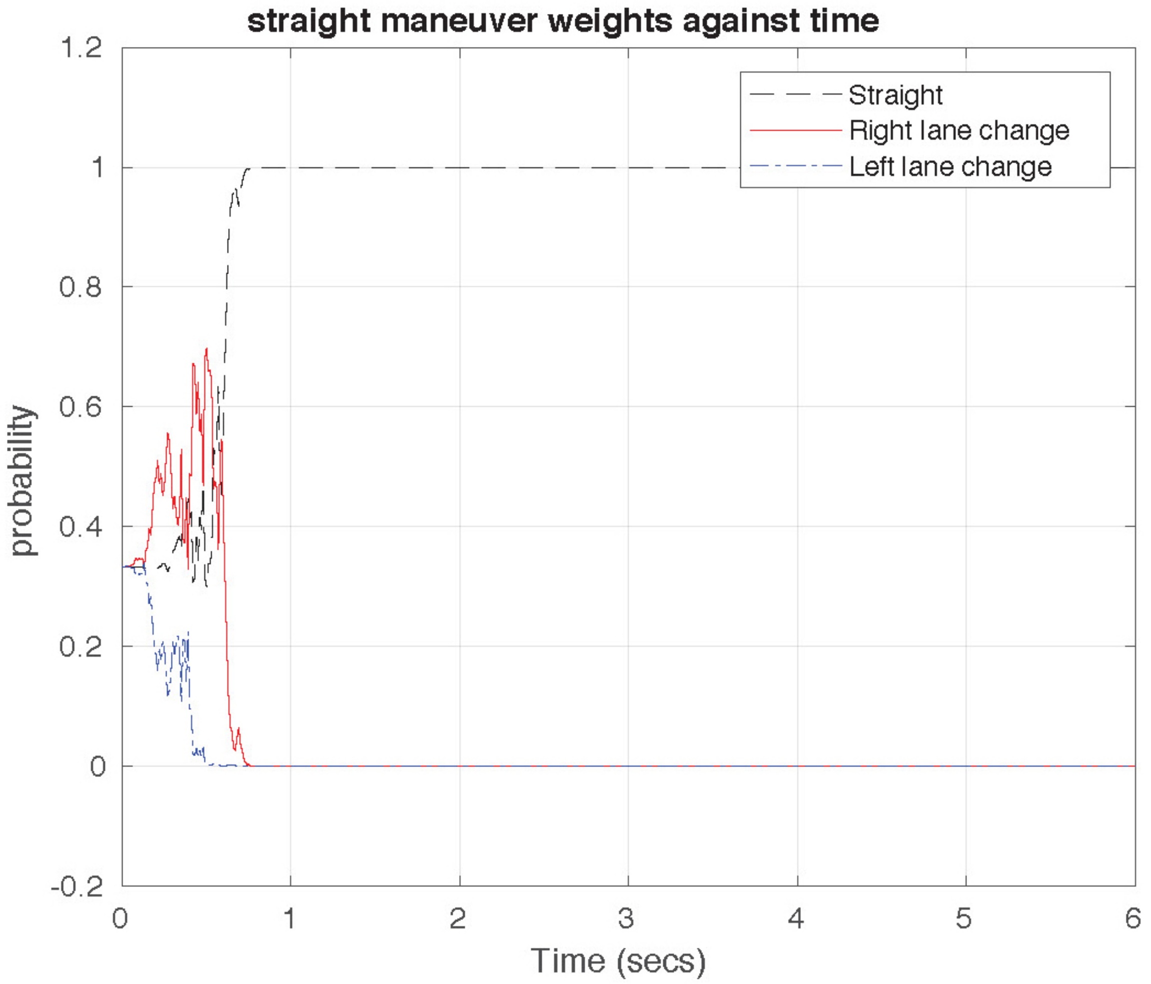


Figure 5.4: Straight maneuver weights for case 1

lengths for left and right lane changes were configured to 60 m with a maneuver execution duration of about 6 seconds. The times at which the weights cross 90% and remain above are captured as detection times.

Non surprisingly, a noisy process increases maneuver detection times (see Table 5.1). Figures 5.3 and 5.4 depict the weight evolution for right lane change (LC) and straight motion of case. For all other cases, some switching behavior between LC and straight maneuvers can be observed before detection (Figure 5.5 for case 3) which is discussed below. The right LC version of cases 4 and 5 demonstrates continued switching in detection without

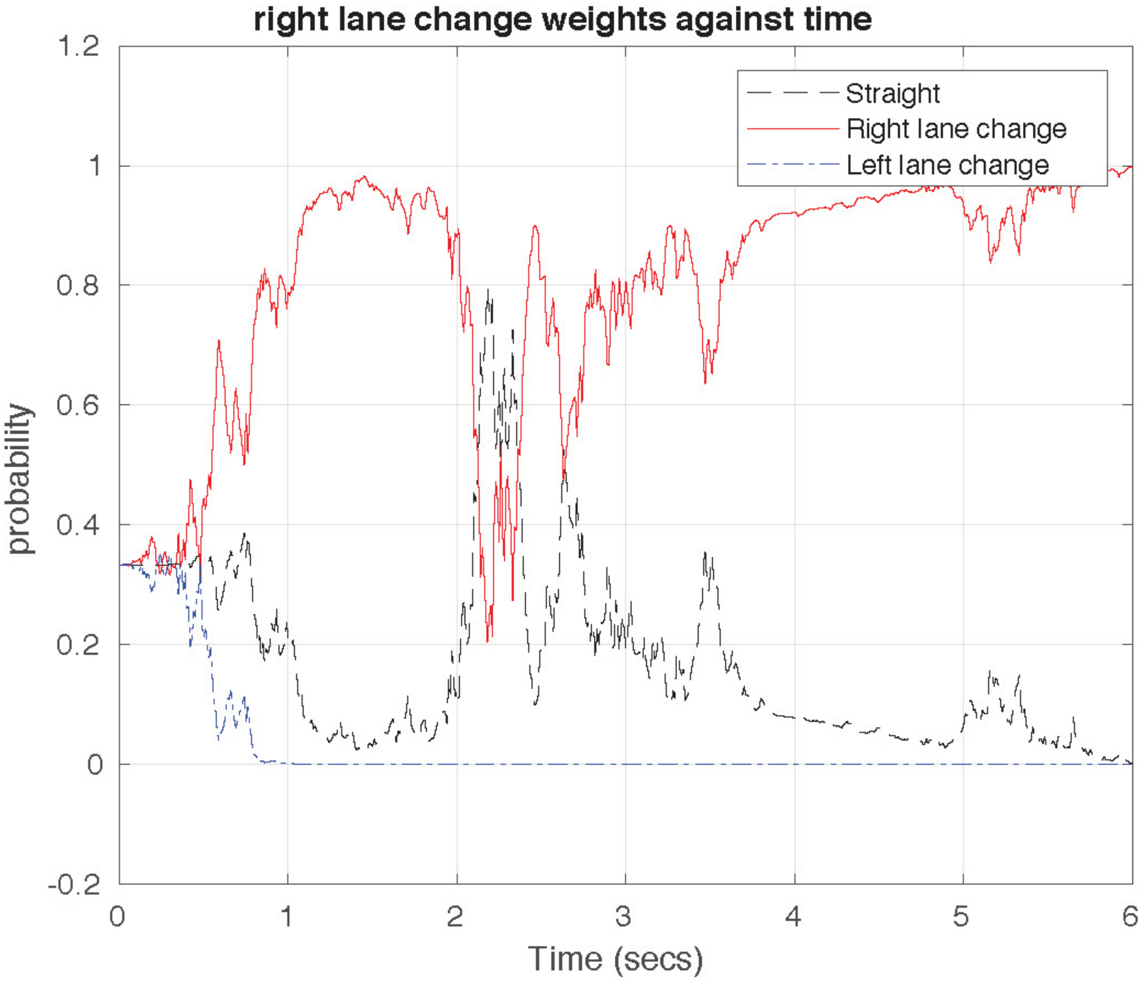


Figure 5.5: Right lane change weights for case 3

convergence and has been represented as 'inf' in table 5.1. For any given process noise, increasing the measurement noise level showed marginal effect on detection times despite adding fluctuations in the probabilistic weights. Interestingly, there is an asymmetry in the detection times for left and right LC. The cause for this is yet to be investigated. These results are obtained under the assumption that the initial states of the observed vehicle match the initial conditions of the filter banks. For the kinematic states, this is a valid assumption. However, if the motion model includes maneuver specific states that have significant uncertainty at the initiation of maneuver this is a strong assumption. An example of this is

Table 5.1: Results of varying the process and measurement noise for MMAE

			Maneuver detection time (s)		
Case No	Q(diagonal)	R(diagonal)	Straight	Left lane change	Right lane change
1	0.001	0.0025	0.7	0.7	0.5
2	0.01	0.0025	1	1.3	2.3
3	0.025	0.0025	1.3	1.5	3.8
4	0.05	0.0025	1.7	1.7	inf
5	0.1	0.0025	2.2	3.6	inf
6	0.025	0.025	1.6	1.6	4.2
7	0.025	0.25	1.6	1.6	inf

maneuver length for a lane change, which becomes more apparent as the participant progresses into the maneuver. In such cases, the initial covariances might need to modified a relatively higher setting, signifying a lower confidence in the initial estimate. This will be discussed in the next chapter.

5.3.2 Evaluation with the single track vehicle model

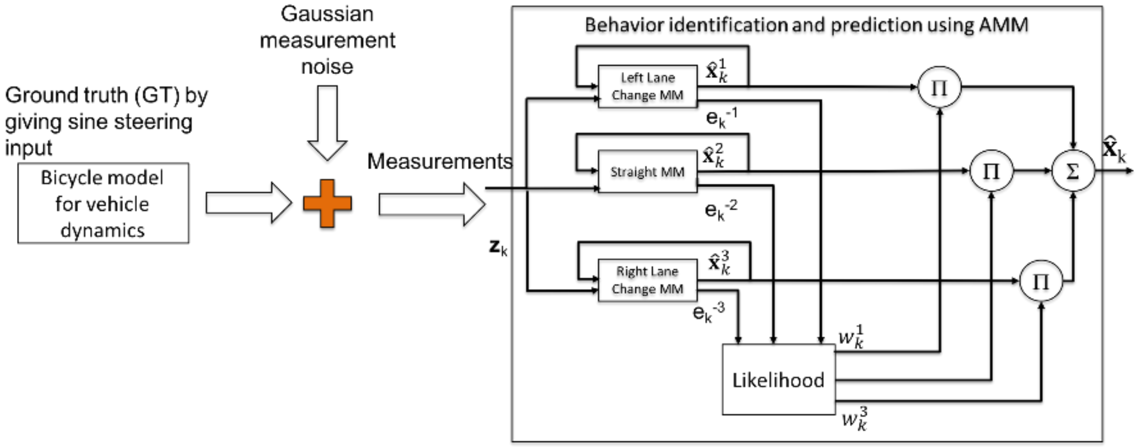


Figure 5.6: Evaluation of the maneuver identification model with a bicycle model

In the second simulation stage, the MMAE behavior model is evaluated using the single track model with vehicle parameters from [18]. Fig. 5.6 summarizes the evaluation process. The straight maneuver is created by maintaining a zero steering wheel angle for the vehicle model whereas the left and right lane change maneuvers result from a sinusoidal input to the steering wheel of 10 second period length, therefore leading to a 60 meter maneuver. Measurements were generated by adding Gaussian noise with parameters described in table 5.2. The individual filters in the MMAE behavior model assume a fixed length of 60 meters for left and right lane change.

The detection times for all maneuvers under different process and noise parameters are summarized in table 5.2. For cases 1 and 2, the model tends to switch between the respective lane change and a straight maneuver for the first few seconds before convergence. A likely explanation for these result is the intitial delay between a sinusoidal steering wheel input and the (non-sinusoidal) path followed by the vehicle at first. As a result, the innovation equations of the lane change maneuver and the straight maneuver initially compete with each other reflected in the switching mode. However, this can be addressed by including this discrepancy in the process noise. Figures 5.7 and 5.8 depict the results for case 6. Even though the detection times in this case is slightly increased, the weight transients follow a gradual and unambiguous trend after divergence from the initial condition. Under straight driving conditions, as the weights for left and right lane change maneuver are identical and cancel in the combined estimate, a consistent net straight maneuver identification throughout the entire simulation interval can be observed. Under the right lane change, after diverging from the initial conditions, the weights for the right as well as the straight maneuvers rise together for some time, before the right starts dominating. The resulting combined estimate leads to a net right maneuver identification after 1 second into the maneuver execution, making this case superior to the other cases. Realistic process noise should associate modeling errors primarily with velocities as they vary between different driver. Reflecting this expectation by attributing more process noise to velocities instead of positions yields even improved results as shown for cases 8 and 9.

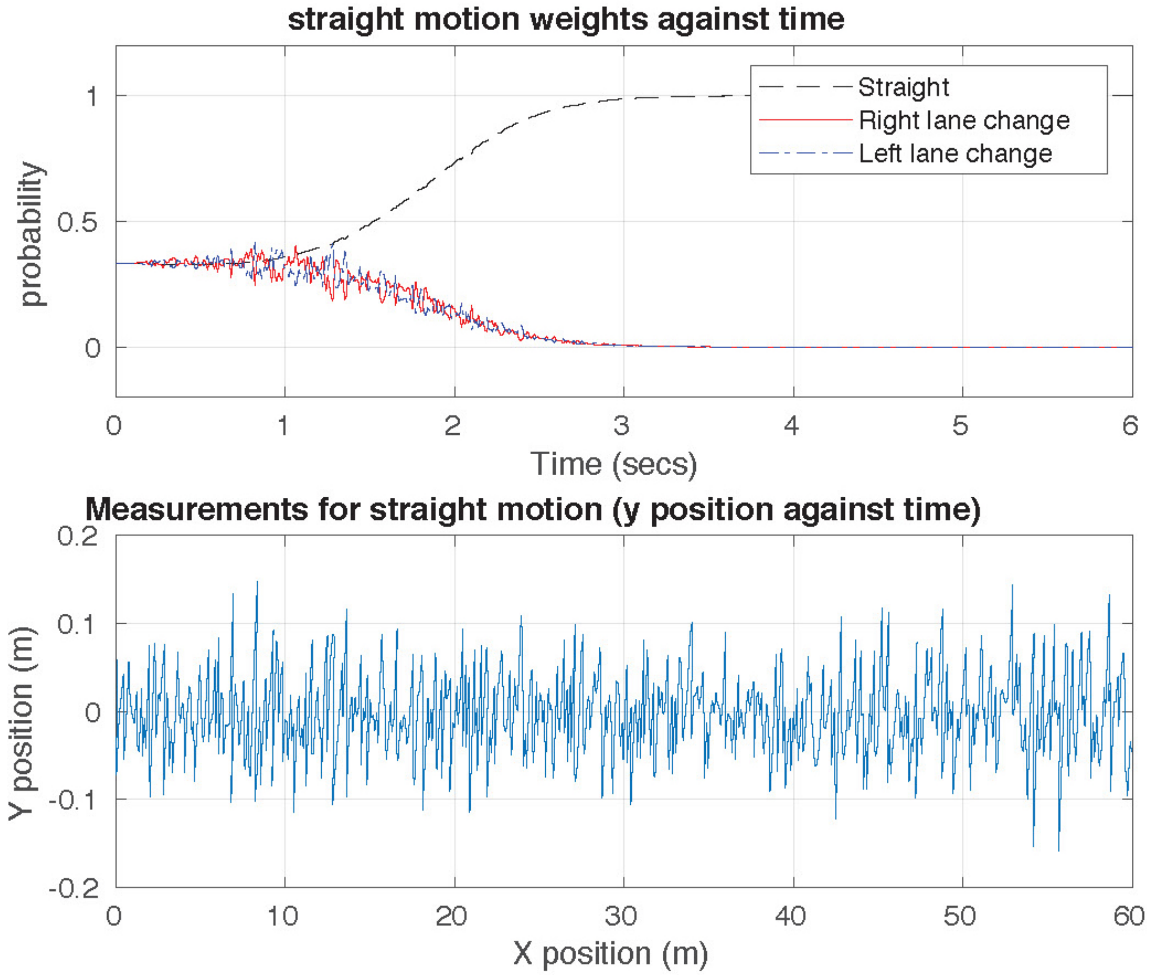


Figure 5.7: Straight maneuver detection when Q=0.1 and R=0.0025 with vehicle model

Note that there is an unusual sensitivity associated with the proper selection of initial covariance parameters for the employed individual filters. There is not too much emphasis in literature on the initial state estimate and associated covariance for the Kalman filter due to its rather quick convergence property. The distinguishing feature in the presented application, however, is the utilization of the elementary filters for classification of maneuvers. In general, two or more elementary filters in MMAE can have equally high confidence in their estimate (reflected by small estimation error covariances) despite employing two fundamentally different underlying system models. This can be observed if uncertainty in the

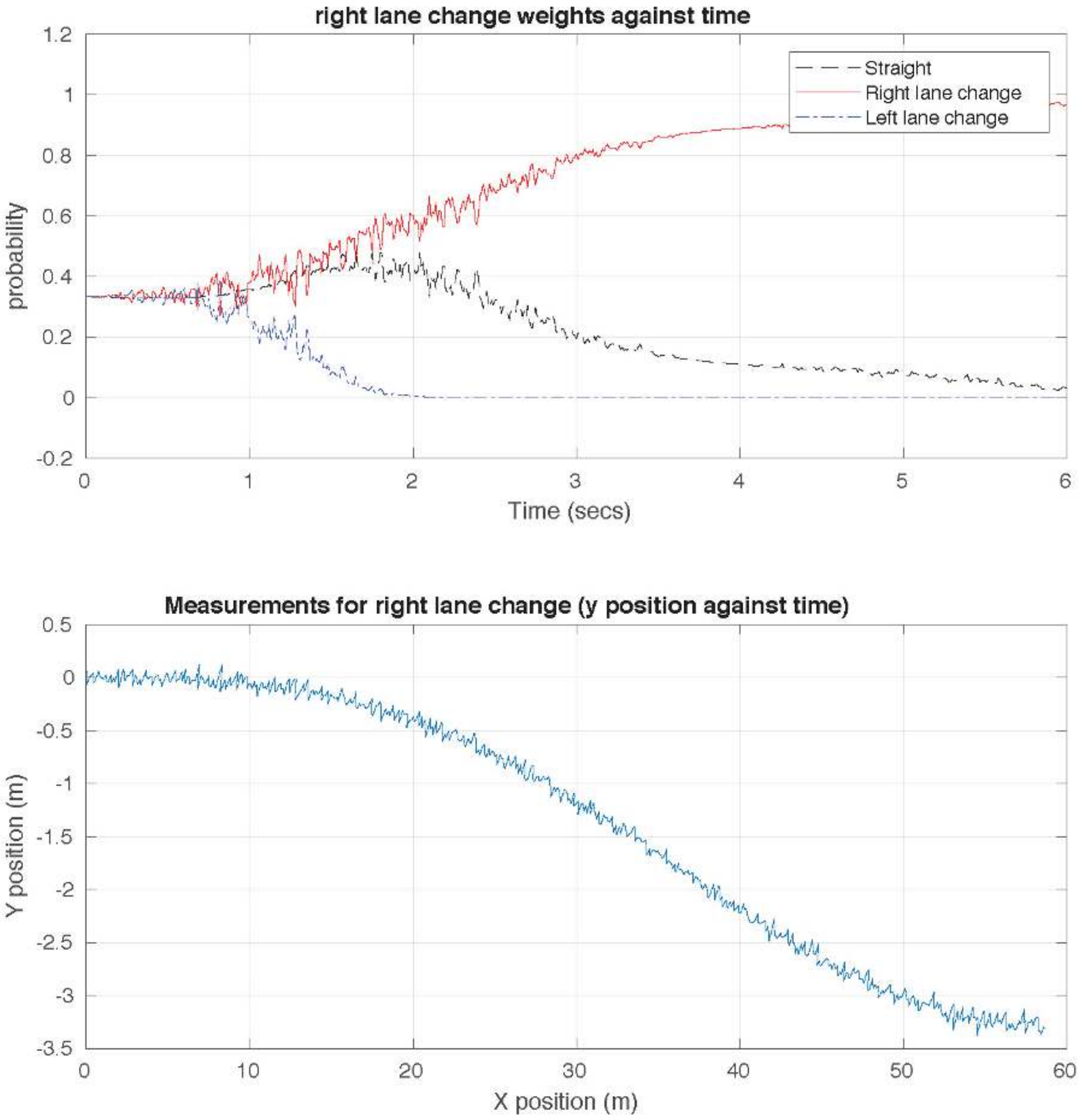


Figure 5.8: Right lane change detection when Q=0.1 and R=0.0025 with vehicle model

models is high while measurement noise is very low. Here, both filters would rely heavily on measurements. For the estimate itself, it is without consequence what the associated weights are as long as the resulting state estimate is optimal. There is a fundamentally different perspective, however, if the application goal consists of the inference of a correct model from a selection. Therefore, very careful tuning of the involved error statistics is

Table 5.2: Results of varying the process and measurement noise when evaluating with vehicle model

			Maneuver detection time (s)		
Case No	Q(diagonal)	R(diagonal)	Straight	Left lane change	Right lane change
1	0.001	0.0025	0.6	3.1	3.1
2	0.005	0.0025	0.9	2.8	2.8
3	0.01	0.0025	1.1	2.9	2.9
4	0.025	0.0025	1.9	1.3	1.3
5	0.05	0.0025	1.8	3.5	3.5
6	0.1	0.0025	2.4	4.3	4.3
7	[0.025 1 0.025 1]	0.0025	1.9	1.7	1.7
8	[0.01 1 0.01 1]	0.0025	1.39	2.4	2.4

necessary. A similar problem occurs at the beginning of the classification if the filters are initialized with a too large (individual) covariances: sudden model association switching can be observed during the early stages of the estimation when the transients in filter convergence are still significant. We have analyzed these issues extensively, and the findings will be subject of a future publication.

5.4 Conclusion

The challenge in complex traffic environments exists primarily due to the uncertainty in the maneuvers drivers may execute. A novel approach for identifying and predicting these maneuvers using an AMM is exemplary demonstrated for integration into the proposed architecture. Furthermore, detailed results of filter tuning and its evaluation with a single track vehicle model are discussed. As there is a significant variability in the way drivers execute the maneuvers, a popular approach is to lump these effects into process noise. Yet, literature provides very little analysis of the effect of process and measurement noise selection when applying MMAE for classification instead of tracking. The presented results therefore include a detailed analysis of noise parameter effects on the performance of the maneuver identification model. From the results, it can be concluded that the measurement noise has relatively less impact on the classification performance, the main impact is due to process noise. Hence, selecting the right motion model is undoubtedly the key in getting the most optimal performance. Further, it was observed that the filter is sensitive to proper initial state estimate covariance. An initial estimate with high covariance estimate affects the convergence performance of the filter. Hence, it is best to base the initial estimates from a filter rather than raw measurements.

Chapter 6

Behavior Identification in Prolonged Observations using AMM

In the previous chapter, an Autonomous Multiple Model (AMM) filter was evaluated for the effect of process and measurement noise employing only a single maneuver, with the assumption that it was synchronized right when the observed traffic participant initiated the maneuver. In practice however, observations will span across multiple maneuvers with the uncertainty associated with the initial maneuver conditions of the observed vehicle. The challenge addressed in this chapter is to enable inference using an AMM for a prolonged observation interval so that transitions in the behavior of the traffic participants can be captured.

6.1 Motion model for lane change

In the previous chapter, we evaluated an AMM based behavior identification model running three different elemental filter motion models, each corresponding to a maneuver. The motion model corresponding to straight behavior constituted of Constant Velocity, whereas the ones corresponding to left and right lane changes were based upon sinusoidal wave geometry with the maneuver length (L) assumed as a parameter. Hence, the states assumed were $\begin{bmatrix} x & v_x & y & v_y \end{bmatrix}'$. We defined the length of the maneuver as the longitudinal distance when starting from the origin lane and ending at the center of the target lane. This assumption aided in evaluating the filter parameters and understanding the intuition of filter tuning. For prolonged observation however, more information is required by the filters to identify the behavior correctly in the form of the maneuver initiation point (x_{mip}). x_{mip} is defined as the longitudinal position at which the traffic participant has accepted the gap and initiated the maneuver, as shown in Fig. 6.1. This was implicitly assumed as $x_{mip} = 0$ in the previous chapter. For prolonged observations however, information about both x_{mip} and L have to be obtained at run-time. In this section, two approaches are explored for achieving this requirement. In the first approach, x_{mip} is assumed as a parameter and L is modeled as a state. In the second approach, x_{mip} is modeled as a state in the motion model while L is assumed as a parameter. The following subsections describe the results of the approaches in detail.

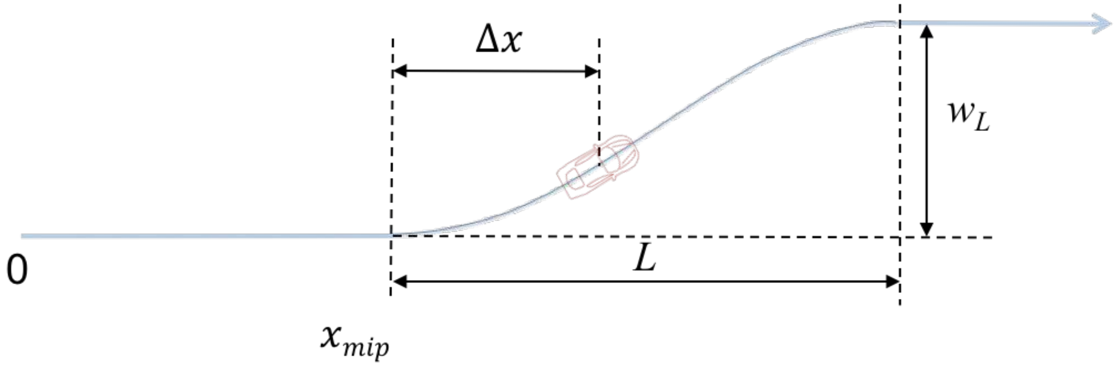


Figure 6.1: Exemplary scenario for behavior identification

6.1.1 Maneuver initiation point assumed as a parameter and maneuver length as a state

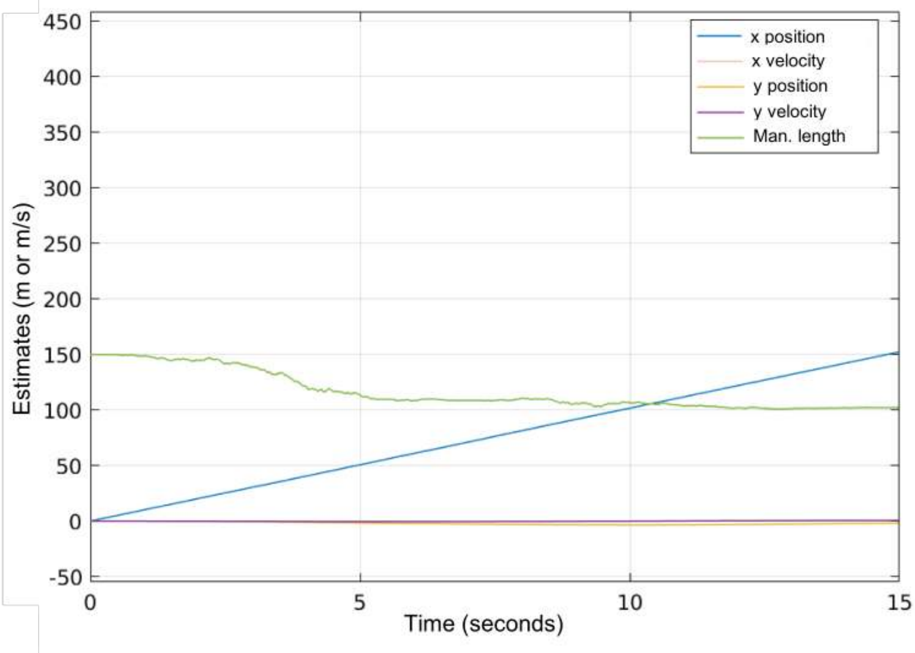


Figure 6.2: Evolution of estimated maneuver length (green) with $P_0 = 100$

The premise for this approach is the idea that if the maneuver initiation point can be obtained through a separate means (such as a curve fit, or lateral lane measurements), the maneuver length can be estimated online. Here, three motion models were considered, Constant Velocity for straight motion and sinusoidal waveforms for each of the left and right lane changes. The states for the lane change motion models were subsequently augmented as:

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y & L \end{bmatrix}' \tag{6.1}$$

with motion equations as described in section 3.1.5.2.

In this case, initial values of the states x, v_x, y, and v_y for the filter can be obtained from sensor measurements or from a tracking filter at a lower hierarchical level in the archi-

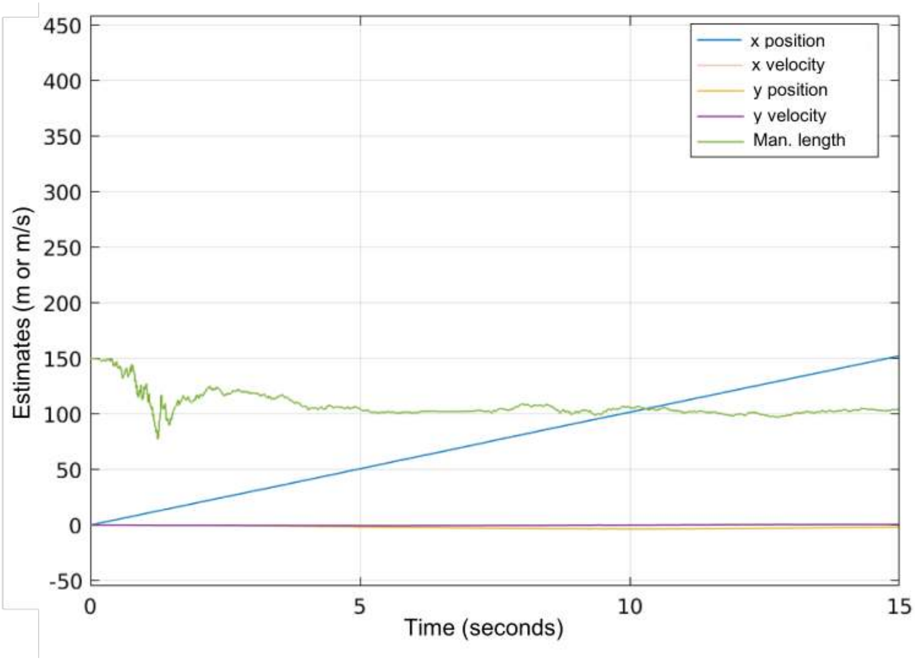


Figure 6.3: Evolution of estimated maneuver length (green) with $P_0 = 2500$

tecture, with high level of initial confidence (i.e., very low initial estimate covariances P). However, L cannot be measured directly and will have an uncertainty associated with it at the beginning of the maneuver as to the kind of maneuver that will be executed. Hence it has to be initiated with a nominal value of L, and with high initial covariance (P_0). If initial covariance of L is set to a very low value, then it takes longer for the filter to converge to an approximate maneuver length. In Fig. 6.2, the measurements were generated with a maneuver length of 100 m, and the EKF initial state estimate was set to 150 m, with a covariance of 100 m^2, i.e. a standard deviation of 10 m. The convergence time for the maneuver length estimate to settle within 5% band (less than 105 m) was 10.7 secs. With 2500 m^2 (50 m standard deviation) initial covariance for the initial value of L = 150 m and process noise of $Q = 10$, it took less than 5 secs (Fig. 6.3).

For the initial length of the maneuver set as 100 m, process noise $Q = 0.0025$ and measurement noise $R = 0.0025$, $P_0 = \text{diag}\left(\begin{bmatrix}100 & 100 & 100 & 100 & 1e5\end{bmatrix}\right)$ and an executed

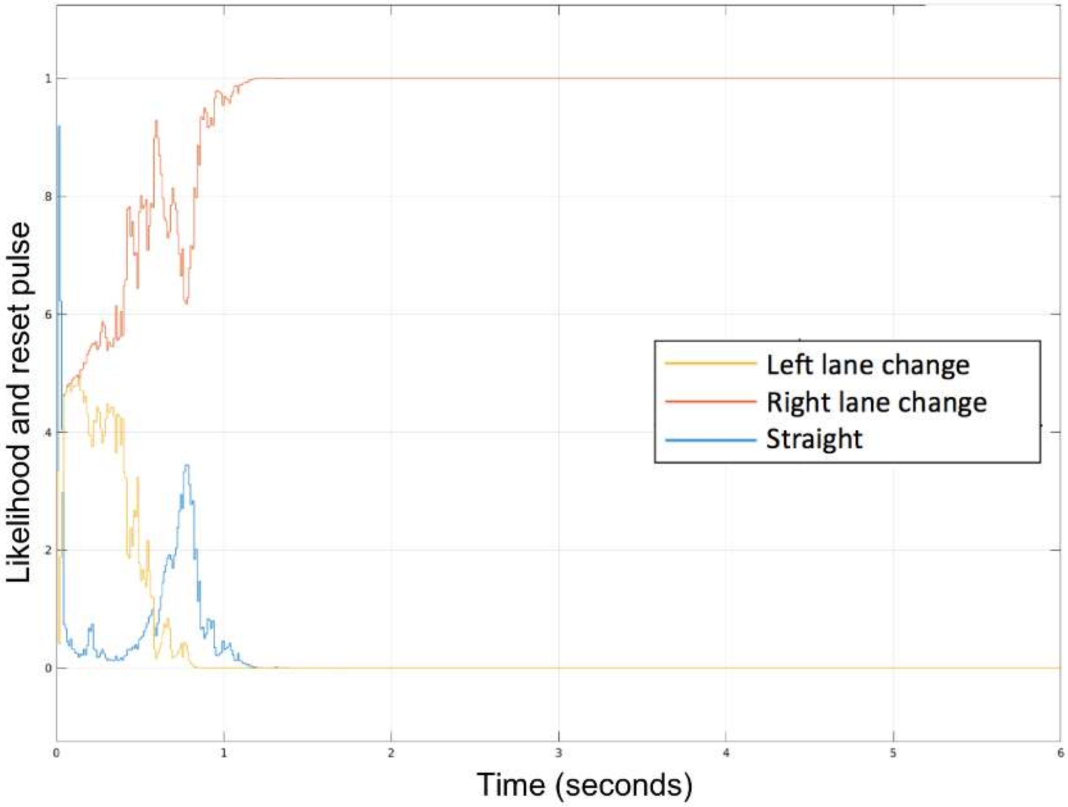


Figure 6.4: Probabilistic weights for maneuvers with AMM estimating maneuver length L

maneuver length of 60 m for a right lane change, the probabilistic weights of the AMM were as shown in Fig. 6.4. The evolution of the estimate of the maneuver length for the elemental filter running the right lane change motion model was as shown in Fig. 6.5, converging rather quickly. Not surprisingly, the evolution of the estimated L for the elemental filter running the left lane change model deviates largely from the ground truth as shown in Fig. 6.6. Hence, one needs to be careful when combining the estimates as per the filter weights and should rather not be included in the combined estimates. Since the weights of the left filter fall to zero within a second, the influence of the maneuver length is only there at the initial stages of the maneuver.

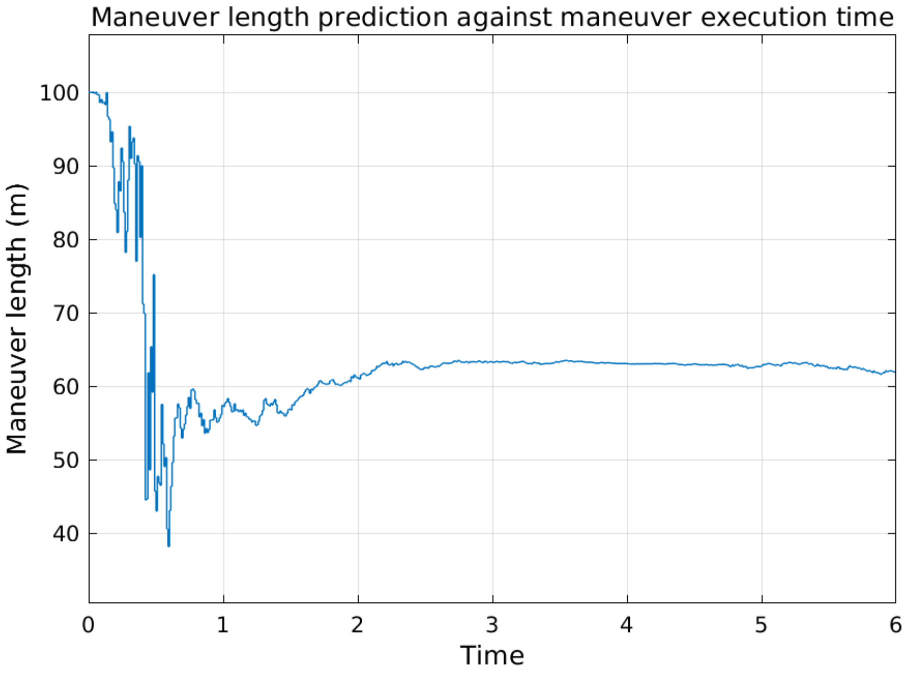


Figure 6.5: Evolution of estimated maneuver length for right lane change elemental filter

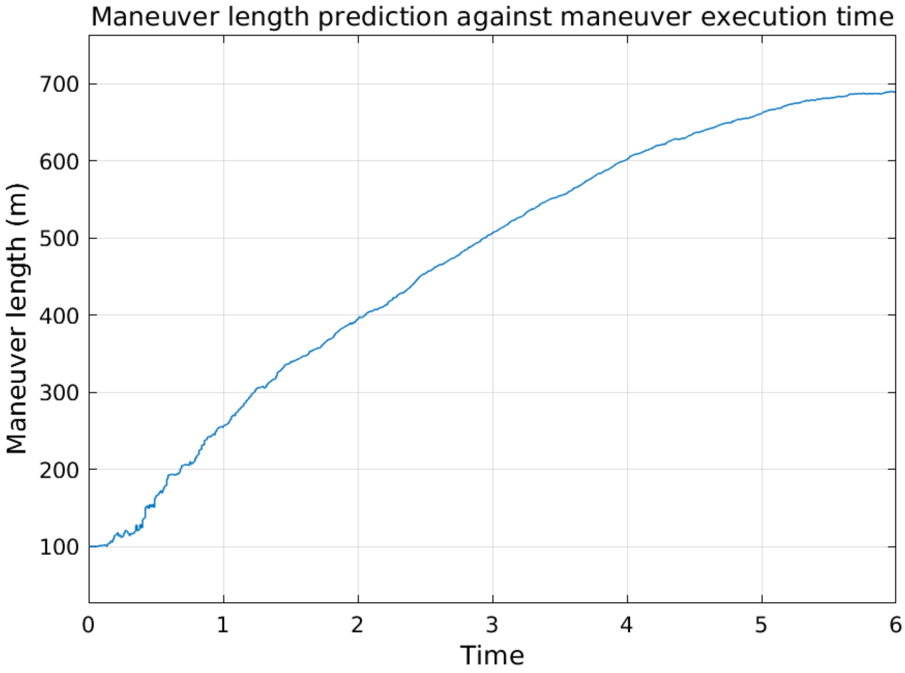


Figure 6.6: Evolution of estimated maneuver length for left lane change elemental filter

6.1.2 Maneuver initiation point as a state and maneuver length as a parameter

The premise for this approach is the idea that if the right set of candidate lane change behaviors, each with a different maneuver length, is assumed then it is possible to approximate the maneuver initiation point. Again, three elemental filters were considered for this case. First filter employed a constant longitudinal velocity and zero lateral velocity motion model for the straight maneuver, the other two filters employed sinusoidal waveforms motion model with the states augmented as:

$$\mathbf{x} = \begin{bmatrix} x & v_x & y & v_y & x_{mip} \end{bmatrix}' \tag{6.2}$$

where x_{mip} is the longitudinal position of the vehicle when the maneuver was initiated. The motion equations are as described in section 3.1.5.3. From the simulated experiments, it was learnt that the initial covariance of the x_{mip} should be kept high (i.e. low confidence in the initial estimate) when the filter is initialized since there is an uncertainty even in the initial estimate of the maneuver initiation point. Fig. 6.7 shows the trajectory of x_{mip} for the initial covariance of 200 in the exemplary use case. It was also found that as long as the filter is initialized even as the traffic participant is performing the maneuver, it converges quickly. If the filter stays in the straight maneuver for too long however, there is a delay in convergence due to the inertia of the Kalman Filter. Hence, if the straight maneuver has continued beyond a certain time, the filter needs to be reset.

To evaluate this model, the vehicle has to be observed for a while before it initiates the maneuver. Hence, measurements were generated from a synthetic ground truth for a simulated scenario as shown in Fig. 6.8. In the scenario, the vehicle travels straight with a constant velocity of 10 m/s for 10 seconds, and then performs a left lane change maneuver

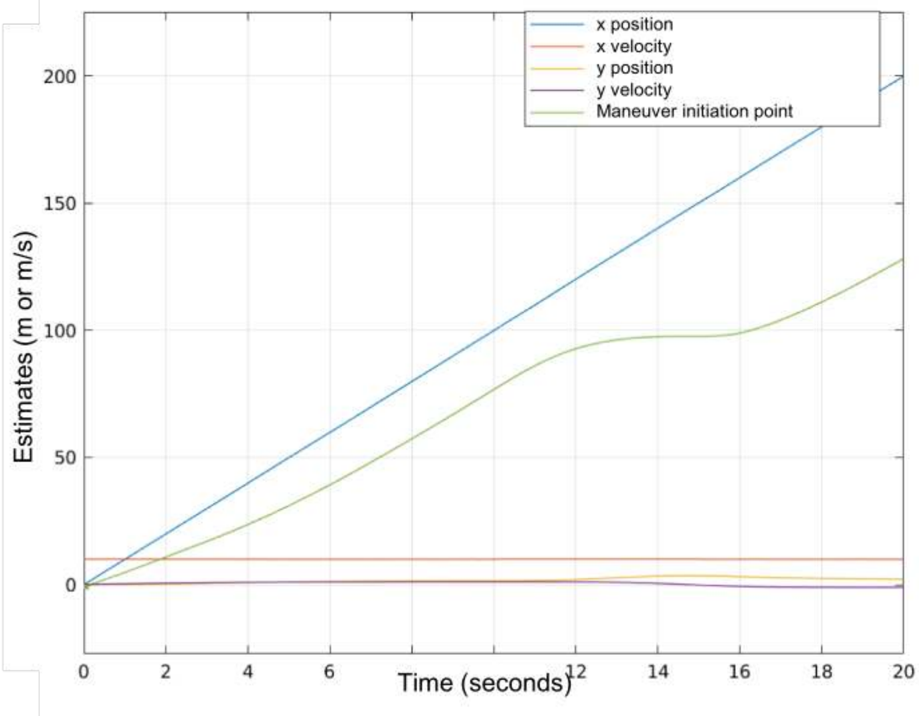


Figure 6.7: Evolution of estimated maneuver initiation point (green)

of 50 m, and then drives away from the scene at a constant velocity of 10 m/s.

From the model probability update equations of the AMM, Eqn. (5.1), it can be seen that the pre-normalized weights are calculated by multiplying the likelihood in the current time step of filtering with the normalized weights from the last time step. As can be seen from the probabilistic weights in Fig. 6.9, the likelihood drops instantaneously for the straight maneuver upon initiation of the lane change . Thereafter, it fluctuates a little and then bottoms out because the sine wave is almost linear in the middle. Finally, it rises towards the end of the left lane change maneuver.

The likelihood for the left lane change keeps spiking periodically (every 6 seconds) while the motion is straight. At these instants the corresponding motion model is almost linear in the longitudinal direction. Same can be concluded for the right lane change likelihoods. Based on the simulations, it was observed that employing asymmetric motion mod-

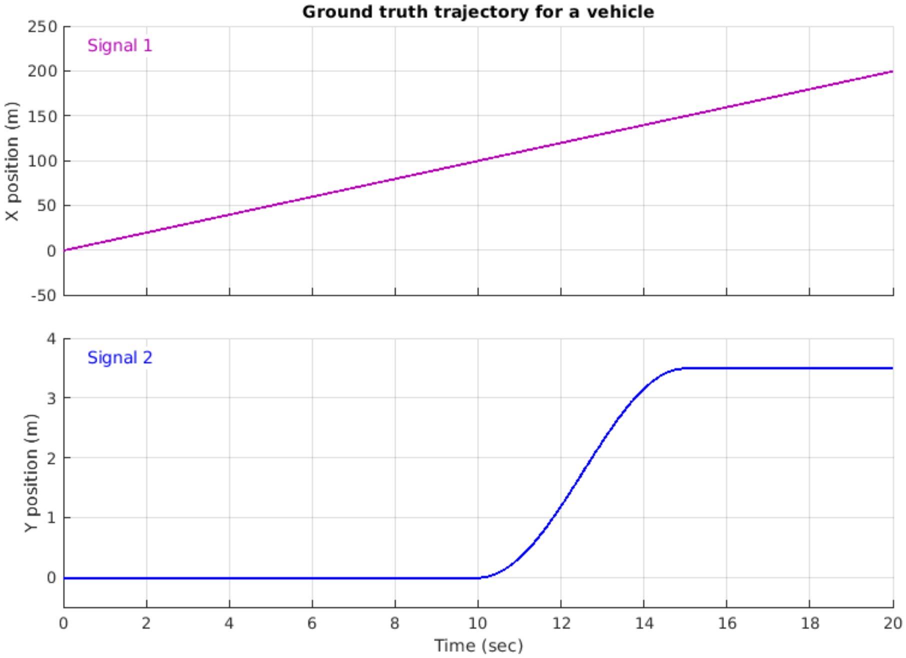


Figure 6.8: Simulated ground truth for the left lane change scenario

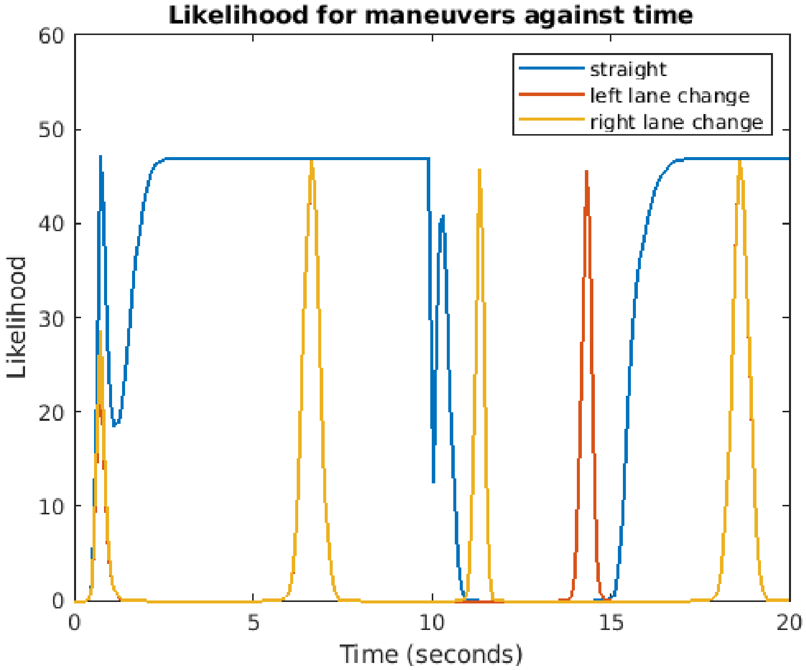


Figure 6.9: Maneuver likelihoods under no measurement noise for the lane change scenario

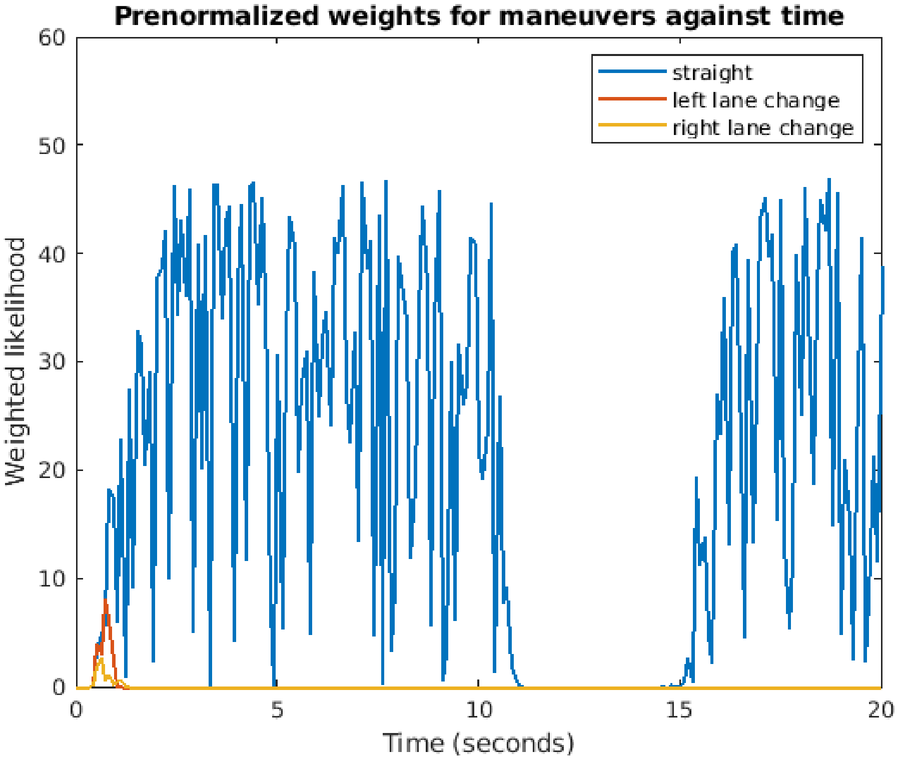


Figure 6.10: Pre-normalized weights of the elemental filters for the scenario

els (which have dissimilar lateral and longitudinal models) for the straight maneuver gave better results. CA or CV could be used for longitudinal motion, but for lateral, stationary (i.e zero velocity) motion model was superior.

As can be seen from Fig. 6.10, prenormalized weights (which are a product of weights from the last cycle and the likelihood) for the lane change behaviors do not rise once the lane change is initiated (i.e. between 10 and 15 seconds of the simulation time). This is because the weights from the last cycle were 0 and hence zero out, not reflecting the right history representation. This uncovers a limitation of AMM in switching from one mode to another. To handle this situation, AMM needs to be reset using an external logic. A possible option is to reinitialize it as soon as the likelihood for the straight maneuver drops. Owing to noisy measurements, the likelihood may also have fluctuations and will have to be smoothed (filtered) to detect a deviation from an expected trend. This will be discussed in the next section.

6.2 Smoothing the likelihood of the default behavior

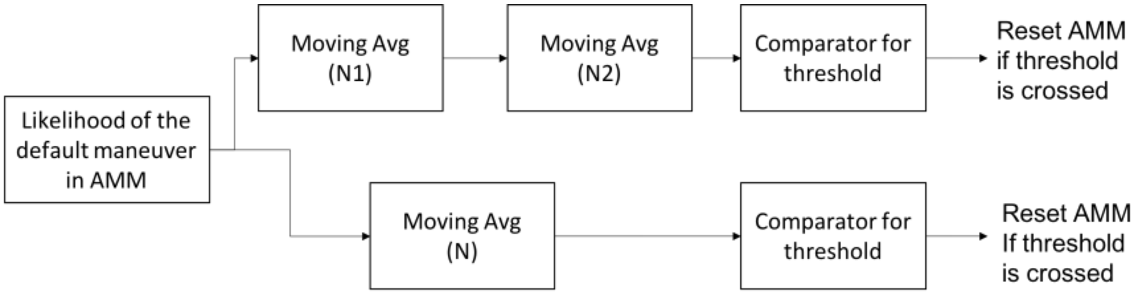

Likelihood of the default maneuver in AMM
Moving Avg (N1)
Moving Avg (N2)
Comparator for threshold
Reset AMM if threshold is crossed
Moving Avg (N)
Comparator for threshold
Reset AMM If threshold is crossed

Figure 6.11: Block diagram for a cascaded moving average filter

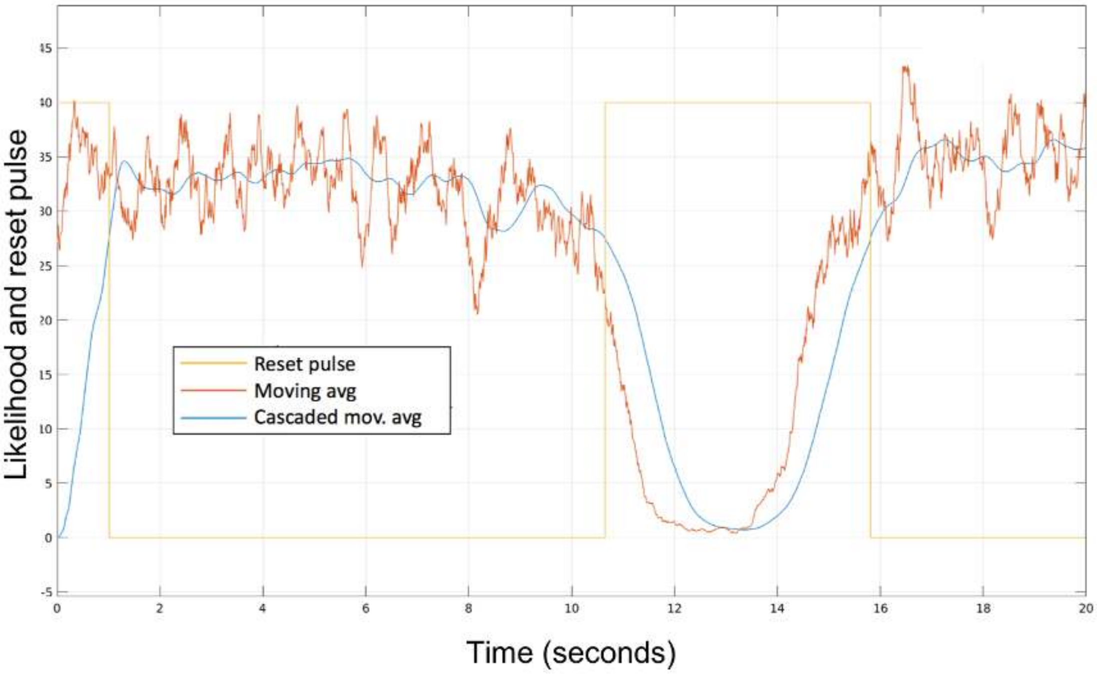


Figure 6.12: Filtering the likelihood of the straight behavior

For smoothing the likelihood, it was determined that a cascaded moving average filter was more effective at smoothing than a single moving average filter. Fig. 6.11, shows the block diagram depicting how single moving average filter (N = 5) and a cascaded moving average filtering (N1 = 5, and N2 = 10) can be utilized for resetting the AMM after smoothing. It should be noted that smoother output could also be obtained from a single filter by increasing the average window(for eg. N = 50), however, this introduces heavy lag in the filter output and a delayed detection of the behavior compared to the cascaded moving average filter. Applying a comparator on the unfiltered likelihood signal leads to undesired switching. A smoothed likelihood from the cascaded filter on the other hand, can help identify the deviation of likelihoods sooner, and can potentially have an earlier detection even though the signal is delayed (Fig. 6.12). The next section will show the results of implementing this reset logic circuit with the AMM.

6.3 AMM with reinitialization logic

To ensure the filter weights are switched in time with the performed behaviors, the filters have to be reset at the right time and not too late from the point when the likelihood of the default behavior (straight) reduces. Delays in this reset add to the cumulative delays in the behavior identification, which means the threshold for the increasing likelihood becomes a tuning factor. If not tuned properly, the likelihood of the straight maneuver doesn't rise from zero. The cascaded moving average approach applied above can also be extended to the weighted recency averaging, which uses the divergence ratio to identify a change in the likelihood. This is a popular approach in the robotic Simultaneous Localization And Mapping (SLAM) to identify an update in the robot's environment or to identify a kidnapped robot situation [61, 65]. Even in this case however, a threshold needs to be identified for the divergence ratio.

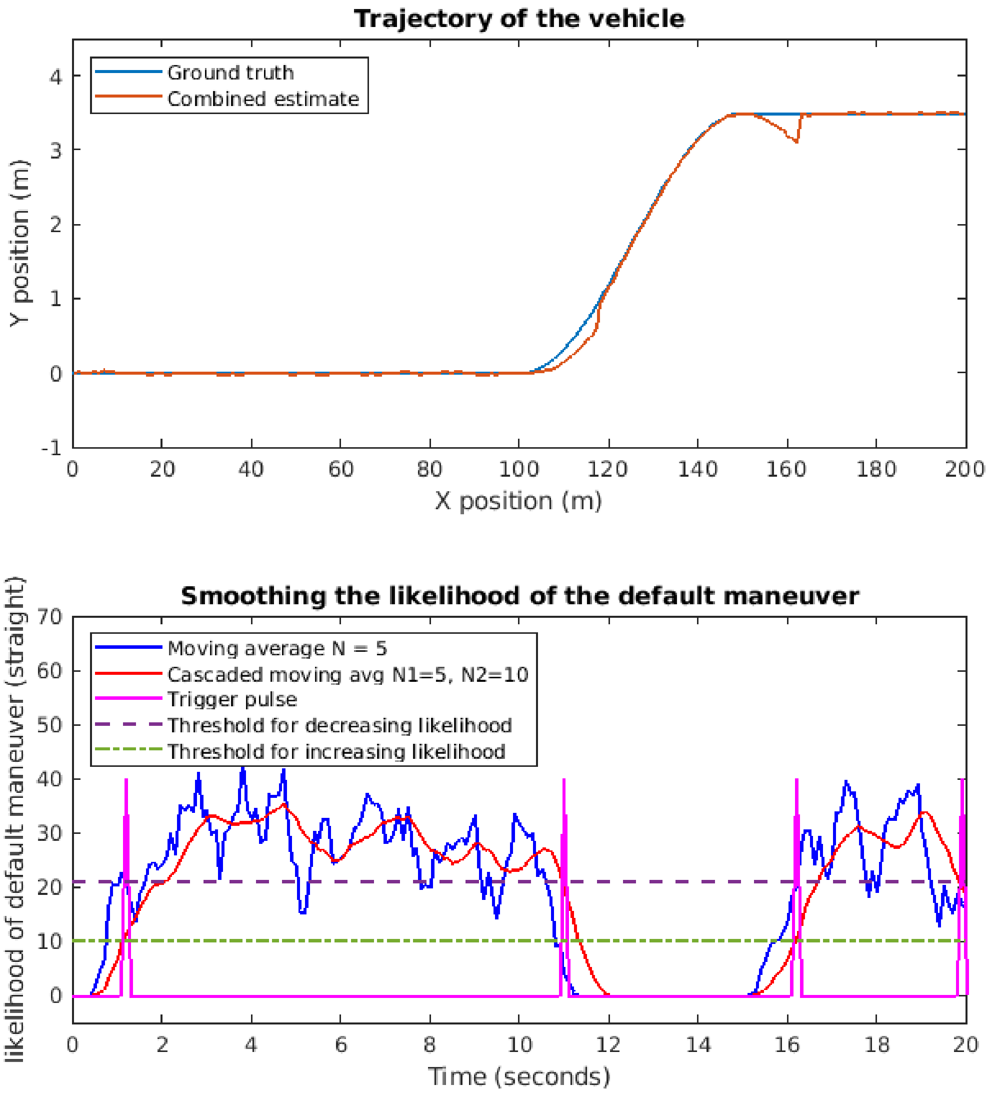


Figure 6.13: Tracking performance of the AMM with the reset logic

Figure 6.13 shows the results of the AMM filter reset logic. It should be noted that the AMM needs a reset at two points, once when the traffic participant exits from the default behavior, and another when it returns to the default behavior. The lower plot shows the times at which the reset pulse was enabled for reinitializing the filter weights as well as the initial states appropriately. It also illustrates how the cascaded moving average as well as

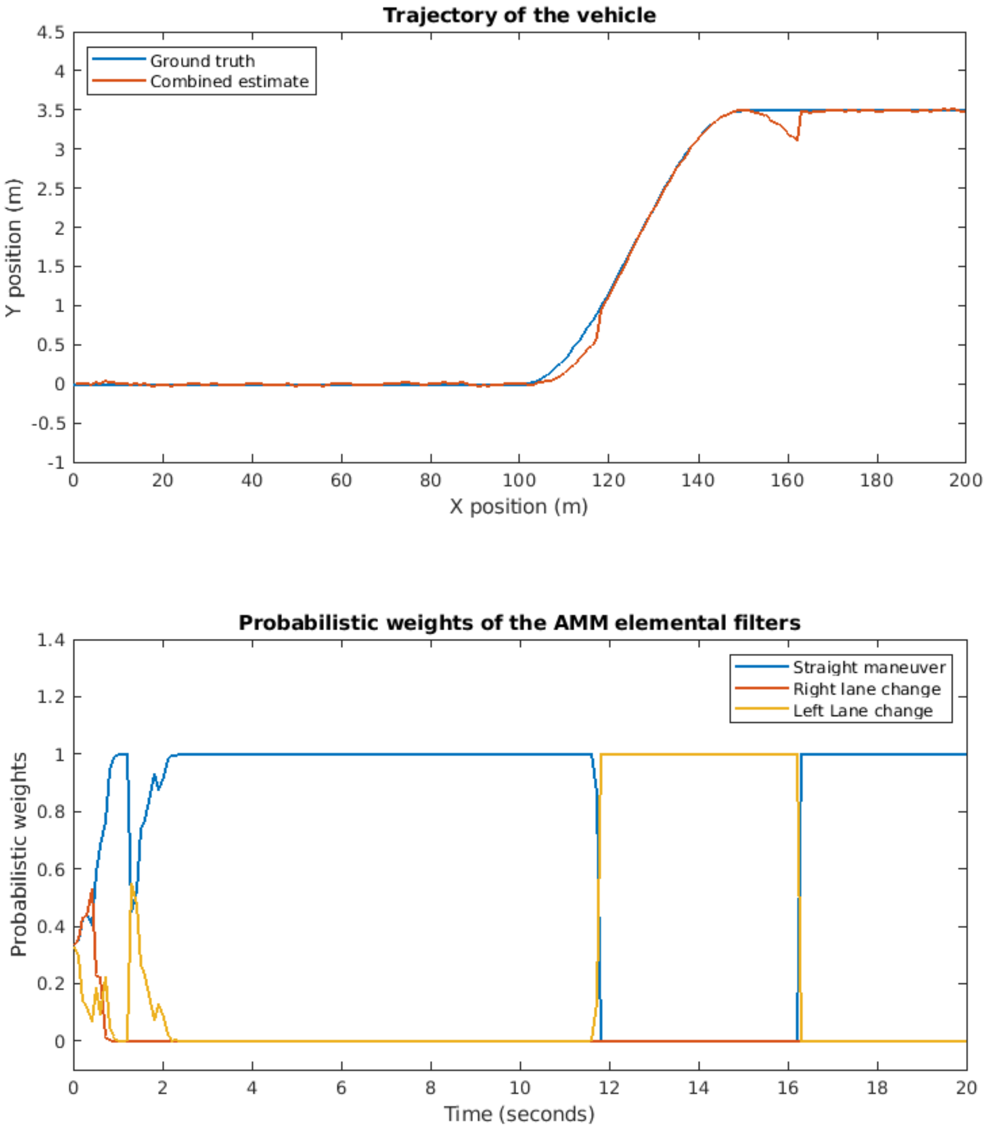


Figure 6.14: AMM behavior weights with the reset logic

the single moving average filter evolve over the time. The plot on the top shows the results of the combined estimates that the filter is following against the ground truth. The delay in the reset as well as in detection is observed for both when the behavior is initiated and after it is completed. After initiation of the maneuver, the reset pulse activates with a delay of 1

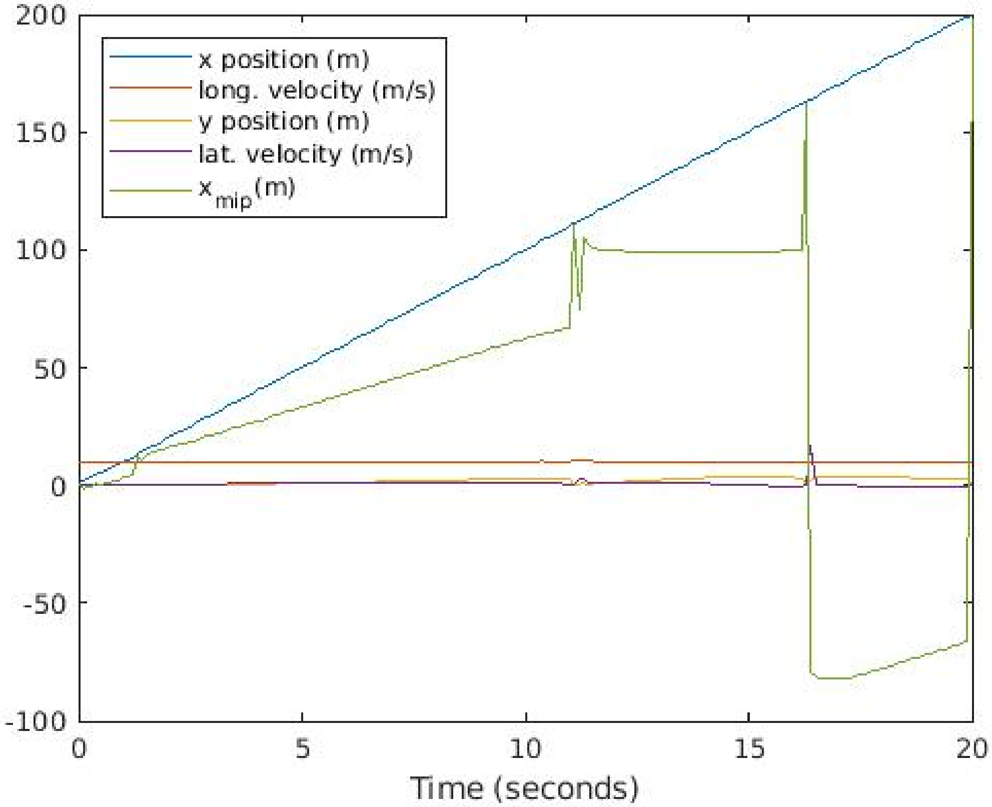


Figure 6.15: Evolution of estimates showing the correction in X_{mip} due to reset pulses

second. The lane change maneuver is detected 0.7 seconds later, resulting in a cumulative detection time of 1.7 seconds. After the maneuver is completed, another reset pulse is activated with a delay of 1.1 seconds, the straight maneuver is detected 0.2 seconds later, leading to a cumulative detection time of 1.3 seconds. As a result of these detection delays, the filter's trajectories diverge slightly from the ground truth at the maneuver initiation and completion instants. The root mean squared error (RMSE) of the filter's estimated position with the ground truth are 0.0778 m for x-position and 0.0735 for the y-position. Fig. 6.14 shows the behavior weights of the AMM plotted against the time. As can be seen, the model correctly identifies the left lane change behavior. Fig. 6.15 shows the trajectories of the combined estimates of the filter. It can be seen that the reset pulses help correct the maneuver initiation point (X_{mip}) estimates.

6.4 Conclusion

In this chapter, two approaches for modeling the lane change behaviors in prolonged observations were suggested and evaluated. The first approach modeled the maneuver initiation point as a parameter and the maneuver length as a state. The second approach considered the maneuver initiation point as a state, while keeping maneuver length as a parameter. By carefully selecting the appropriate set of lane change behaviors (each with a different maneuver length as a parameter), a close approximation for the maneuver initiation point can be determined at run time. The latter approach saves the additional computation and complexity involved when alternatively employing external means for identifying the maneuver initiation point such as curve fit. After exploring both approaches, it was also discovered that AMMs have an inherent limitation in their weight update equations, such that the system may not transition from one mode to another, and subsequently requires an external reset. An approach for resetting the AMM utilizing changes in the likelihood of the default behavior has been presented and applied to the use case. Two limitations were observed in this approach: first, the thresholds for the likelihoods have to be determined and may vary depending on the operating conditions (determined heuristically in this study); second, the reset logic has its own inherent delay, which adds to the lag in the detection of the behavior. To address these limitations, techniques like Interacting Multiple Model filter (as they have re-initialization inherently incorporated) should be explored.

Chapter 7

Behavior Identification using IMM

In this chapter, Interacting Multiple Model (IMM) filters will be analyzed and employed for behavior identification of traffic participants. This chapter addresses the challenge of enabling inference using an IMM for prolonged observations. In doing so, it also compares their performance with the AMM filters. First, the Interacting Multiple Model (IMM) algorithm is introduced. Then, the IMM is analyzed against different process noise and compared with AMM for behavior identification. The subsequent section evaluates the performance of IMM with the use case from the previous chapter. Finally, the developed technique is validated with a vehicle trajectory extracted from a real world traffic dataset from German highways.

7.1 Interacting Multiple Model alogrithm

Autonomous Multi Model (AMM), also known as classic MMAE, employs a bank of Kalman filters (or variations such as EKF, UKF) running in parallel, independently of each other while each represents a different mode of operation. They have proven success [25] in parameter identification or model identification of a system that does not change over time. However, if the system switches between different mutually exclusive modes over time, the probabilistic weights of the AMM may not follow the system as it does not include a stage to re-initialize the system. Hence, a technique external to the AMM algorithm has to be employed to re-initialize its model weights. Typically, motion models representing different behaviors fall under this category of switching systems. Since a single traffic participant may follow a straight behavior for a while, then perform a behavior for the left lane change maneuver, for instance, before following a straight behavior, the motion model of the traffic participant keeps changing over time. This is where Interactive Multi Model (IMM) algorithm applies. In addition to the three steps similar to AMM (i.e. model specific filtering, model probability update and combination), an IMM has a fourth step called mixing/interaction, which carries out two things based on certain conditions: i) reinitializing the initial weights of each model specific filter for the next time step by mixing the updated probability weights from the current time step; ii) reinitializing the initial state estimate as well as the associated covariance for the next time step by mixing them with the weights from the current time step. This enables the IMM to follow the system faster when mode switching occurs. The mixing of the weights, estimates, and associated covariances is based on the predicted transition probabilities between discrete switching states of the hybrid system.

7.1.1 Transition probability matrix for an IMM

Just to recall, a hybrid system is characterized by two elements, a discrete part (modes) and a continuous part (base states). In this work we will restrict our scope to Markov Jump Linear Systems which are conveniently used to represent the traffic participants as hybrid systems. The discrete and the continuous elements of the system can be represented with following equations:

$$\mathbf{p}(M_{k+1}) = \Lambda'_{trans} \cdot \mathbf{p}(M_k) \tag{7.1}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), M_k) \tag{7.2}$$

Equation (7.2) represents the state propagation equation for the continuous part of the system given that the system is in mode M_k at time step k, with $t \in [(k-1)T_s, kT_s]$, where T_s is the sampling time. Eqn. (7.1) describes the discrete part of the system, where, M_k represents the mode of the system at a discrete time step k, with $M_k \in [m_1, m_2, ...m_n]$; $\mathbf{p}(M_k)$ is the vector of probabilities of the modes of the system at time interval k, i.e. $\mathbf{p}(M_k) = [\mathbb{P}(m_1), \mathbb{P}(m_2), ..., \mathbb{P}(m_n)]'$; Λ_{trans} is the transition probability matrix that governs the transition of the system between any mode time step k and any mode at time step $k+1$. For a system that can operate in n different modes, this yields a square matrix of dimensions n by n,

$$\Lambda_{trans} = \begin{bmatrix} \lambda_{11} & \dots & \lambda_{1n} \\ \vdots & \ddots & \\ \lambda_{n1} & & \lambda_{nn} \end{bmatrix} \tag{7.3}$$

where elements of each row sum up to 1, i.e.

$$\sum_{b=1}^{n} \lambda_{ab} = 1, \ \forall \ a \in [1, n] \tag{7.4}$$

The following section will describe how the IMM algorithm utilizes the transition probability matrix.

7.1.2 Stages of IMM

Figure 7.1 describes the IMM process. In the figure, $\mathbf{e}_k^{-q}$ is the vector of measurement residuals, $\mathbf{z}_k$ is the measurement vector, $\hat{\mathbf{x}}_k^{+q}$ is the estimated state vector for the q^{th} elemental filter, where $q \in [1, M.I]$ and $\hat{\mathbf{x}}_k^{+0q}$ is the mixed initial estimate for the q^{th} elemental filter. The IMM consists of four stages [24]:

1. *Interaction and Mixing:* In the very first stage of the cycle, the weights and the estimates from the last cycle are mixed as per the Markov transition probabilities. At time $t = 0$, these will result from the initial conditions. The predicted probability for the filter to end up in mode j in this cycle, given that it was in the mode i during the last cycle is given by:

$$w_k^{(i|j)} = \frac{1}{\mu_k^j} w_{k-1}^i \lambda_{ij} \tag{7.5}$$

$$\mu_k^j = \sum_{i=1}^{M} w_{k-1}^i \lambda_{ij} \tag{7.6}$$

where $\lambda_{ij} \in \Lambda_{trans}$ is the transition probability from mode i to j, μ_k^j is a normalizer and basically represents the predicted probability of the the system ending up in mode j (i.e. filter j being the one with highest likelihood) in the current cycle. This is nothing but the result of the probabilistic weights of all the filters in the last cycle multiplied by the Markov mode transition probabilities for transition to mode i. Since this stage is at the beginning of the current cycle, before the measurements have arrived, these are called as predicted probabilities. These probabilities are then utilized in computing the mixed initial conditions for each elemental filter yielding:

$$\hat{\mathbf{x}}_k^{+0j} = \sum_{i=1}^{M} w_k^{(i|j)} \hat{\mathbf{x}}_k^{+i} \tag{7.7}$$

$$P_k^{+0j} = \sum_{i=1}^{M} w_k^{(i|j)} \left[\left(\hat{\mathbf{x}}_k^{+i} - \hat{\mathbf{x}}_k^{+0j} \right) \left(\hat{\mathbf{x}}_k^{+i} - \hat{\mathbf{x}}_k^{+0j} \right)^T + P_k^{+i} \right] \tag{7.8}$$

2. *Model specific filtering:* The expressions for prediction and update for each single-model filter resemble the ones employed in the Extended Kalman filter (EKF) [24]. The mixed initial conditions computed from the first stage above are utilized for every time step in the prediction stage, i.e.

$$\hat{\mathbf{x}}_{k+1}^{q} = F_k^q \hat{\mathbf{x}}^{+0q} + B_k^q \mathbf{u}_k^j \tag{7.9}$$

$$P_{k+1}^{-q} = H_k^q P_k^{+0q} H_k^{qT} + Q_k^q \tag{7.10}$$

3. *Model probability update:* This stage computes the likelihood of the measurements for every single-model filter estimate, and subsequently determines the associated weights for each filter. For M filters, the initial weight for every filter is uniformly distributed, i.e. $w_0^q = 1/M$ for q = 1, 2, ... M. The update relations utilize the predicted probabilities calculated in step 1, yielding:

$$w_k^q = \mu_{k-1}^q \mathbf{p}(\mathbf{z}_k | \mathbf{x}_k^{-q}) \tag{7.11}$$

$$w_k^q \leftarrow \frac{w_k^q}{\sum_{q=1}^{M} w_k^q} \tag{7.12}$$

$$p(\mathbf{z}_k | \mathbf{x}_k^{-q}) = \frac{1}{\left| 2\pi E_k^{-q} \right|^{1/2}} e^{\left\{ -\frac{1}{2} \mathbf{e}_k^{-qT} (E_k^{-q})^{-1} \mathbf{e}_k^{-q} \right\}} \tag{7.13}$$

$$E_k^{-q} = E \left\{ e_k^{-q} [e_k^{-q}]^T \right\} = H_k^q P_k^{-q} [H_k^q]^T + R_k^q \tag{7.14}$$

4. *Combination:* The estimates and covariances from all single-model filters are fused utilizing their associated weights to provide a combined estimate, i.e.

$$\hat{\mathbf{x}}_k^+ = \sum_{q=1}^{M} w_k^q \hat{\mathbf{x}}_k^{+q} \tag{7.15}$$

$$P_k^+ = \sum_{q=1}^{M} w_k^q \left[\left(\hat{\mathbf{x}}_k^{+q} - \hat{\mathbf{x}}_k^+ \right) \left(\hat{\mathbf{x}}_k^{+q} - \hat{\mathbf{x}}_k^+ \right)^T + P_k^{+q} \right] \tag{7.16}$$

Note that if the Markov transition probability matrix Λ_{trans} is an identity matrix, the IMM algorithm reduces to an AMM.

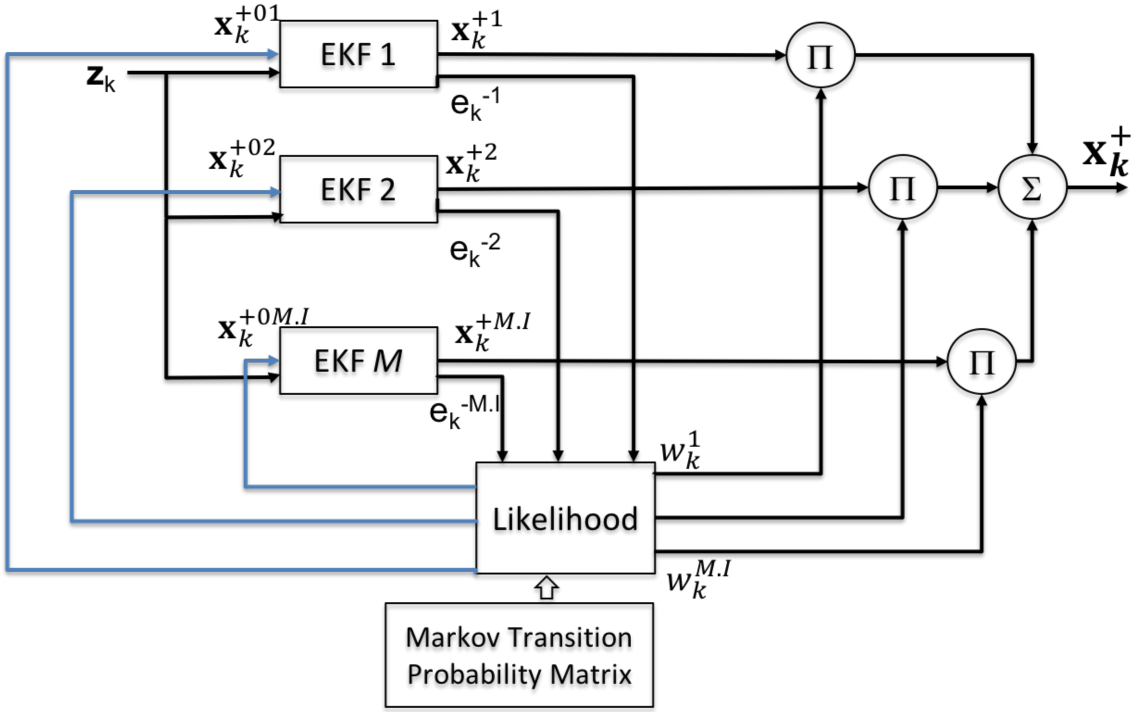


Figure 7.1: Interacting Multiple Model filter approach

7.2 Maneuver identification using IMM

7.2.1 Motion models and measurement model

Similar to chapter 5, the IMM is employed to distinguish between three maneuvers in this section : straight motion, left lane change and right lane change. The straight motion assumes a Constant Velocity motion model. The lane changes in this chapter assume a sinusoidal motion model with the maneuver initiation point (x_{mip}) as a state and maneuver length as a parameter according to the description in section 3.1.5.3. The measurement model is assumed to be linear with x and y positions directly observable.

7.2.2 Evaluation of the maneuver identification model

In this section, the IMM based maneuver identification model is evaluated using the single track vehicle dynamics model described in chapter 5. To recall, the ground truth for the straight maneuver is created by maintaining a zero steering wheel angle for the vehicle model whereas the ground truth for left and right lane change maneuvers result from a sinusoidal input to the steering wheel of 10 second period length, therefore leading to a 60 meter maneuver. Measurements were generated by adding white Gaussian noise with parameters given in table 7.1. The elemental filters in the IMM for left and right lane changes assume a fixed maneuver length of 60 meters.

For underlying real physical processes, the Markov transition probabilities for the IMM are a function of sampling time. In this study, we assumed a sampling time of 0.1 seconds. The transition probabilities were then determined heuristically with the assumption that the lane change maneuvers would typically last anywhere from 4 to 10 seconds. In a 10 second maneuver, there would be a chance in the order of 1/100 for the traffic participant to switch from a lane change maneuver to a straight maneuver. As a result, the observed participant will spend the majority of the time maintaining the current state at this given sampling rate. Similar transition probabilities were assumed for switching from straight to any of the maneuvers. It should be noted however, that the straight maneuvers are one of the default maneuvers in driving and traffic participants typically spend majority of time in this condition. So switching from straight to one of the lane change maneuvers would have an even lower occurrence rate with a sampling time of 0.1 seconds. With these assumptions, the transition probability matrix was chosen as

$$\Lambda_{trans1} = \begin{bmatrix} 0.97 & 0.015 & 0.015 \\ 0.015 & 0.97 & 0.015 \\ 0.015 & 0.015 & 0.97 \end{bmatrix} \tag{7.17}$$

Remark. *Whereas, the chosen transition probabilities work well for our evaluation, these should be determined from traffic data statistics in practice. The chances of a participant switching from a straight maneuver to a lane change maneuver, for example, depend on the driving conditions. For free flowing (highway) traffic, these chances will be much lower compared to peak hour bumper-to-bumper traffic. Hence, contextual information can be utilized in determining the appropriate set of pre-determined transition probabilities in real-time.*

It should be noted that the lane change motion models in our maneuver identification model employ an additional state, x_{mip}, the maneuver initiation point, in contrast to the straight driving motion model. Hence, mixing, re-initialization and combination of probabilities between the filter cycles is performed only for x, v_x, y and v_y states for all the elemental filters. The estimates for the maneuver initiation point are autonomous for the respective elemental filter.

The detection times for all maneuvers under different process and noise parameters

Table 7.1: Detection times for IMM with Λ_{trans1}

Case no	Q (diagonal)	Straight Detection time (s)	RMSE x	RMSE y	Right Detection time (s)	RMSE x	RMSE y	Left Detection time (s)	RMSE x	RMSE y
1	0.001	0.9	0.1399	0.0201	2.8	0.1383	0.0443	2.3	0.1455	0.0615
2	0.005	1.1	0.109	0.0255	2.8	0.1083	0.0347	2.3	0.1105	0.0397
3	0.01	1.2	0.0922	0.0288	2.8	0.0917	0.0331	2.3	0.0932	0.0369
4	0.025	1.5	0.0705	0.0336	2.8	0.0701	0.0345	2.3	0.071	0.0373
5	0.05	1.7	0.0575	0.0369	2.2	0.0572	0.0369	2.4	0.0579	0.0388
6	0.1	2.3	0.0496	0.0397	2.9	0.0493	0.0393	2.6	0.0498	0.0407
7	0.025 1 0.025 1	1.6	0.0717	0.0351	2.1	0.0715	0.0353	2.4	0.0718	0.038
8	0.01 1 0.01 1	1.4	0.0896	0.0323	2.1	0.0894	0.0335	2.3	0.0897	0.037

Table 7.2: Detection times for IMM with Λ_{trans2}

Case no	Q (diagonal)	Straight Detection time (s)	RMSE x	RMSE y	Right Detection time (s)	RMSE x	RMSE y	Left Detection time (s)	RMSE x	RMSE y
1	0.001	0.9	0.1399	0.0182	2.8	0.1363	0.0464	2.3	0.1455	0.0625
2	0.005	1.1	0.109	0.0245	2.8	0.1083	0.0351	2.3	0.1105	0.0396
3	0.01	1.2	0.0922	0.0282	2	0.0917	0.0329	2.3	0.0932	0.0372
4	0.025	1.4	0.0705	0.0332	2.1	0.0701	0.0343	2.3	0.071	0.0371
5	0.05	1.7	0.0575	0.0367	2.1	0.0572	0.0368	2.4	0.0579	0.0386
6	0.1	2	0.0496	0.0396	2.8	0.0493	0.0393	2.6	0.0498	0.0405
7	0.025 1 0.025 1	1.5	0.0718	0.0377	2.1	0.0715	0.0351	2.4	0.0718	0.0377
8	0.01 1 0.01 1	1.4	0.0896	0.0319	2.1	0.0894	0.0333	2.3	0.0897	0.0369

Table 7.3: Detection times for AMM

Case no	Q (diagonal)	Straight Detection time (s)	RMSE x	RMSE y	Right Detection time (s)	RMSE x	RMSE y	Left Detection time (s)	RMSE x	RMSE y
1	0.001	0.9	0.1399	0.0175	1.9	0.1383	0.0593	2.3	0.1459	0.0841
2	0.005	1	0.109	0.0241	1.9	0.1083	0.0353	2.3	0.1106	0.0448
3	0.01	1.2	0.0922	0.0279	1.9	0.0917	0.0328	2.3	0.0932	0.0383
4	0.025	1.4	0.0705	0.033	2.1	0.0701	0.0342	2.3	0.071	0.0369
5	0.05	1.6	0.0575	0.0366	2.1	0.0572	0.0367	2.4	0.0579	0.0384
6	0.1	2.2	0.0496	0.0396	2.8	0.0493	0.0393	2.5	0.0498	0.0404
7	0.025 1 0.025 1	1.5	0.0717	0.0347	2.1	0.0715	0.0351	2.3	0.0718	0.0375
8	0.01 1 0.01 1	1.4	0.0896	0.0317	2	0.0894	0.0332	2.3	0.0897	0.0368

are summarized in table 7.1. For cases 1 and 2, the model tends to switch between the respective lane change and a straight maneuver for the first few seconds before convergence. A likely explanation for lies in the initial delay between a sinusoidal steering wheel input and the (non-sinusoidal) path followed by the vehicle at first. As a result, the innovation equations of the lane change and the straight maneuver initially compete against each other, thus reflected in the switching. However, this can be addressed by including this discrepancy in the process noise. Although, detection times for the lane change maneuvers do not exhibit much of a variation with change in process noise, the switching between the straight as well as the respective lane change maneuvers is less for the cases 4, 5 and 6. This is also reflected in the RMSE between the ground truth and the estimated trajectory of the traffic participant. The RMSE is the least for the process noise 0.05 and 0.1 (cases 5 and 6) in the table.

Table 7.3 shows the detection times for the AMM based maneuver identification mode for comparisonl. Here, the AMM based identifier is marginally superior to the IMM in identifcation in a given maneuver. However, the IMM exhibits its strength in the prolonged observations as will be described in the next section. Further, if the transition probability matrix is chosen as below, the IMM behaves almost the same as the AMM. This is reasonable as an IMM with a diagonal transition probability matrix functions as a pure AMM. The detection times for the IMM with the following transition probabilities are shown in table 7.2.

$$\Lambda_{trans2} = \begin{bmatrix} 0.99 & 0.005 & 0.005 \\ 0.005 & 0.99 & 0.005 \\ 0.005 & 0.005 & 0.99 \end{bmatrix} \tag{7.18}$$

7.3 IMM for prolonged observations

After evaluation under different assumptions of process noise, the IMM maneuver identifier the straight, left lane change and right lane change motion models was assessed against the use case we identified in chapter 6, subsection 6.1.2. To recall, in the use case the observed participant travels with a constant velocity of 10 m/s in a straight motion for 10 seconds, performs a left lane change maneuver of length 50 m and then proceeds straight for another 5 seconds. The initial conditions for the filter were chosen as $x = 0$, $y = 0$, $v_x = 10$ and $v_y = 0$. The maneuver length parameter for the elemental filters were set to 50 meters.

Figure 7.2 shows the inference results for the use case with IMM. The lower plot illustrates the probabilistic weights for the maneuvers changing as the scenario evolves. The weights diverge after 0.6 seconds, identifying the straight maneuver correctly by 0.8 seconds. As the traffic participant initiates a left lane change maneuver at 10 seconds, it is identified in 1.1 seconds. After the traffic participant completes the maneuver, some ambiguity between the left lane change and straight maneuvers appears briefly before the weights switch at 0.4 seconds after the completion. The straight maneuver is again classified correctly 0.7 seconds after the participant completes the maneuver. The upper plot shows how closely the estimates are tracking the measurements and the ground truth. The RMSE error between the ground truth and the estimated position is 0.0539 m and 0.0298 m for x and y positions respectively, as shown in table 7.3. Compared to the AMM based maneuver identification evaluated in the last chapter the IMM proves superior, as the AMM exhibited the detection times of 1.7 and 1.3 seconds, as well as the RMSE errors for x and y positions of 0.0778 and 0.0735.

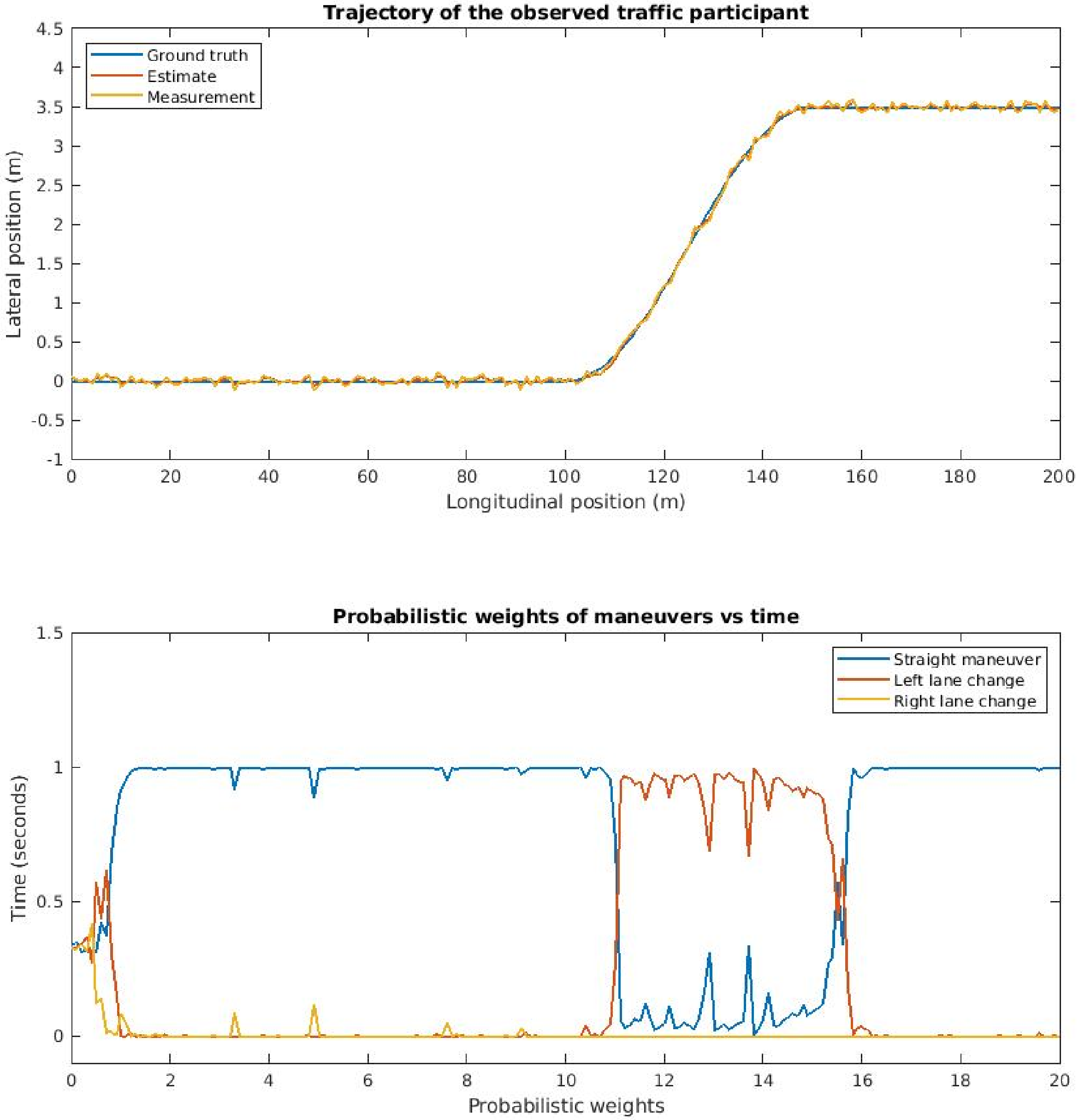


Figure 7.2: IMM based maneuver identification estimates and probabilistic weights under prolonged observations

Table 7.4: AMM and IMM approach comparison under prolonged observation

Behavior identifier	Detection time straight to left LC (s)	Detection time left LC to straight (s)	RMSE x position (m)	RMSE y position (m)
IMM based	1.1s	0.7s	0.0539m	0.0298m
AMM based	1.7s	1.1s	0.0778m	0.0735m

7.4 IMM for behavior identification

The IMM developed above was then extended to identify five different behaviors: straight motion, and two different behaviors each for left and right lane change maneuvers. The straight behavior was modeled with the same constant longitudinal velocity motion model with zero lateral velocity. The lane changes employed sinusoidal lateral motion. For each lane change maneuver, one behavior utilized 50 meters maneuver length, whereas the second one is realized with 100 m maneuver length. The initial conditions for the scenario were kept identical to the last use case. However, the IMM based identifier was tested against two use case scenarios of 25 seconds each. First scenario was similar to last use case, where the traffic participant drives straight for the first 10 seconds, performs a short left lane change maneuver (50 m maneuver length) and then keeps going straight for another 10 seconds. In the second scenario, the traffic participant drives straight for 10 seconds, performs a long left lane change maneuver (100 m length) lasting 10 seconds, and then keeps going straight for another 5 seconds.

The transition probabilities in the IMM for the five behaviors were chosen as

$$\Lambda_{tras} = \begin{bmatrix} 0.97 & 0.075 & 0.0075 & 0.0075 & 0.0075 \\ 0.027 & 0.97 & 0.001 & 0.001 & 0.001 \\ 0.027 & 0.001 & 0.97 & 0.001 & 0.001 \\ 0.027 & 0.001 & 0.001 & 0.97 & 0.001 \\ 0.027 & 0.001 & 0.001 & 0.001 & 0.97 \end{bmatrix} \tag{7.19}$$

The underlying order of the transition probabilities in Eq. 7.19 are as follows: straight maneuver, short left lane change, long left lane change, short right lane change and long right lane change. Hence, the top row corresponds to the condition in which the traffic participant is performing the straight maneuver in the last cycle. The bottom row corresponds to the case where the traffic participant is performing a long right lane change in the last cycle. It should be noted that the transition probabilities for a participant switching from any left lane change behavior to a right lane change behavior (or vice-versa) are chosen as very low (0.001). This is a valid assumption, since in a real driving scenario like this case the participant will be momentarily back to the straight maneuver between the switch. Also, for a given maneuver, the transition probabilities for the participant switching from one behavior to other is chosen very low. This is a strong assumption as in real driving conditions chances of a participant switching from one behavior to other depend entirely on the traffic conditions. However, this refinement is outside the scope of the current work. For all the lane change behaviors, the exit path leads to the straight maneuver as the typical default state for a participant. Again, these values were determined heuristically. In practice, these values should be determined from traffic statistics.

Figure 7.3 depicts the inference results of the IMM for the first use case. After the initial switching, the weights diverge after 1 second detecting the straight maneuver correctly. Once the participant initiates a short left lane change behavior at 10 seconds into

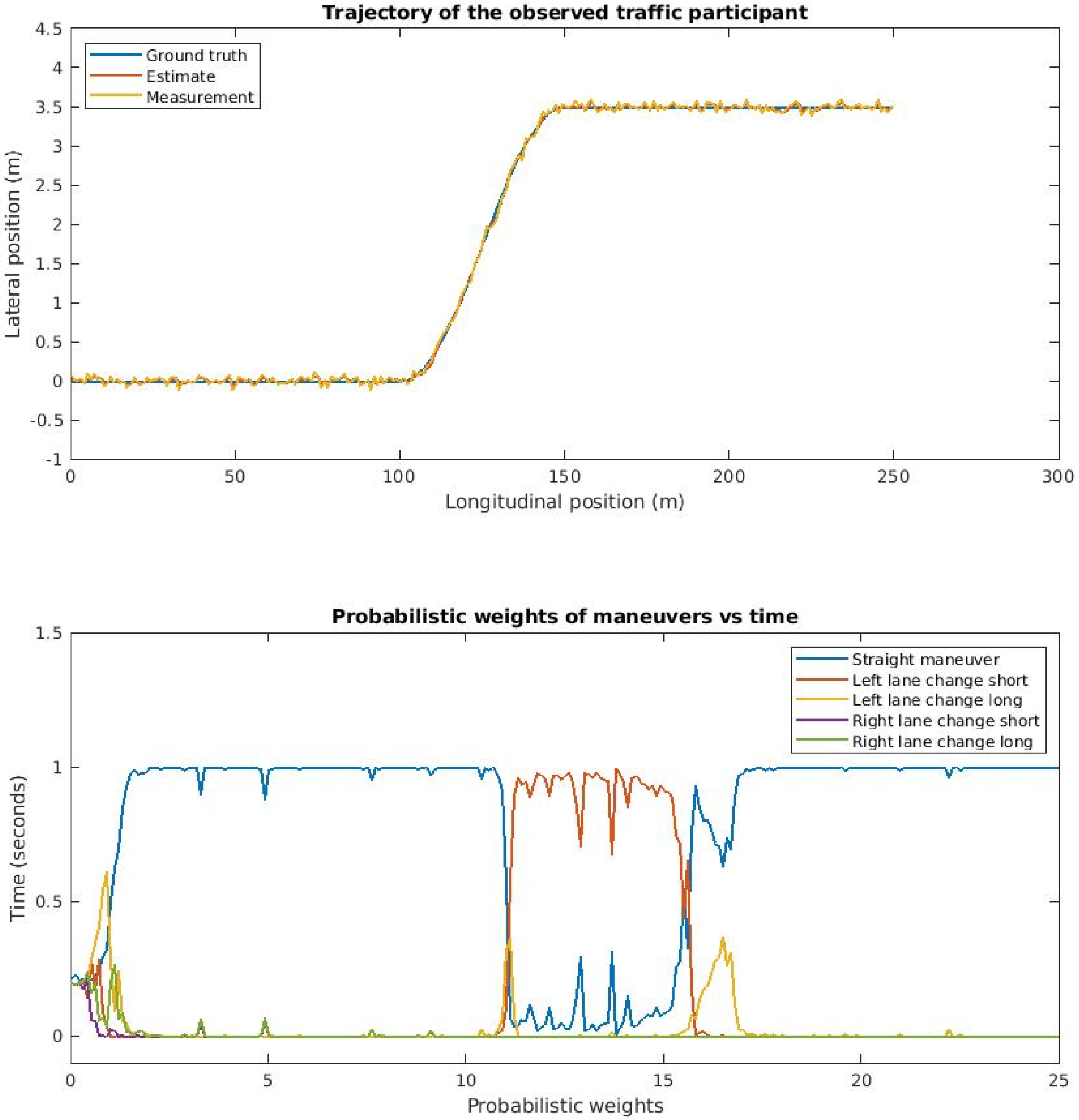


Figure 7.3: IMM based behavior identification estimates and probabilistic weights for short left lane change under prolonged observations

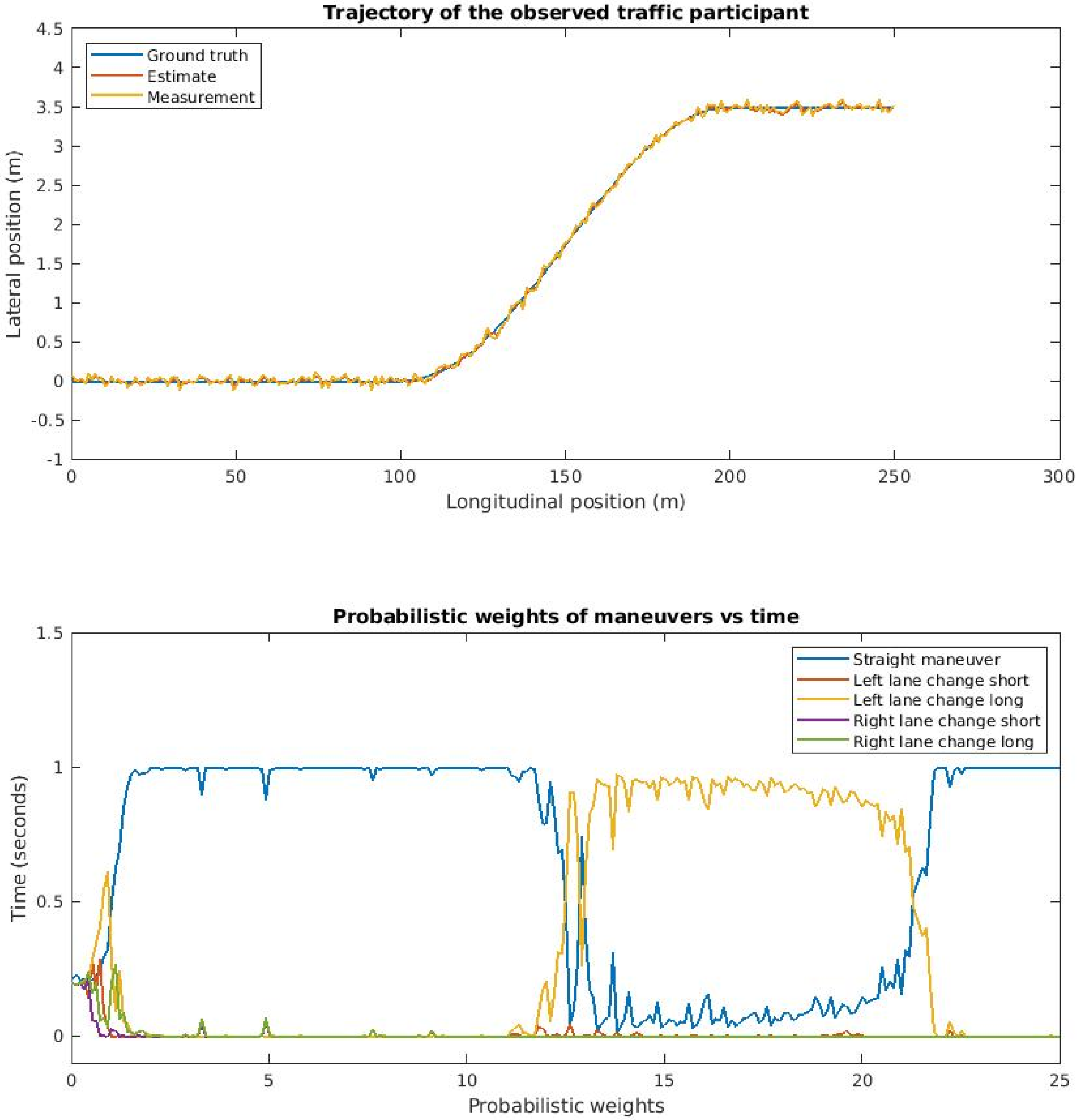


Figure 7.4: IMM based behavior identification estimates and probabilistic weights for long left lane change under prolonged observations

the simulation, the correct behavior is identified in 1.1 seconds (table 7.4), and the weights of the appropriate behavior rise quickly at 11^{th} second. As the behavior is completed, the model infers some ambiguity between the left lane change and the straight behaviors, before correctly identifying straight within 0.7 seconds (at time 15.7 seconds). Notice the spike in the long left lane change behavior weights as the maneuver is initiated by the traffic participant. This is due to the ambiguity between the two behaviors the participant is likely to execute. As a result, the innovations for both the elemental filters initially remain close, before diverging in the next few time steps. Interestingly, after the behavior completion, the weights of the long left lane change behavior rise again. This is likely due to the fact that the *a priori* estimates of the straight and the long left lane change elemental filters were in the vicinity so as to result in similar innovations. The upper plot shows how closely the estimates are tracking the measurements and the ground truth. The RMSE error between the ground truth and the estimated position is 0.0498 m and 0.0298 m for x and y positions respectively.

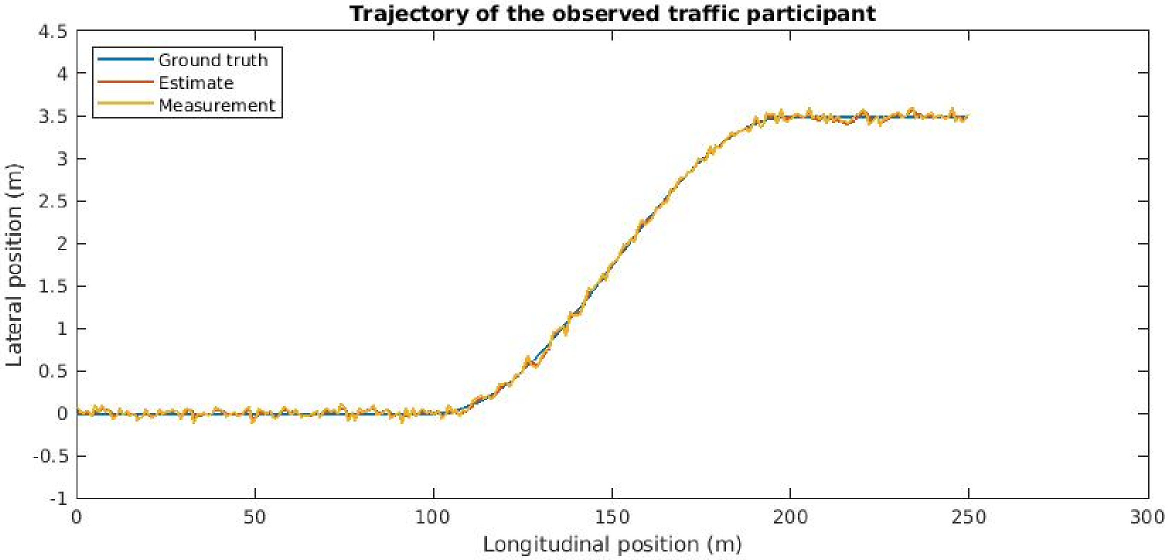
Figure 7.4 shows the inference results of the IMM for the second use case. After the initial switching, the weights diverge at about 1 second similar to the first use case, detecting the straight maneuver correctly. Once the participant initiates a long left lane change behavior, the filter sees some ambiguity between the straight behavior and the long left lane change behavior for a short period. This is intuitively expected as the longer left lane change behavior includes an extended almost straight traversal at the beginning. As a result the left lane change behavior weights rise gradually before corectly identifying it 3 seconds into the behavior (Table 7.4). As the behavior is completed, the model identifies the straight behavior correctly within 1.2 seconds of reaching the center of the lane. This is acceptable as the lane change behavior becomes almost linear at this stage. The RMSE between the ground truth and the estimated position is 0.0497 m and 0.0302 m for x and y positions respectively. It should be noted that there is no ambiguity between the short and the long lane change behavior at the initiation of the maneuver in this use case scenario. This leads to an important conclusion: when a short lane change is performed, the IMM based behavior might predict a longer lane change for a short period, which may lead to underestimation of risks initially.

Table 7.5: IMM based behavior identification performance for the two behaviors

Behavior	Detection time straight to left LC (s)	Detection time left LC to straight (s)	RMSE x position (m)	RMSE y position (m)
Short LC	1.1s	0.7s	0.0498m	0.0298m
Long LC	3.0s	1.2s	0.0497m	0.0302m

There is a reason why an asymmetric motion model (constant longitudinal velocity motion with zero lateral velocity) was chosen for the straight behavior versus a symmetric constant velocity motion model. Figure 7.5 depicts the performance of the IMM with the latter. As the sinusoidal waveform becomes almost linear towards the middle of the amplitude transition the influence of the straight maneuver weights is demonstrated relatively higher during the lane change behavior. Whereas the tracking performance as well as the identification of weights for the behavior remain adequate, weights of the straight behavior have an increased influence in the prediction and may lead to an incorrect prediction in the middle of the maneuver.

Remark. *It should be noted that although a constant longitudinal velocity motion model with zero lateral velocity works well for our use case, the weights of the straight motion exhibit a delay in rising after a lane change behavior is completed. An improved performance could be obtained with a constant longitudinal acceleration motion model with zero lateral velocity. It was observed that the weights of the straight behavior rise faster towards the end of the lane change behavior in that case.*



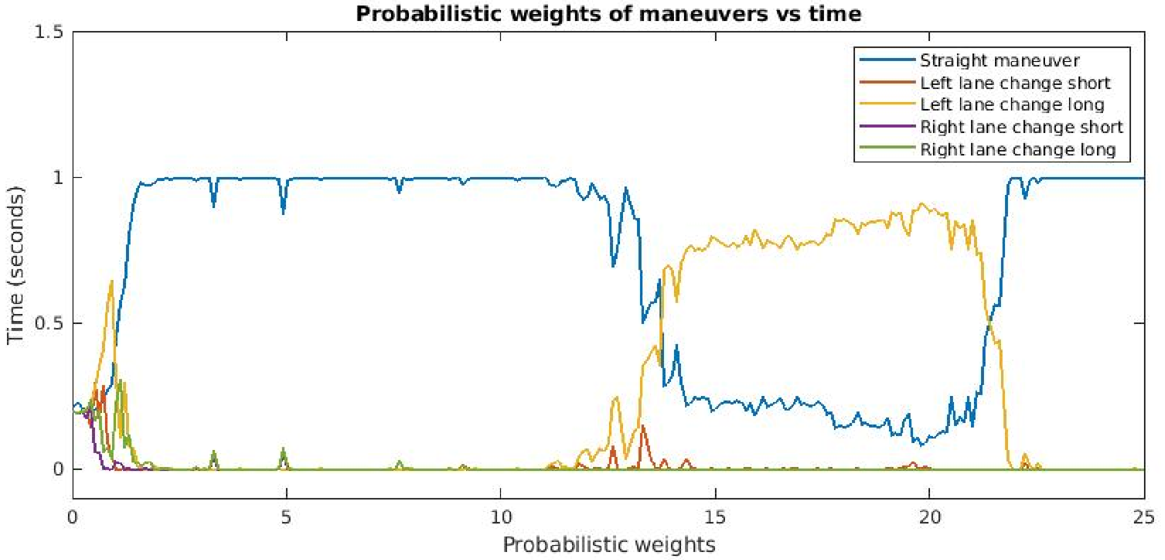


Figure 7.5: IMM based behavior identification with constant velocity motion model for straight behavior

7.5 Validation of the maneuver identification model

The validation for the maneuver identification model was carried out using a trajectory extracted from the highD dataset [43]. The following section will provide the overview of this dataset with the subsequent section discussing the results of the behavior identification module.

7.5.1 highD dataset overview

The highD dataset, short for Highway Drone dataset contains naturalistic trajectories of vehicles collected using a high resolution (4K) drone-based camera over German highways. The dataset consists of trajectories of 110,500 different vehicles recorded over 147 hours from six different locations around Cologne, Germany. The records include both light as well as heavy traffic conditions and feature multiple lanes in both driving directions. The claimed position accuracy is of at least 10 cm for about 400 meters stretch of vehicle observations. Apart from trajectories, the dataset also provides enhanced information such as class of the vehicle, information about the surrounding vehicles, time headway (THW), distance headway (DHW) as well as the Time-to-Collision (TTC) along with the lane change information. Further, the dataset also provides some MATLAB and Python based tools that aid in visualization (Fig. 7.6). The dataset comes with 60 different record-

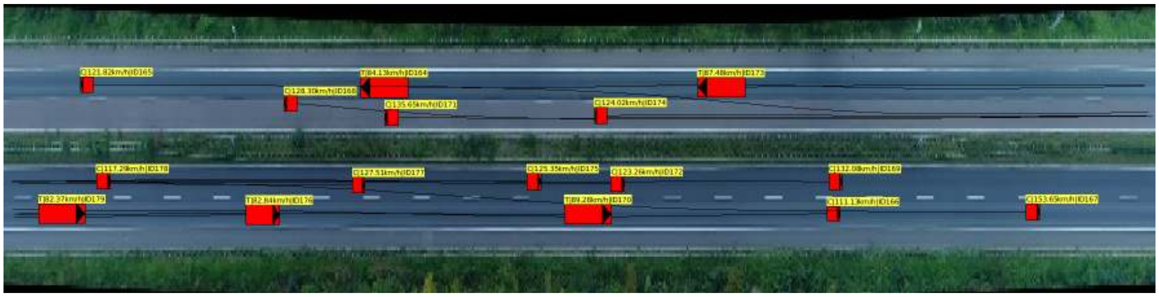


Figure 7.6: Visialization of highD dataset with vehicle 167 in *track 01* trajectories

ings calls as *tracks*. Each track is in a *csv* format containing vehicle trajectories information in comma separated values.

7.5.2 Evaluation using a highD dataset trajectory

From the dataset, a trajectory of vehicle 167 from the *track 01* was chosen for evaluating the developed IMM based behavior identification model. The measurements are available at a frame rate of 25 per second. The traffic participant is a vehicle of class 'Car' and its trajectory has a minimum longitudinal velocity of 36.38 m/s and maximum velocity of 43.46 m/s. In the travel distance of 410 m, it performs a single lane change of maneuver length of approximately 250 m, executed 1.5 seconds into the scene. For inference, an IMM with two different lane change behaviors of 200 m and 300 m maneuver lengths were implemented for each left and right lane change maneuver.

For evaluation, the ground truth was chosen as the smoothed trajectories provided by the dataset and the measurements were generated by adding white Gaussian noise of variance 0.0025 to this ground truth. The initial conditions of the filter were set as the initial conditions observed in the dataset ($x = 6.005$ m, $v_x = 36.38$ m/s, $y = 22.64$ m and $v_y = 0$ m/s). The process noise for the straight elemental filter was assumed as diag([0.01 0.01 0.01 0.01]), whereas the process noise of the lane change behavior filters were assumed as diag([0.01 0.01 0.01 0.01 1]). The measurement noise for all the filters was matched with the added Gaussian noise and assumed as diag([0.0025 0.0025]). This is a valid assumption as the statistics of measurement noise can be estimated from sample measurements.

Figure 7.8 shows the results of inference of the IMM with the identified vehicle and table 7.5.2 depicts the detection times as well as the RMSE for the identifier. The corresponding weights for the behavior start rising 1.4 seconds into the maneuver, with the behavior completely identified 1.7 seconds into the maneuver. The RMSE between the

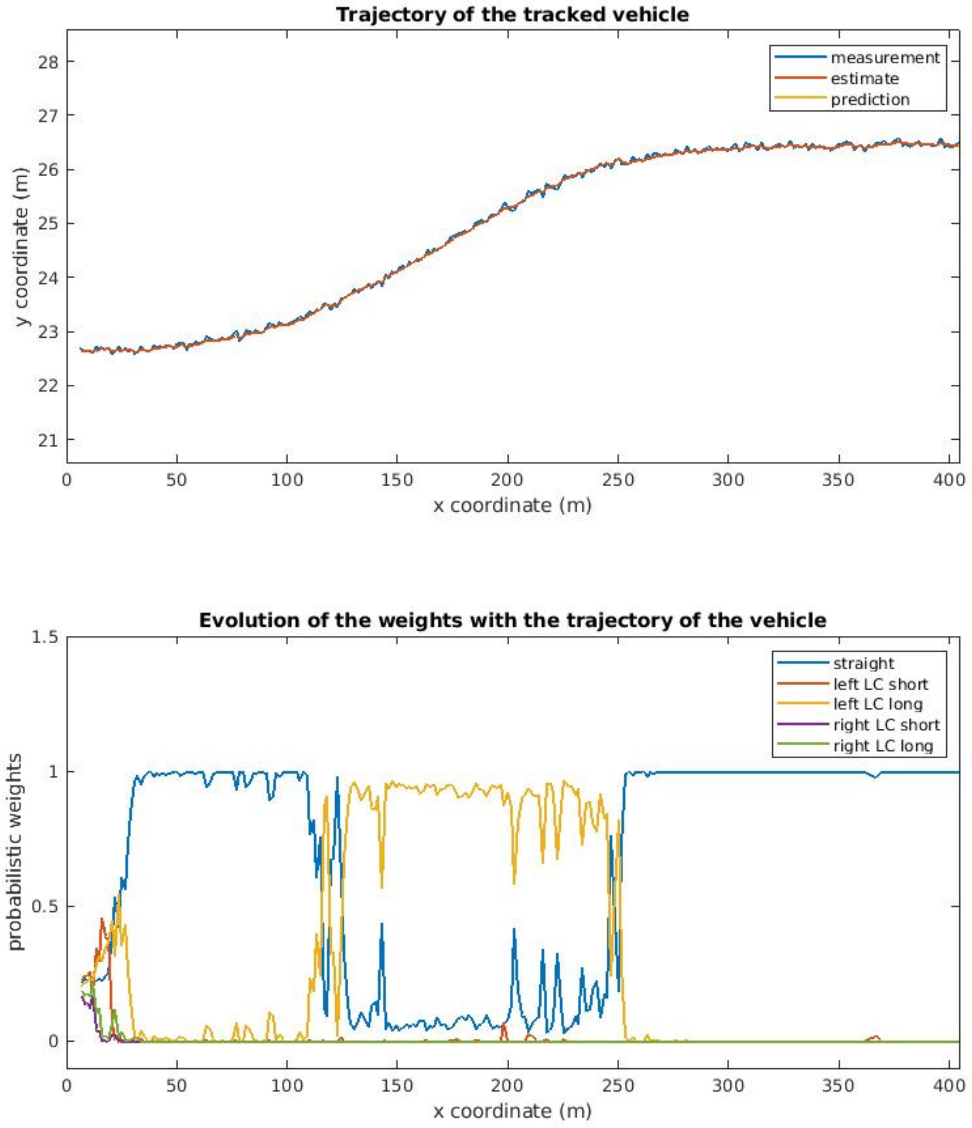


Figure 7.7: IMM performance with highD dataset trajectory

ground truth of the vehicle trajectory and the estimated position of the vehicle is 0.1005 m in the x direction, and 0.0283 m in the y direction.

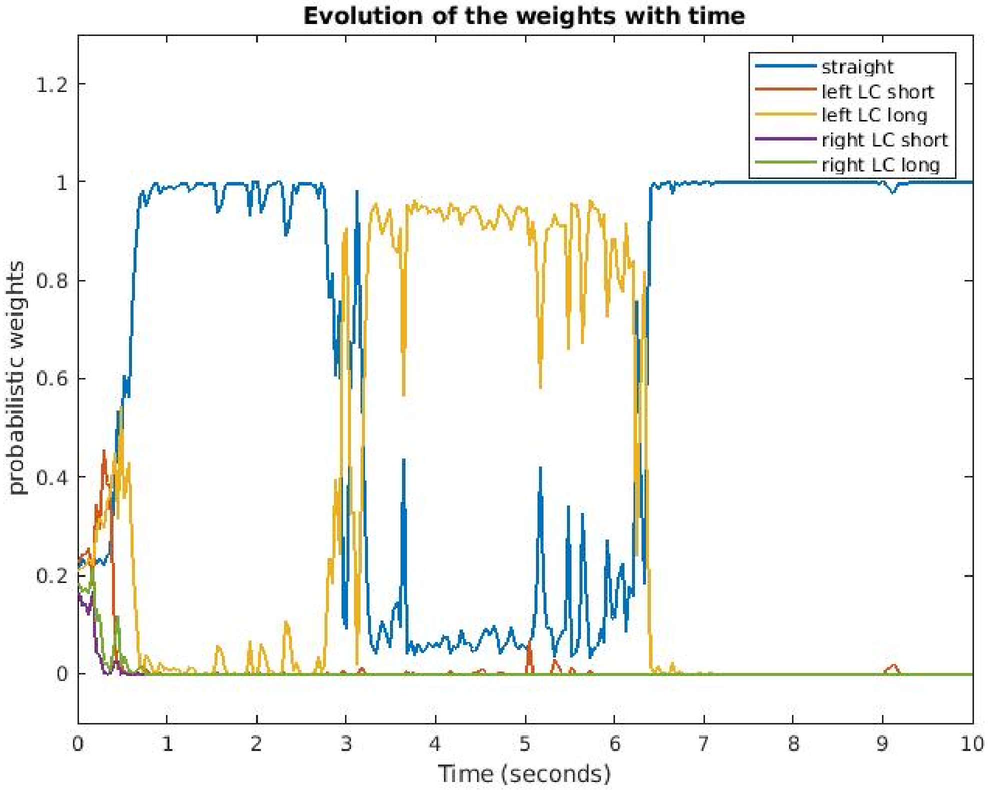


Figure 7.8: IMM performance with highD dataset trajectory against time

Table 7.6: IMM based behavior identification on a trajectory from HighD dataset

Behavior identifier	Detection time straight to left LC (s)	RMSE x position (m)	RMSE y position (m)
IMM based	1.7s	0.1005m	0.0283m

7.6 Conclusion

Interacting multiple model (IMM) filters were adapted for behavior identification and evaluated in this chapter. As the reset and mixing is part of the inherent framework of the IMM hence makes them suitable for representing the behaviors of the traffic participants as they are able to handle transitions between maneuvers in contrast to an AMM. Furthermore, the fact that they take transition probabilities of modes into account, allows them to integrate with discrete stochastic approaches to represent hybrid systems. IMMs were evaluated with the measurements generated from a single track vehicle model, and analyzed with varying process noise. Here, the intuition of selecting the transition probabilities for the IMM was also described. Although AMMs were found to be marginally superior in the identification of a single maneuver, for prolonged observations, IMMs exhibited their superior potential. Their performance was then analyzed with simulated use case scenarios. Finally, their performance was validated with a realistic trajectory from highD traffic dataset.

Chapter 8

Probabilistic Framework Implementation with IMM

This chapter describes the implementation and evaluation of the proposed interaction-aware Dynamic Hierarchical Bayesian Network using IMM. The challenges addressed in this chapter are, reformulation of the proposed framework to be used with an IMM and evaluating the performance of the framework. In section 4.4, we presented a generic approach for representing the driver behavior using various stages of maneuver. While that approach was more suitable for implementation with AMMs, using this approach with an IMM requires a bit of reformulation, which will be elaborated in the next two sections. For the sake of readability, some portions of the formulation will be reiterated below. The resulting IMM will be context-aware, hence, it is occasionally referred in the text as Contextual Interacting Multiple Model (CIMM) filter. Thereafter, the framework will be demonstrated by implementing it for a simple use case scenario. Finally, we will compare it's performance with the IMM based behavior identifier implemented in the last chapter.

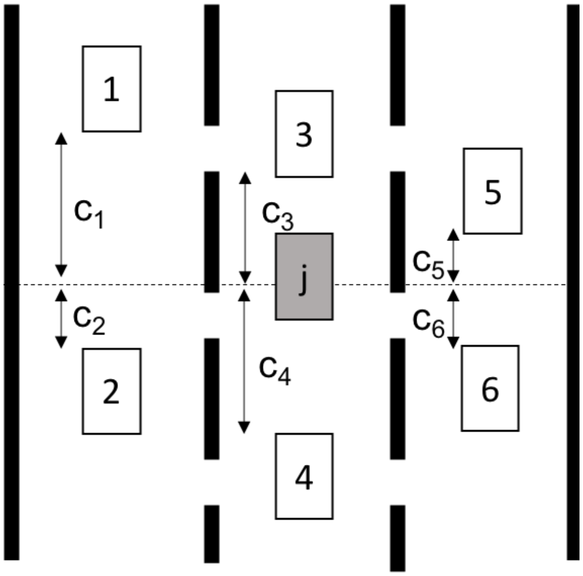


Figure 8.1: Traffic scenario as an example. j is the subject vehicle being observed, *1 - 6* are the relevant traffic participants for j, with c_1 - c_6 forming the relations for the context

8.1 Probabilistic framework revisited

Recalling from section 4.3, given the vector of possible behaviors and their associated paramters, the core challenge for risk analysis based on behavior prediction for a traffic participant consists of two stages, i.e. for every traffic participant j and driving style i:

1. determine $\mathbb{P}({}^{q}b_k^j)$, where $q = i + I \cdot (i - 1)$.

2. determine the trajectory $\hat{\mathbf{x}}_{k+1}^{j,i}$.

The solution to the above challenge can be addressed by utilizing a Hierarchical Dynamic Bayesian Network (HDBN) in the following way. Let $\mathbf{x}_k^j = [x_k^j\ y_k^j\ v_{x,k}^j\ v_{y,k}^j\ \psi_k^j]^T$ represent the base states for the jth traffic participant at the kth time step, i.e. longitudinal and lateral positions and velocities as well as the yaw angle. The quantity $\mathbf{z}_k^j = [\tilde{x}_k^j\ \tilde{y}_k^j\ \tilde{v}_{x,k}^j\ \tilde{v}_{y,k}^j\ \tilde{\psi}_k^j]^T$ represents the observed states for the j^{th} traffic participant at the k^{th} time step. As described earlier, we will we employ the concept of context vector that establishes the necessary spatio-temporal relationships between a subject vehicle and the relevant traffic participants around it for every traffic participant, i.e. C_k^j for the j^{th} traffic participant at the k^{th} time step. Furthermore, let $\mathscr{G}_k^j$ be a binary state at time k indicating that the j^{th} traffic participant has accepted a gap in a traffic scenario for maneuver execution.

Based on certain conditional independence between quantities, the joint probability density function for the traffic scenario can then be decomposed as follows:

$$
\begin{aligned}
\mathbb{P}(\hat{X}_{k+1}, \hat{X}_k, C_{k+1}, \mathscr{G}_{k+1}, \mathscr{G}_k, \mathscr{M}_{k+1}, \mathscr{M}_k, \mathscr{D}_{k+1}, \mathscr{D}_k, \mathscr{B}_{k+1}, \mathscr{B}_k, Z_{k+1}) \\
= \mathbb{P}(\hat{X}_k)\mathbb{P}(\mathscr{G}_k)\mathbb{P}(\mathscr{M}_k)\mathbb{P}(\mathscr{D}_k)\mathbb{P}(\mathscr{B}_k) \\
\mathbb{P}(\hat{X}_{k+1}|\hat{X}_k, \mathscr{B}_k)\mathbb{P}(C_{k+1}|X_{k+1}) \\
\mathbb{P}(\mathscr{G}_{k+1}|\mathscr{G}_k, C_{k+1}, \mathscr{D}_k)\mathbb{P}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1}) \\
\mathbb{P}(\mathscr{B}_{k+1}|\mathscr{M}_{k+1}, \mathscr{D}_k, \mathscr{B}_k, \mathscr{G}_{k+1}) \\
\mathbb{P}(Z_{k+1}|\hat{X}_{k+1})\mathbb{P}(Z_{k+1}|\mathscr{D}_{k+1}) \quad (8.1)
\end{aligned}
$$

The resulting HDBN can be respresented as depicted in Fig. 8.2. The application of the introduced abstract framework above to a particular traffic scenario will be illustrated via the example of multi-lane driving on a highway. The presented framework and its employed terms, however, are of such a generic nature that they can be easily adapted to other urban as well as highway driving scenarios. For the exemplary application, we assume three maneuvers (left lane change, LL, right lane change, LR, and straight driving, S), two driver types (aggressive, d_1, and passive, d_2) and a context vector based upon the relative longitudinal distances shown in Fig. 8.1. Hence, the introduced sets yield

$$
\begin{aligned}
\mathscr{M} &:= [m_{LL}, m_{LR}, m_S], \quad \mathscr{D} := [d_1, d_2] \\
C_k^j &:= [c_1^j\ c_2^j\ c_3^j\ c_4^j\ c_5^j\ c_6^j]
\end{aligned}
$$

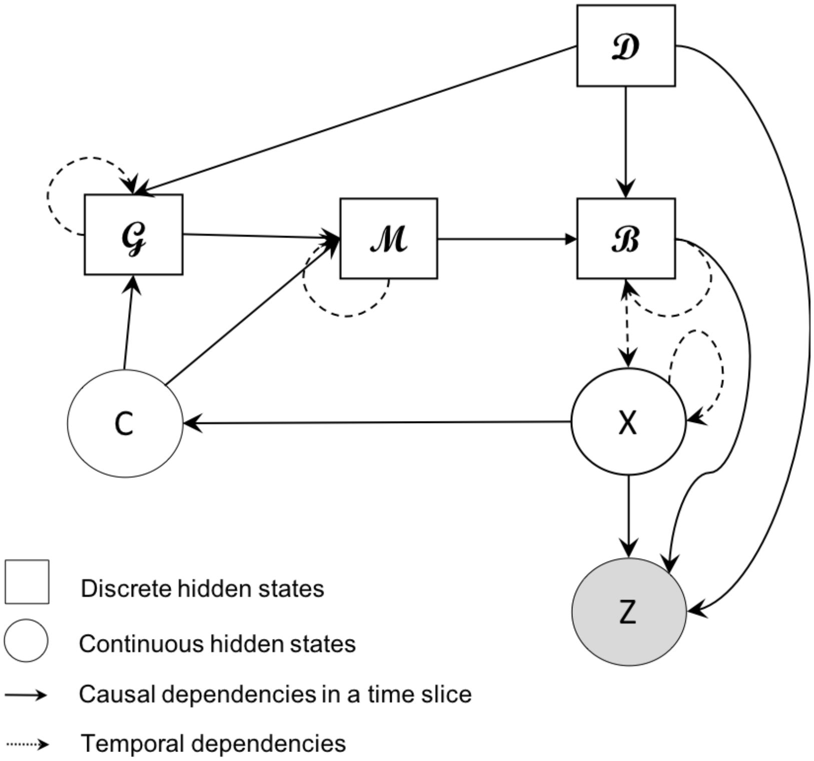


Figure 8.2: Hierarchical Dynamic Bayesian Network representing a traffic participant

Based on our example, our behavior vector can be represented as per Fig. 8.3

In chapter 7 we assumed the transition probabilities for the IMM for behaviors heuristically. In a typical system, these transition probabilities are determined from sample

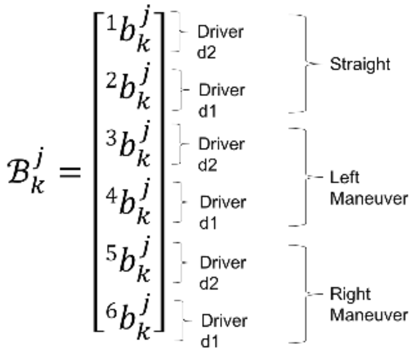


Figure 8.3: Behavior vector for a traffic participant j with two driver styles (d1 and d2) and with 5 behaviors, 2 for left and right lane change, and one for straight.

observations of the system under relevant conditions. However, when the traffic participants are represented as hybrid systems, and the modes of the system represent their behaviors for maneuvers, these probabilities cannot rely simply on predetermined values. The maneuvers and behaviors are typically influenced by the context of the traffic participant. Hence, in such case the transition probabilities have to be determined online. Following section will illustrate this with the exemplary implementations.

8.2 Representing a traffic participant using IMM

To realize this with IMM, we will have to employ a motion model for each of these behaviors as elemental filters in the IMM bank. Also, we will need the transition probability matrix Λ_{beh} for the behaviors, which in our case will be a 6 by 6 matrix:

$$\Lambda_{beh} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{15} & \lambda_{16} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} & \lambda_{24} & \lambda_{25} & \lambda_{26} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \lambda_{34} & \lambda_{35} & \lambda_{36} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} & \lambda_{44} & \lambda_{45} & \lambda_{46} \\ \lambda_{51} & \lambda_{52} & \lambda_{53} & \lambda_{54} & \lambda_{55} & \lambda_{56} \\ \lambda_{61} & \lambda_{62} & \lambda_{63} & \lambda_{64} & \lambda_{65} & \lambda_{66} \end{bmatrix} \tag{8.2}$$

where λ_{ab} is the predicted transition probability of the participant to execute the behavior b in the next time step, given that it is executing the behavior a in the current time step, such that $a, b \in [1, 6$. Also, for every row a for Λ_{beh},

$$\sum_{b=1}^{6} \lambda_{ab} = 1, \forall\, a \in [1, 6] \tag{8.3}$$

The probabilities of the behavior can then be predicted as:

$$\mathbf{p}(\mathscr{B}^j_{k+1}) = \Lambda'_{beh} \cdot \mathbf{p}(\mathscr{B}^j_k) \tag{8.4}$$

To utilize this transitional probability matrix with IMM, we need to find out the coefficients λ_{ab}. In reality, all the behaviors are coupled with each other and determining all the coefficients of the transition probability can be complex. However, using certain assumptions below, their calculation can be simplified.

i. If a traffic participant is executing a particular non-default maneuver, it will not switch to another mid-way. It may choose to abort the maneuver and come back to the default maneuver, i.e. straight, or return to the default maneuver after it has executed the current behavior in entirety.

ii. If a traffic participant chooses a maneuver with a particular driving style, it is expected to complete it without switching the driving style.

iii. The only way a participant switches to another non-default behavior is after it arrives back at the default behavior.

With the above assumptions, the transition probability matrix Λ_{beh} for the traffic participant looks like:

$$\Lambda_{beh} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{15} & \lambda_{16} \\ \lambda_{21} & \lambda_{22} & 0 & \lambda_{24} & 0 & \lambda_{26} \\ \lambda_{31} & 0 & \lambda_{33} & 0 & 0 & 0 \\ 0 & \lambda_{42} & 0 & \lambda_{44} & 0 & 0 \\ \lambda_{51} & 0 & 0 & 0 & \lambda_{55} & 0 \\ 0 & \lambda_{62} & 0 & 0 & 0 & \lambda_{66} \end{bmatrix} \tag{8.5}$$

Following the order of the behaviors in our behavior vector $\mathscr{B}_k^j$, the above Λ_{beh} can be read as follows. Given that the participant is in the default straight maneuver, there is a λ_{11} chance that it will continue with the same behavior, λ_{12} chance that it may switch to an aggressive straight behavior, λ_{13} chance that it may switch to a passive left lane change and so on. On the other hand, if the traffic participant is executing an aggressive (driver d_1) right lane change behavior, there is a λ_{66} probability that it will continue executing the behavior, and λ_{61} probability that it will return to an aggressive straight maneuver. The predictions for the other lane change behaviors can be interpreted in a similar way.

It is important to note that these transition probability matrix coefficients are influenced by factors such as the context of the traffic participant, the situation they are in, their mission goals as well as their opportunistic inclination. Hence these coefficients need to be determined at run-time and cannot be entirely predetermined. Every subsequent explanatory section below contains exemplary implementations of different terms in Eqn. (8.1). The relevant sections also illustrate a way to determine these coefficients.

1. **Motion prediction, $\mathbf{p}(\hat{X}_{k+1}|\hat{X}_k, \mathscr{B}_k)$:** The predicted motion of a traffic participant depends on its current states, the driver type, and the maneuver that the driver intends to perform. The driver type combined with the intended maneuver determine the corresponding parameters for the trajectory generation of the participant as contained in the behavior set $\boldsymbol{B}_k^j$. The identified trajectory can be used as the underlying model for propagating the states of every traffic participant.

2. **Context, $\mathbf{p}(C_{k+1}|\hat{X}_{k+1})$:** The context experienced by each traffic participant aids in establishing the local traffic situation. In its simplest case, it can be derived directly from the base states by creating a vector reflecting the spatio-temporal relations to the neighboring/near-by other traffic participants, for instance distances, velocity differences, etc. Although the context in our example below arises from a linear com-

bination of distances, it can be expanded to include velocity or to any general state relationship that can aid in predicting a driver's intent. Since we assume our base states to be Gaussian perturbed, our context vector represents a mixture of Gaussian terms.

3. **Gap acceptance policy, $\mathbf{p}(\mathscr{G}_{k+1}|\mathscr{G}_k, C_{k+1}, \mathscr{D}_k)$**: Given a current context around a vehicle, the decision for gap acceptance varies with the driver. In addition to the conditioning on context and driver, we also include a dependence of gap acceptance on the decision in the previous step to include memory. In an exemplary realization, the gap acceptance pdf could be defined as

$$\mathbf{p}(\mathscr{G}^j_{k+1} \mid \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k) = \begin{bmatrix} \mathbb{P}(\mathscr{G}^j_{k+1} = 0 \mid \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k) \\ \mathbb{P}(\mathscr{G}^j_{k+1} = 1 \mid \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k) \end{bmatrix} \tag{8.6}$$

where $\mathscr{G}^j_k = 0$ denotes gap rejection for participant j while $\mathscr{G}^j_k = 1$ stands for gap acceptance. The terms on the right hand side of Eq. (8.6) can be further augmented as

$$\begin{aligned} &\mathbf{p}(\mathscr{G}^j_{k+1} \mid \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k) \\ &\quad \propto \Big[\mathbb{P}(\mathscr{D}^j_k = d_1) \cdot \mathbf{p}\big(\mathscr{G}^j_{k+1} | \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k = d_1\big) \\ &\qquad + \mathbb{P}(\mathscr{D}^j_k = d_2) \cdot \mathbf{p}\big(\mathscr{G}^j_{k+1} | \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k = d_2\big)\Big] \end{aligned} \tag{8.7}$$

The first conditional probability on the right hand side of Eq. (8.7) can be expanded as

$$\mathbf{p}\big(\mathscr{G}^j_{k+1} | \mathscr{G}^j_k, C^j_{k+1}, \mathscr{D}^j_k = d_1\big) = \begin{bmatrix} 1 - g_{ag}(C^j_{k+1}) \\ g_{ag}(C^j_{k+1}) \end{bmatrix} = \begin{bmatrix} g_{NAd1} \\ g_{Ad1} \end{bmatrix} \tag{8.8}$$

where g_{NAd1} and g_{Ad1} are the resultant probabilities that the driver d_1 has accepted and not accepted the gap, whereas $g_{ag}(\cdot)$ is a gap function associated with the aggressive driver. Here, the context for the specific traffic participant is accepted as an input, returning an associated probability of gap acceptance. This could, for instance, be realized via a step function, i.e.

$$g_{ag}(\boldsymbol{C}_{k+1}^{j}) = \begin{cases} 0 & (c_1 + c_2) < \tau_{ag} \wedge \ (c_5 + c_6) < \tau_{ag} \\ 1 & \text{otherwise} \end{cases} \tag{8.9}$$

Here, τ_{ag} is a distance threshold for an aggressive driver for gap acceptance. To allow for a more gradual decision transition, a logistic function could be utilized alternatively, yielding

$$g_{ag}(\boldsymbol{C}_{k+1}^{j}) = \frac{1}{1 + e^{-\zeta(gap - \tau_{ag})}}$$

with ζ being a tuning factor and with gap being a placeholder for any of the relevant observed gaps. The second conditional probability on the right hand side of Eq. (8.7) can be treated in a similar fashion by using the corresponding gap functions for a passive driver. Let g_{NAd2} and g_{Ad2} be the respective probabilities of the driver 2 accepting or not accepting a gap. Eq. (8.7) hence becomes:

$$\mathbf{p}(\mathscr{G}_{k+1}^{j} \mid \mathscr{G}_{k}^{j}, \boldsymbol{C}_{k+1}^{j}, \mathscr{D}_{k}^{j}) \propto \mathbb{P}(d_1) \cdot \begin{bmatrix} g_{NAd1} \\ g_{Ad1} \end{bmatrix} + \mathbb{P}(d_2) \cdot \begin{bmatrix} g_{NAd2} \\ g_{Ad2} \end{bmatrix} \tag{8.10}$$

$$\mathbf{p}(\mathscr{G}_{k+1}^{j} \mid \mathscr{G}_{k}^{j}, \boldsymbol{C}_{k+1}^{j}, \mathscr{D}_{k}^{j}) \propto \begin{bmatrix} \mathbb{P}(d_1) \cdot g_{NAd1} + \mathbb{P}(d_2) \cdot g_{NAd2} \\ \mathbb{P}(d_1) \cdot g_{Ad1} + \mathbb{P}(d_2) \cdot g_{Ad2} \end{bmatrix} \tag{8.11}$$

The normalization term, η_{gap} in the above equation will be the sum of both the elements of Eq.(8.11), i.e.:

$$\eta_{gap} = \sum_{g=0,1} \mathbb{P}(\mathscr{G}^j_{k+1} = g \mid \mathscr{G}^j_k, \boldsymbol{C}^j_{k+1}, \mathscr{D}^j_k)$$
$$= \mathbb{P}(d_1) \cdot g_{NAd1} + \mathbb{P}(d_2) \cdot g_{NAd2} + \mathbb{P}(d_1) \cdot g_{Ad1} + \mathbb{P}(d_2) \cdot g_{Ad2} \quad (8.12)$$

After normalizing, let the conditional probability be represented as:

$$\mathbf{p}(\mathscr{G}^j_{k+1} \mid \mathscr{G}^j_k, \boldsymbol{C}^j_{k+1}, \mathscr{D}^j_k) = \begin{bmatrix} g_{NA} \\ g_A \end{bmatrix} \quad (8.13)$$

where g_{NA} and g_A are the probabilities that the gap is accepted or not accepted respectively, given the context.

It should be noted that these are only exemplary realization of the introduced gap functions only utilizing distance relations in the context. A more sophisticated gap processing based upon relative velocities (or other relations) could certainly be employed without loss of generality. The context vector would require modifications according to this case.

4. **Maneuver policy,** $\mathbf{p}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1})$: The maneuver policy can formally be expanded similar to the gap acceptance policy, i.e.

$$\mathbf{p}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1}) = \prod_{j=0}^{J-1} \mathbf{p}(\mathscr{M}^j_{k+1}|\mathscr{M}^j_k, \mathscr{G}^j_{k+1}, C^j_{k+1}) \quad (8.14)$$

$$
\begin{aligned}
&\mathbf{p}\big(\mathscr{M}^j_{k+1} \mid \mathscr{M}^j_k, \mathscr{G}^j_{k+1}, C^j_{k+1}\big)\\
&\quad\propto \Big[\mathbb{P}(\mathscr{G}^j_{k+1} = 0)\mathbf{p}\big(\mathscr{M}^j_{k+1} | \mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 0\big)\\
&\qquad+\mathbb{P}(\mathscr{G}^j_{k+1} = 1)\mathbf{p}\big(\mathscr{M}^j_{k+1} | \mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 1\big)\Big]
\end{aligned}
\tag{8.15}
$$

with $\mathscr{G}^j_{k+1} = 0$ signifying that the driver did not accept the gap.

Here, no maneuver other than a nominal driving maneuver (for instance defined as straight) will be executed, irrespective of the driver type and the maneuver currently being executed. Hence, the conditional probability for this case in Eq. (8.15) reduces to

$$
\mathbf{p}\Big(\mathscr{M}^j_{k+1} \mid \mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 0\Big) = \begin{bmatrix} \mathbb{P}(m_S) \\ \mathbb{P}(m_{LL}) \\ \mathbb{P}(m_{LR}) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
$$

Similarly, the gap acceptance term in Eq. (8.15) can be realized as

$$
\mathbf{p}\Big(\mathscr{M}^j_{k+1} \mid \mathscr{M}^j_k, C^j_{k+1}, \mathscr{G}^j_{k+1} = 1\Big) = \begin{bmatrix} 0 \\ f_{LL}(C^j_{k+1}) \\ f_{LR}(C^j_{k+1}) \end{bmatrix}
$$

where $f_{(,)}(\cdot)$ is a maneuver function establishing a relationship between the context and the left and right lane maneuver, respectively. A simple exemplary implementation of such a function yields

$$
f_{LL}(C_{k+1}) = \frac{c_1 + c_2}{(c_1 + c_2) + (c_5 + c_6)} \tag{8.16}
$$

For countries with asymmetric lane changes, another parameter could be included in the maneuver function reflecting different weights for left and right. From Eq. (8.15), the maneuver policy for participant is:

$$\mathbf{p}(\mathscr{M}_{k+1}|\mathscr{M}_k, \mathscr{G}_{k+1}, C_{k+1}) = \begin{bmatrix} g_{NA} \\ g_A\, f_{LL} \\ g_A\, f_{RL} \end{bmatrix} \tag{8.17}$$

As the framework already factors the driver influence into the gap acceptance policy, the maneuver policy does not depend on the driver directly. Yet, this does not prohibit the consideration of the driver's influence in a maneuver decision explicitly if such an extension of the policy is desired in the future.

5. **Behavior, $\mathbf{p}(\mathscr{B}_{k+1}|\mathscr{M}_{k+1}, \mathscr{D}_k, \mathscr{B}_k)$:** As we assume that the maneuver and driver types can evolve independently for a specific traffic participant, they are addressed separately instead of jointly in the behavior space.

   The transition probability matrix for the behaviors can then be obtained using the maneuver policies in Eq. (8.17) and the probabilistic weights of the drivers from the last time step. For the sake of brevity, in the following equation, $d_1 = \mathbb{P}(D_k = d_1)$ and $d_2 = \mathbb{P}(D_k = d_2)$.

$$\Lambda_{beh} = \begin{bmatrix} \lambda_{11} & \eta_1\, g_{NA} & \eta_1 d_2 f_{LL} g_A & \eta_1 d_1 f_{LL} g_A & \eta_1 d_2 f_{RL} g_A & \eta_1 d_1 f_{RL} g_A \\ \eta_2 g_{NA} & \lambda_{22} & 0 & \eta_2 g_A f_{LL} & 0 & \eta_2 g_A f_{RL} \\ \lambda_{31} & 0 & \lambda_{33} & 0 & 0 & 0 \\ 0 & \lambda_{42} & 0 & \lambda_{44} & 0 & 0 \\ \lambda_{51} & 0 & 0 & 0 & \lambda_{55} & 0 \\ 0 & \lambda_{62} & 0 & 0 & 0 & \lambda_{66} \end{bmatrix} \tag{8.18}$$

where,

$$\eta_1 = \frac{1-\lambda_{11}}{g_{NA} + d_2 f_{LL} g_A + d_1 f_{LL} g_A + d_2 f_{RL} g_A + d_1 f_{RL} g_A} \tag{8.19}$$

$$\eta_2 = \frac{1-\lambda_{22}}{g_{NA} + g_A f_{LL} + g_A f_{RL}} \tag{8.20}$$

The non zero transition probabilities marked with λ_{ab} in the Eqn. 8.18 need to based on statistical data. The above transition probability matrix can now be used with the IMM for mixing and reinitialization stage. Also, the predicted behaviors of the traffic participant can then be determined using:

$$\mathbf{p}(\mathscr{B}^j_{k+1}|\mathscr{M}_{k+1}, \mathscr{D}_k, \mathscr{B}_k) = \Lambda'_{beh} \cdot \mathbf{p}(\mathscr{B}^j_k) \tag{8.21}$$

6. **Likelihood, $\mathbf{p}(Z_{k+1}|\hat{X}_{k+1})$:** When used with the IMM, the likelihood is calculated for all the elemental filters and the updates applied to calculate the estimates. Further, these likelihoods are then used to update the mode probabilities of different modes, which in our case are behaviors.

7. **Driver policy update, $\mathbf{p}(Z_{k+1}|\mathscr{D}_{k+1})$:** For every traffic participant, a bank of 6 (i.e. the number of behaviors) EKFs can be employed in the Interactive Multiple Model (IMM) filter setting to propagate trajectories and to estimate the likelihood of a particular behavior. The probabilistic driver type itself is a parameter in the propagation in between measurement intervals. It will be updated based upon the likelihood of the measurements for the employed behavior in the individual single model filters. The driver model probability then yields

   $$\mathbb{P}(d_2) = \frac{\mathbb{P}({}^1b^j_k) + \mathbb{P}({}^3b^j_k) + \mathbb{P}({}^5b^j_k)}{\sum_{q=1}^{M\cdot I} \mathbb{P}({}^q b^j_k)} \tag{8.22}$$

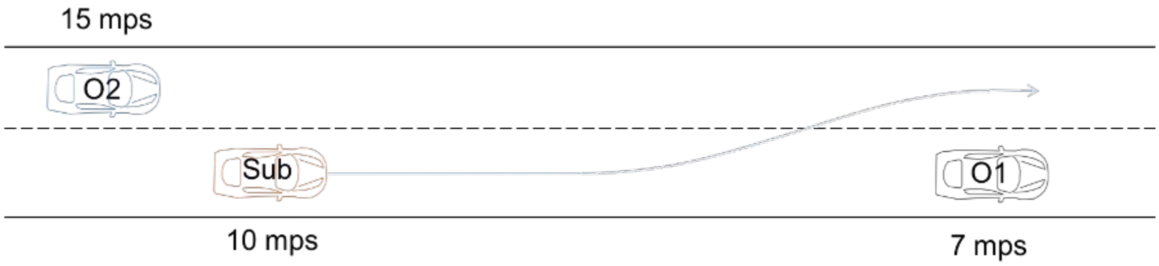


Figure 8.4: Simulation scenario with *Sub* being the observed vehicle, *O1* and *O2* are other traffic participants in the scene

8. **Filter equation:** The complete filter equation for the presented framework then yields

$$\begin{aligned}
&\mathbb{P}\big(\hat{X}_{k+1}, C_{k+1} \mid Z_{k+1}\big) \\
&\quad = \mathbb{P}(Z_{k+1}|\hat{X}_{k+1}) \;\; \mathbb{P}(Z_{k+1}|\mathscr{D}_{k+1}) \\
&\qquad \int_{\mathscr{G}_k,\mathscr{M}_k,\mathscr{D}_k,\mathscr{B}_k} \quad \mathbb{P}(\hat{X}_{k+1}, C_{k+1}, \mathscr{G}_{k+1}, \mathscr{M}_{k+1}, \mathscr{D}_{k+1}, \\
&\qquad\qquad\qquad \mathscr{B}_{k+1}|\mathscr{G}_k, \mathscr{M}_k, \mathscr{D}_k, \mathscr{B}_k, \hat{X}_k) \\
&\qquad\qquad\qquad \mathbb{P}(\mathscr{G}_k, \mathscr{M}_k, \mathscr{D}_k, \mathscr{B}_k, \hat{X}_k)
\end{aligned} \tag{8.23}$$

8.3 Simulation setup

To evaluate the filter, a simulation scenario as shown in Fig. 8.4 was considered. The scenario consists of two lanes with the observed vehicle (*Sub*) driving in the right lane at a speed of 10 m/s. When encountering a slower vehicle *O1* ahead moving at a speed of 7 m/s, in the left lane, a faster vehicle *O2* is approaching at a faster speed of 15 m/s. From here the use case bifurcates into two potential use cases depending on the subject vehicle behaviors: (i) the *Sub* vehicle can accelerate and perform an aggressive left lane change cutting into the path of the faster vehicle $O2$; (ii) or it can let the $O2$ vehicle pass and then perform a lane change maneuver. The first behavior would be classified as aggressive while the latter would be considered as a passive driver behavior.

The complete simulation run consisted of 25 seconds. The center of the right lane on the left hand side was assumed as the origin for the coordinate frame. The subject vehicle starts from the origin, i.e $x = 0$ m and $y = 0$ m with a longitudinal velocity $v_x = 10$ m/s and lateral velocity $v_y = 0$ m/s. The vehicle O1 starts at $y = 0$ m, with the velocities $v_x = 6.67$ m/s and $v_y = 0$ m/s. The vehicle O2 starts at $y = 3.5$ m (left lane) with velocities $v_x = 15$ m/s and $v_y = 0$ m/s. The initial x position of O1 and O2 will be described along with the use cases in the next section. The frequency of measurements was chosen as 10 Hz generated by simulating the motions of the subject vehicle changing lanes and adding Gaussian white noise to it. For simplicity of demonstration, the inference was run only for the subject vehicle and a simulated context provided for the filter. In practice, the inference should be running for every vehicle in the scene with the context extracted using a helper function from the estimated states of all the vehicles in the scene. Predetermined maps can also be utilized in calculating the context. The context vector chosen for the use case consists of 8 elements,

$$C_k^j = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 \end{bmatrix} \tag{8.24}$$

The elements (c_1 to c_6) are the longitudinal distances from the longitudinal center of the observed vehicle to the vehicles in the front and rear in the current as well as adjacent lanes. (c_7 and c_8) are the relative velocities from the succeeding vehicles (i.e., vehicles behind) in the adjacent lanes. For the case of no preceding or succeeding vehicle being present, the relevant distance vector was artificially assumed as a large value (here 100 m).

A simplified version of the model was considered for this simulation scenario as it had only two lanes. Two driver types were considered, *aggressive* and *passive* executing two potential maneuvers, straight and left lane change. As a result, the behavior matrix consisted of four behaviors as following:

i) **Passive straight behavior:** The longitudinal motion was modeled with constant velocity whereas the lateral motion was assigned with zero lateral velocity motion models. The behavior parameter (${}^{1}b_{k}^{j}$) for this maneuver was chosen to contain only acceleration (being zero for this behavior).

ii) **Aggressive straight behavior:** The longitudinal motion was modeled with constant acceleration whereas the lateral motion maintained zero lateral velocity. The behavior parameter for this behavior similarly reduces to acceleration (ideally being a value suggesting a strong increase or decrease in speed).

iii) **Passive left lane change behavior:** The longitudinal motion was modeled with constant acceleration whereas the lateral behavior was modeled with sinusoidal motion model with a longer maneuver length of 100 m. The maneuver length then becomes the behavior parameter for this maneuver.

iv) **Aggressive left lane change behavior:** The longitudinal motion was modeled with constant acceleration, whereas lateral motion maintained a sinusoidal lane change with a shorter maneuver length of 70 m.

For each driver type, three thresholds were considered:

i) **Headway distance threshold τ_a:** This represents the minimum distance gap a driver prefers to keep from the vehicle in front. If this distance falls below the threshold the driver is expected to either slow down to increase the distance or change lanes to arrive at a comfortable driving conditions. For an aggressive driver, this threshold is assumed significantly smaller than the one for passive driver.

ii) **Adjacent lane distance threshold τ_b:** This represents the minimum distance a driver considers for gap acceptance. If the distance of the succeeding vehicle (rear vehicle in the adjacent lane) is less than this threshold, the driver rejects the gap to perform a lane change maneuver. For an aggressive driver, this threshold would be smaller than a passive driver.

iii) **Adjacent lane velocity threshold τ_c:** This represents the difference in velocity between the succeeding vehicle in the adjacent lane and the observed vehicle. If this velocity difference is larger than the threshold, the driver is expected to reject the gap for performing a lane change maneuver. For an aggressive driver this threshold would be larger compared to a passive driver. In other words, an aggressive driver is likely to accept the gap even if the vehicle in the adjacent lane has a velocity exceeding its own, possibly cutting into the way of the vehicle in the rear, whereas a passive driver is less likely to accept the gap in such case.

Based on the definitions above, for the aggressive driver, the thresholds were chosen as $\tau_a = 60$ m, $\tau_b = 5$ m and $\tau_c = 3$ m/s. For a passive driver, the thresholds were chosen as $\tau_a = 60$ m, $\tau_b = 15$ m and $\tau_c = -1$ m/s. The gap acceptance policy for each driver is determined based on two factors: current lane satisfaction and adjacent lane dissatisfaction. The current lane satisfaction is calculated as

$$cl_{sat} = c_3 > \tau_a \tag{8.25}$$

whereas, the adjacent lane dissatisfaction is calculated as:

$$al_{diss} = \big(abs(c_2) \leq \tau_b\big) \vee \big((c_2 > \tau_b) \wedge (c_7 - v_x) \geq \tau_c\big) \tag{8.26}$$

Gap rejection and acceptance probabilities for the driver are then calculated as follows based upon a binary vector for the gap functions described in Eqn. (8.8).

$$g_{NAd} = cl_{sat} \vee al_{diss} \tag{8.27}$$

$$g_{Ad} = 1 - g_{NAd} \tag{8.28}$$

Remark. *While the adjacent lane dissatisfaction criterion in our example was based on the distance and velocity relations with the rear vehicles in the target lane, a more sophisticated criteria could be applied in practice. A similar argument can be made about the lane satisfaction criterion, by exploring other factors such as velocity, time headway, etc. An example of a more sophisticated choice for the gap policies are model-based or machine-learning-based techniques in traffic engineering to model drivers.*

Utilizing the above expressions, the resulting transition probability matrix yields

$$\Lambda_{trans} = \begin{bmatrix} \lambda_{11} & \frac{(1-\lambda_{11})g_{NA}}{\eta} & \frac{(1-\lambda_{11})\ d_2\ g_A}{\eta} & \frac{(1-\lambda_{11})\ d_1\ g_A}{\eta} \\ (1-\lambda_{22})g_{NA} & \lambda_{22} & 0 & (1-\lambda_{22})g_A \\ 1-\lambda_{33} & 0 & \lambda_{33} & 0 \\ 1-\lambda_{44} & 0 & 0 & \lambda_{44} \end{bmatrix} \tag{8.29}$$

with

$$\eta = g_{NA} + d_2\ g_A + d1\ g_A \tag{8.30}$$

Here, the diagonal elements of the transition probabilities, λ_{11}–λ_{44}, were determined heuristically as described in the chapter 7. In practice however, these values could be derived from traffic data statistics. An underlying and well justified assumption is that a traffic participant in general has a high probability of staying in its current state yielding constant values, i.e.

$$\lambda_{11} = \lambda_{22} = \lambda_{33} = \lambda_{44} = 0.97 \tag{8.31}$$

Now, for every cycle of the proposed behavior identifier filter, the steps involved are as follows:

1. Extract the context from the prior states of all the traffic participants
2. Find the gap acceptance policy for all the considered driver types for each traffic participant
3. Calculate behavior transition probability matrix for every traffic participant
4. Mix and reinitialize the estimates of the CIMM filter
5. Propagate the estimates using the prediction steps of all the elemental filters
6. When the measurements are available, update the estimates using the correction step
7. Update the probabilistic weights of the behaviors
8. Update the driver type based on the weights of the behaviors executed
9. Calculate combined estimates from each elemental filter
10. Goto Step 1.

To evaluate its performance, the implemented filter was compared with a traditional IMM for the two bifurcations of the above scenario discussed in the next section. The transition probability matrix for the IMM based behavior identifier was chosen as:

$$\Lambda_{trans,(IMM)} = \begin{bmatrix} 0.97 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.97 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.97 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.97 \end{bmatrix} \tag{8.32}$$

8.3.1 Discussion of results

8.3.1.1 Aggressive driver use case

For this use case, Sub and $O1$ start with the conditions described above. In addition, the longitudinal condition for $O1$ is $x = 100$ m. The vehicle $O2$ starts at $x = -106$ m, $y = 3.5$ m with velocities $v_x = 15$ m/s and $v_y = 0$ m/s. As the subject vehicle (Sub) approaches the slower leading vehicle in the right lane, it notices the vehicle $O2$ approaching in the adjacent lane at a higher velocity. The driver hence starts accelerating at simulation time 10 seconds, exhibiting an opportunistic behavior. Eventually, after a gap acceptable to the driver is achieved, it performs an aggressive left lane change 15 seconds into the simulation, cutting-in front of the vehicle $O2$. It should be noted that the transition probability matrix for the IMM stays constant throughout the simulation since it is independent of the context of the vehicle. The proposed Contextual IMM (CIMM) on the other hand, monitors the context of the observed traffic participant and adapts the transition probability matrix as the simulation evolves.

Figure 8.5 demonstrates the results of the inference using CIMM. The CIMM is not only able to identify the behaviors correctly but also correctly classify the driver as aggressive. The detection times for the respective behaviors for both IMM and the CIMM are shown in the table 8.1. Fig. 8.6 and Fig. 8.7 show the trends followed by the probabilistic weights of the elemental filters for the corresponding behavior identification models being evaluated. As depicted in Fig. 8.7, although the IMM correctly identifies the lane change

Table 8.1: Detction times for the IMM and CIMM for the aggressive driver use case

Behavior identifier	Aggressive straight behavior detection time (s)	Aggressive lane change detection time (s)	RMSE x (m)	RMSE y (m)
IMM	1.7	1.5	0.0328	0.0289
Proposed CIMM	1.7	0.6	0.0334	0.028

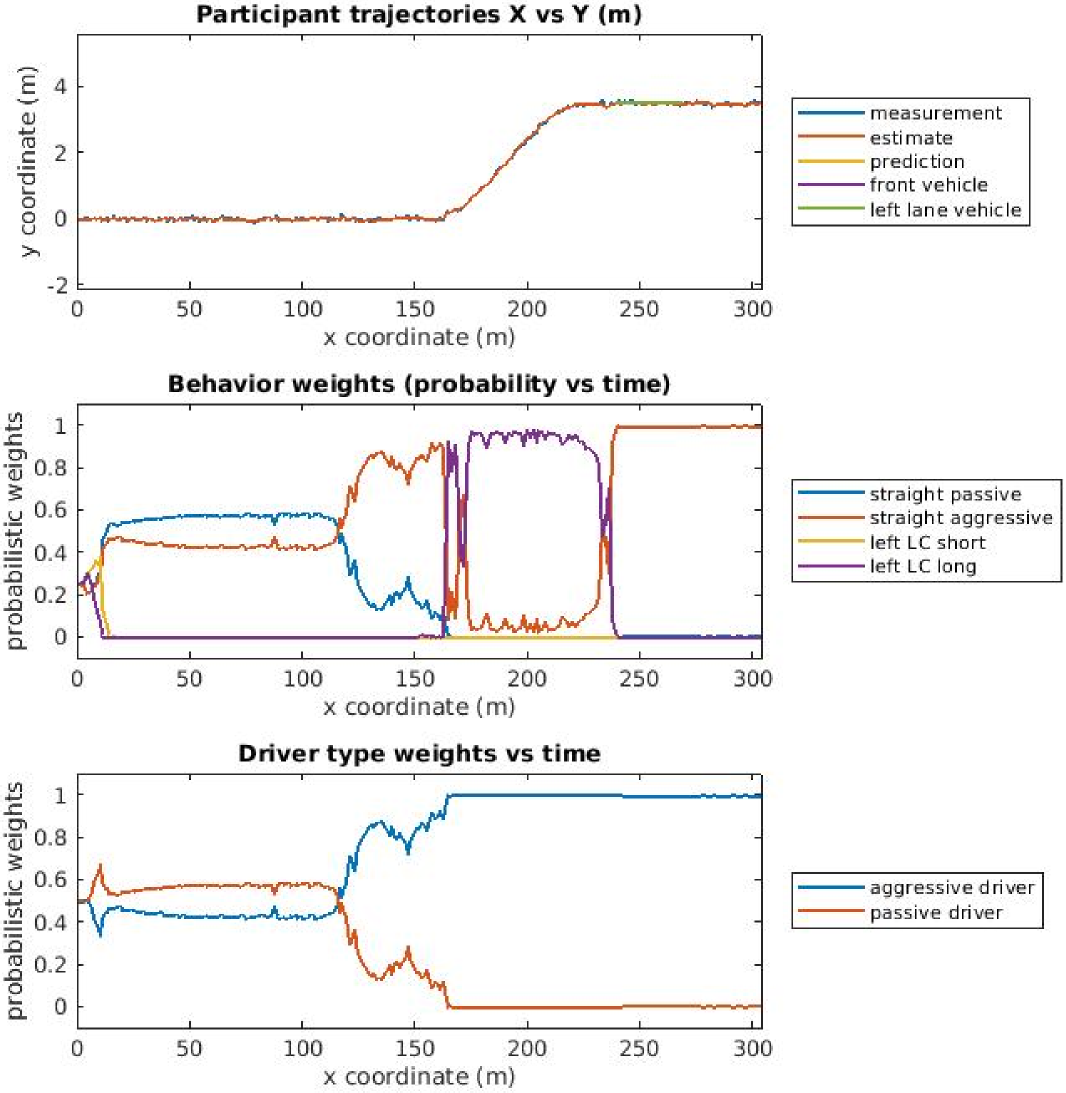


Figure 8.5: CIMM inferencing the *Sub* vehicle for the aggressive driver use case

behavior as an aggressive one, it shows some ambiguity between the aggressive and passive before the weights diverge. The CIMM on the other hand, profits from context-awareness and detects the aggressive left lane change behavior roughly one second earlier. It expects an aggressive behavior as the driver is classified due to the vehicle acceleration in the preparatory phase. The detection times as well as the RMSE for this use case are given in Table 8.1

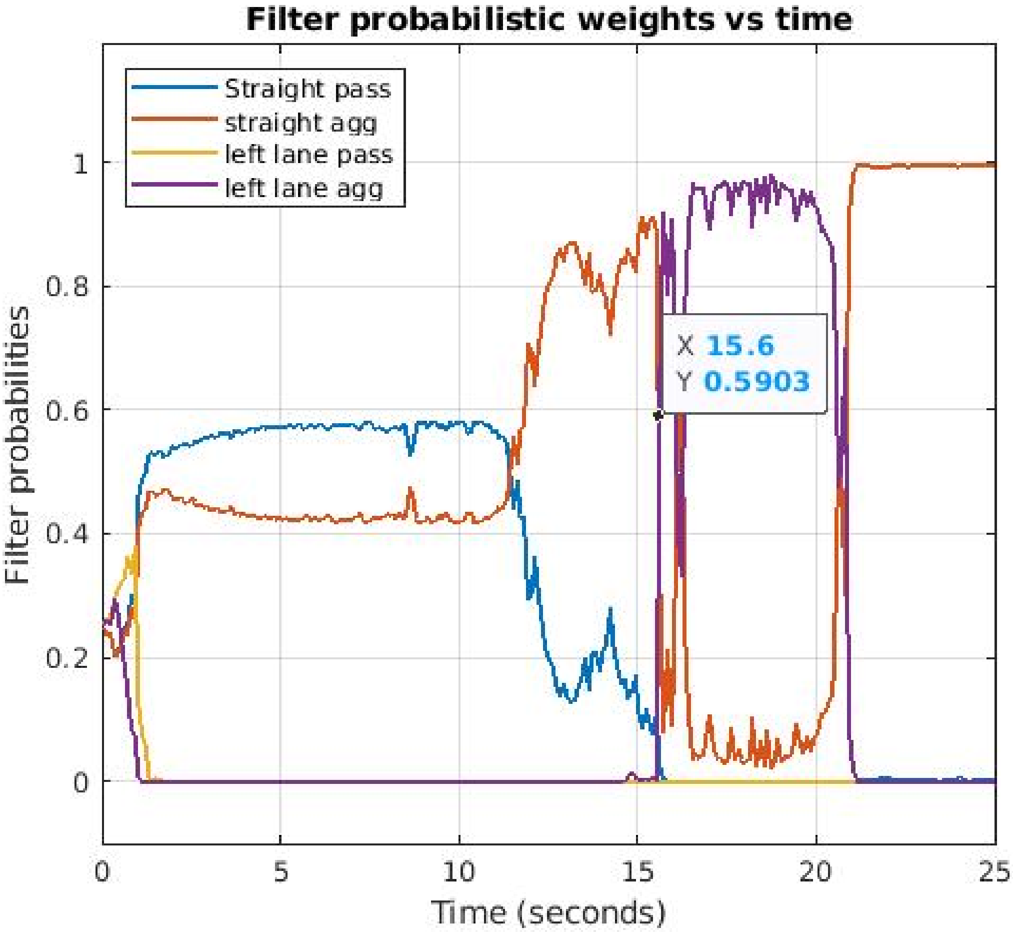


Figure 8.6: Behavior weights for the CIMM

for both the filters.

8.3.1.2 Passive driver use case

Table 8.2: Detction times for the IMM and CIMM for the passive driver use case

Behavior identifier	Passive lane change detection time (s)	RMSE x (m)	RMSE y (m)
IMM	3.2	0.0519	0.0335
Proposed CIMM	2.7	0.0519	0.0329

For this use case, *Sub* vehicle stats with the same initial conditions as before. The

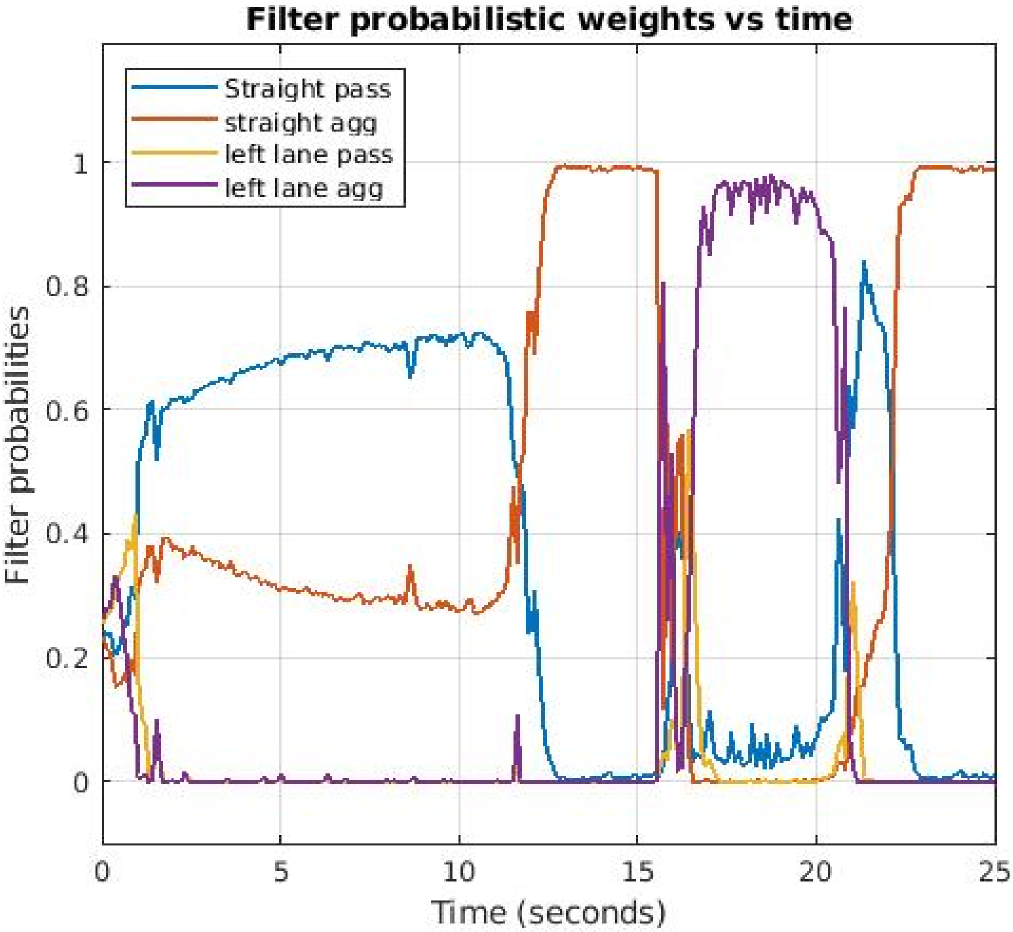


Figure 8.7: Behavior weights for the IMM inferencing the Sub vehicle for the aggressive driver use case

vehicle $O2$ starts at $x = -25$ m, $y = 3.5$ m and the vehicle $O1$ starts at $x = 80$ m, $y = 0$ m. These conditions were chosen such that all the contextual changes in the surroundings of the vehicle occurs during the simulation interval. The filter itself has no dependency on the starting conditions of other vehicles. As the subject vehicle (Sub) approaches the slower vehicle in the front of the right lane, it notices the vehicle $O2$ approaching from the adjacent lane at a higher velocity. The driver maintains the current speed, lets the faster vehicle in the adjacent lane pass and then performs a passive left lane change behavior with longer maneuver length after an acceptable gap. The results for the inference using both

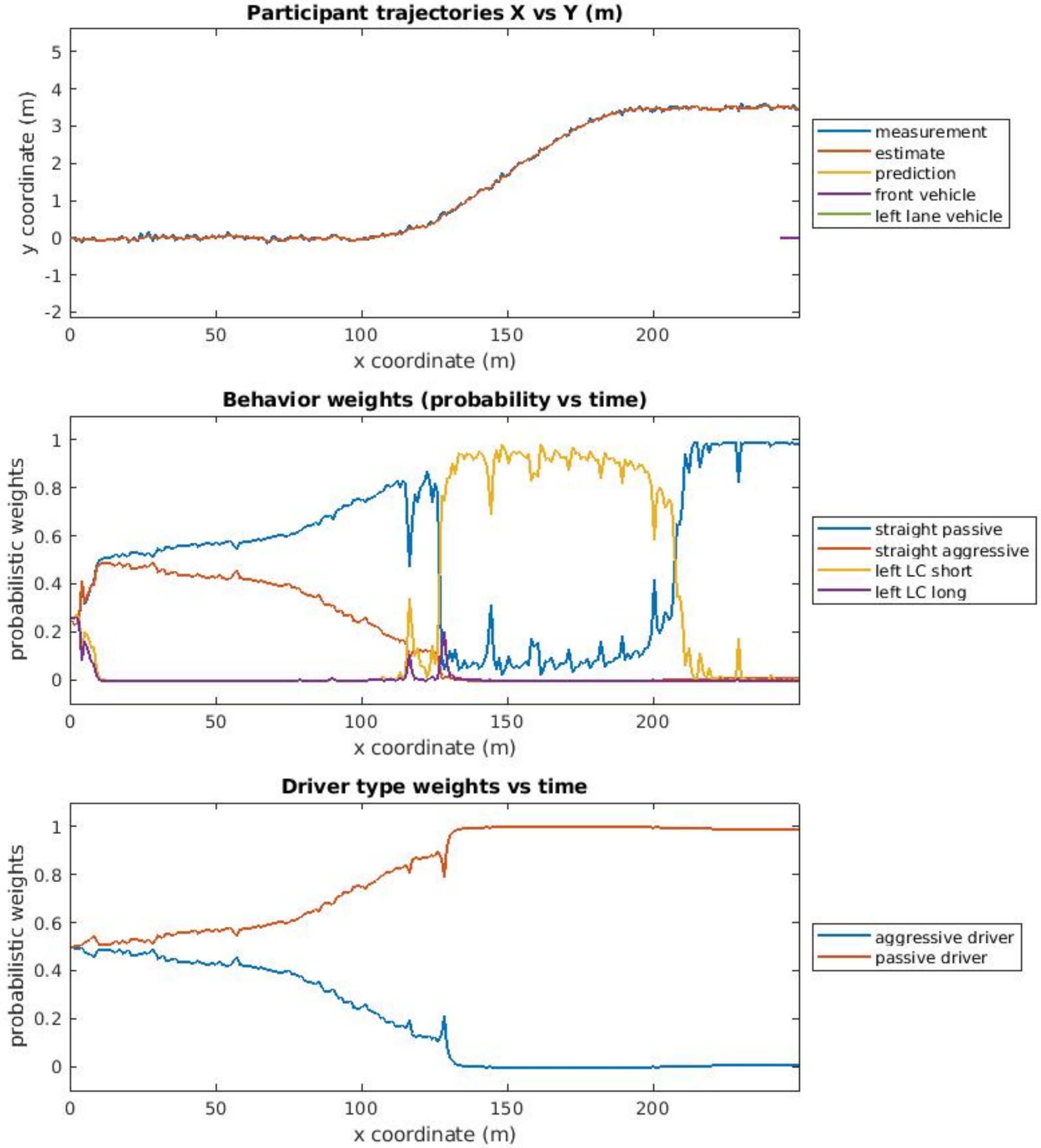


Figure 8.8: CIMM inferencing the *Sub* vehicle for the passive driver use case

the CIMM and IMM filters are shown in the table 8.2. From Fig. 8.8, it can be seen that the CIMM correctly identifies the behaviors and also classifies the driver as passive. Again CIMM detection times are about half a second shorter than the regular IMM. Furthermore,

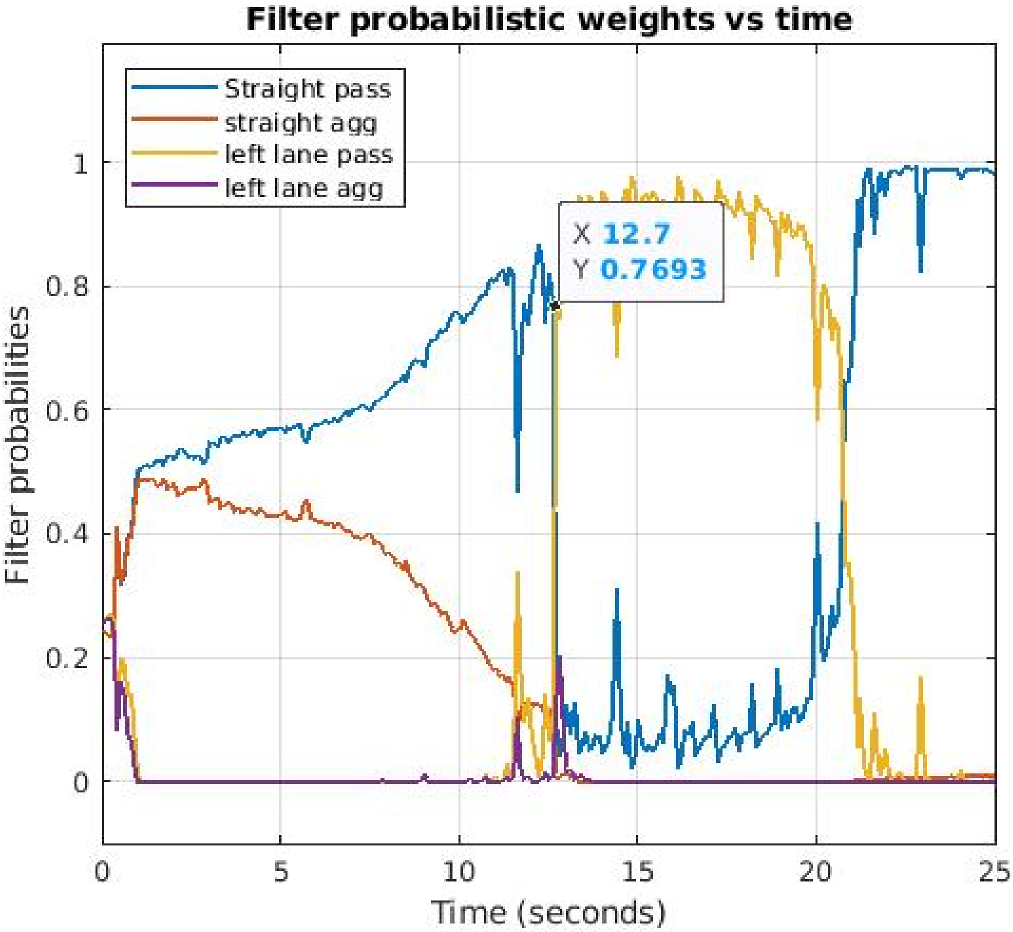


Figure 8.9: CIMM inferencing the *Sub* vehicle for the passive driver use case

the RMSE values exhibit that the CIMM is not only faster, but also at least matches the tracking performance of the regular IMM. It should be noted that given this performance, the proposed CIMM not only improves the behavior identification by monitoring the context of the vehicle, but also provides additional information about the driver behavior. Gathering both pieces of information is a non-trivial task for long term trajectory prediction.

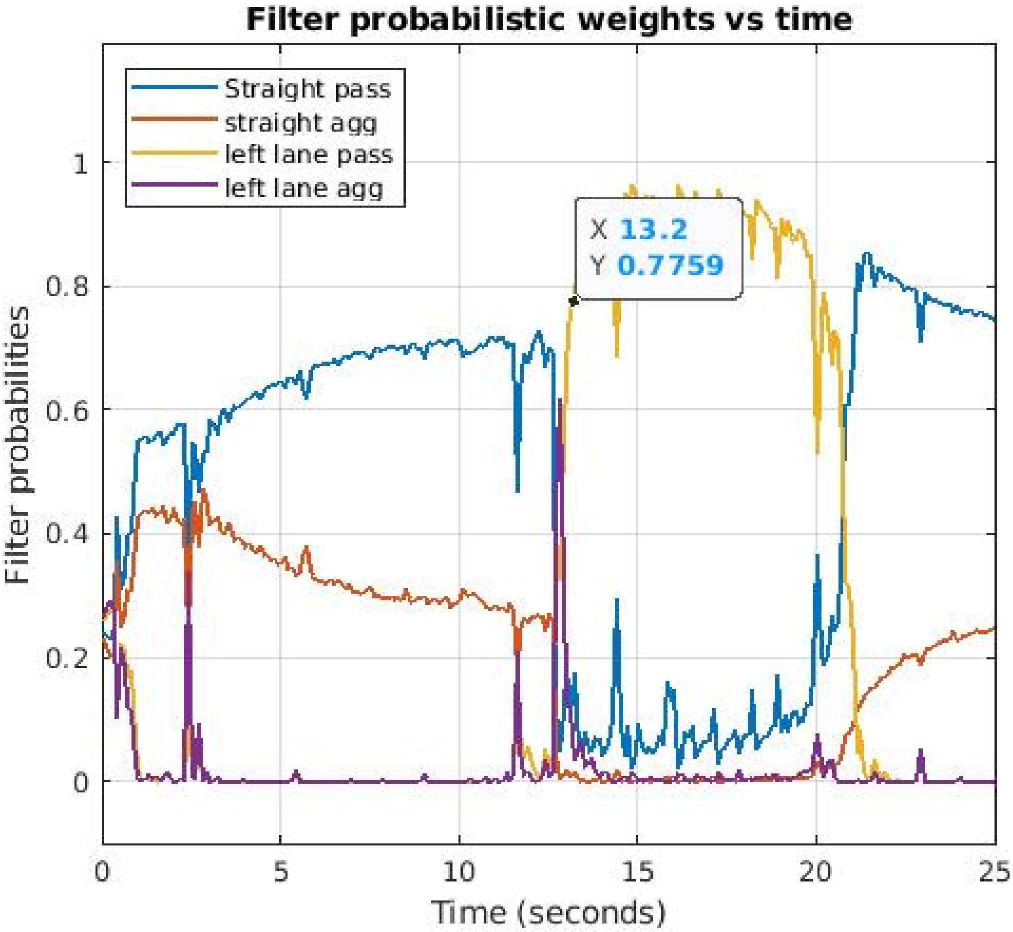


Figure 8.10: IMM inferencing the *Sub* vehicle for the passive driver use case

8.4 Conclusion

In this chapter, we reformulated our previous approach for modeling traffic participants by realizing the proposed framework utilizing an IMM and demonstrated how the contextual information aids in improving the performance of behavior identification. The developed framework was evaluated for two simulated scenarios. To demonstrate its effectiveness, the resultant contextual IMM (CIMM) was compared with the traditional IMM based approach that was detailed in the chapter 7. The proposed CIMM has only exhibited vast improvements in the detection times for behaviors, but its tracking performance was at par with the IMM. Furthermore, it achieved classification of the driving style based on the sequence of behaviors that were executed.

Chapter 9

Conclusions and Future Work

This study has developed and analyzed a new method for predicting the behaviors of the traffic participants operating around autonomous vehicles. Since autonomous vehicles will be sharing the space with human operated vehicles for a foreseeable future, this work contributes towards improving prediction methods for safe operation. The research rests on the hypothesis that a characteristic behavior of a traffic participant spans across multiple maneuvers performed. Furthermore, human drivers perform any maneuver in four stages: intent determination, maneuver preparation, gap acceptance and maneuver execution. Identifying the driver behavior in any of these stages aids in identifying the driver type (i.e. aggressive, passive, cooperative, etc.) and supports the characterization of subsequent maneuvers the driver is likely to perform.

In chapter 4, a probabilistic framework was proposed that takes the contextual information of a traffic participant into consideration, and observes them for the different stages of maneuver execution described above. Based on the behavior, this framework characterizes the driver type so that its influence can be factored in during subsequent predictions. The traffic participant was modeled as a hybrid system using a Dynamic Hierarchical Bayesian Network, and an exemplary approach for implementing the framework was presented.

One of the key challenges in developing the framework is to be able to reliably identify the behaviors that the traffic participants are executing. Since all the other parameters in the model rely on the estimated base and modal states of the system (i.e continuous states and behaviors) in depth attention was given for exploring the methods involved. Chapter 5 - 7 focus on exploring two different approaches in Multiple Model Adaptive Estimation: an Autonomous Multiple Model (AMM) approach and an Interacting Multiple Model (IMM) approach. Chapter 5 analyzes the AMM against different measurement and process noise quantities. Simulations were conducted by synthetically adding process and measurement noise to a geometric model as well as a single track vehicle dynamics model. These synthetic measurements were then employed to evaluate the classification performance of an AMM that modeled three maneuvers: straight motion, left lane change and right lane change. Existing literature provides very limited analysis of the effect of the process noise on the MMAE classification, but rather focuses on tracking. While AMM performance was found satisfactory against the measurement noise, the impact of process noise on classification was significant. Furthermore, the filter has exhibited higher sensitivity to initial covariances of the states.

Chapter 6 extends the inference of the AMM approach to prolonged observations across multiple maneuvers. It was observed that AMMs lack an inherent reset mechanism to enable transitioning to another model after classification is complete. To address this challenge, an approach based on monitoring the likelihood of the default elemental filter was presented. The premise of this approach stems from the fact that traffic participants spend majority of their time executing a default behavior (typically straight or lane following). By monitoring the divergence of the likelihood of a participant from this behavior, the AMM can be reinitialized. This implemented method was evaluated with a simulated scenario and the results were discussed. Yet, the approach involved some tuning factors that had to be determined heuristically.

Chapter 7 explored IMM filters for behavior identification with subsequent analysis against different process noise qualities. Their performance was compared with the AMMs for similar experimental settings. The metrics used for comparison were detection times for a maneuver and root mean squared error (RMSE) for both x and y positions. While AMMs peformed better for identification of the current maneuver executed by an observed participant, IMMs were able to transition between multiple modes effortlessly for prolonged observations. Furthermore, their performance for prolonged observations was found to be superior than AMMs. In the study, an intuition of selecting the involved Markov transition probabilities for the IMM was also presented. The developed IMM was then validated utilizing a trajectory extracted from a naturalistic dataset called highD-dataset (obtained from German highways). From the results it was concluded that IMMs appear to be a superior choice for realizing the proposed framework.

Chapter 8 focused on reformulating the introduced framework for IMMs. The framework rests on the premise that adapting the transition probabilities of the IMM can improve the performance for prediction if the context of the traffic participants can be observed. Hence, a Contextual IMM (CIMM) results. Additionally, the driver style can be taken into account aiding the characterization of subsequent behaviors of the traffic particpants. The framework with CIMM was then evaluated against the IMM evaluated in Chapter 7 with two use cases involving aggressive and passive driving. The CIMM not only outperformed the IMM in behavior classification, but also classified the driver type of the traffic participant correctly, while matching its tracking performance with the IMM.

9.1 Answering the Research Questions

1. How to predict the trajectories of traffic participants for time horizons exceeding the human response time (1-2 secs) using only their extrinsic information?

   Both the type of driver as well as their contextual information, together influence the future trajectories of a traffic participant. In this research, an approach was demonstrated that takes into consideration different driving styles as well as the context of a traffic participant in order to predict subsequent behaviors. The developed framework showed an improvement of atleast 0.5-1 second in behavior identification times compared to a traditional IMM based approach lacking this information.

2. How to model interactions between the traffic participants that follow different driving policies?

   In this research, an approach based on Hierarchical Dynamic Bayesian Network was presented that takes into account different driving policies for modeling the interactions between traffic participants. While this approach was demonstrated with two driving styles (aggressive and passive), it can be extended to a multitude of driving styles.

9.2 Recommendations for Further Research

We suggest to extend this research further in the following directions:

(i) In this study, the filter was utilized to infer the behavior of only a single vehicle. This work should be extended to perform inference on multiple vehicles. The challenge arises that the datasets describing and characterizing different driver behaviors in traffic are very rare and limited. Hence, tools such as Siemens PreScan and Metamoto could be utilized in generating simulated scenarios such that the performance of the framework can be analyzed further.

(ii) Since the presented approach involves adapting the transition probability matrix for an IMM, further work is needed to analyze the efficiency and robustness of the filter. This can be done via traditional efficincy and consistency analysis, e.g. utilizing Cramer Rao error lower bound estimation [25]. However, this is expected to be challenging for multiple model filtering. Developments in this area are relatively recent [63].

(iii) Literature on selecting the right motion model for prediction, in contrast to tracking, is relatively limited. Analyzing existing popular motion models against realistic dataset could help identify effective motion models as the foundation of prediction.

Appendices

Appendix A Linearization and Discretization of Systems

Linear system state space representation is:

$$\dot{X} = A \cdot X + B \cdot U \tag{1}$$

$$Y = C \cdot X + D \cdot U \tag{2}$$

Its important to note that the $\dot{X}$ indicates just an increment in the continuous time step. Whereas x_k in a discrete time equation represents the full state at next time step (which includes state at k-1 time step plus increment/decrement).

A.1 Discretization of Linear Systems

1. To discretize this system, sampling time is important, since changes in the system need to be propagated over the time step. If A is constant the system is an LTI (linear time invariant) system. The state matrix A remains the same in every time step, so it can be calculated once at the beginning

2. If the state transition matrix is dependent on time, i.e. $A(t)$, then it is called as Linear Time Varying system. In this case, the linear state matrix needs to be calculated at every time step before propagation.

3. Discretization solution lies in the time domain solution for a first order differential equation since eqn.(1) is nothing but a first order differential equation (See the handouts in Ayalew's notes). Different approximations are as follows.

4. Euler forward:

$$A_d = I + A \cdot T_s \tag{3}$$

$$B_d = B \cdot T_s \tag{4}$$

$$C_d = C \tag{5}$$

$$D_d = D \tag{6}$$

5. Euler backward:

$$A_d = (I - A \cdot T_s)^{-1} \tag{7}$$

$$B_d = (I - A \cdot T_s)^{-1} \cdot B \cdot T_s \tag{8}$$

$$C_d = C \cdot (I - A \cdot Ts)^{-1} \tag{9}$$

$$D_d = D + C_d \cdot B \cdot T_s = D + C \cdot B_d \tag{10}$$

In MATLAB:

$$B_d = (I - A \cdot T_s) \backslash T_s = A_d \cdot B \cdot T_s \tag{11}$$

$$C_d = C/(I - A \cdot T_s) \tag{12}$$

6. Numerical Integration:

$$A_d = expm(A \cdot T_s) \tag{13}$$

$$B_d = (A_d - I) \cdot A^{-1} \cdot B; \tag{14}$$

$$C_d = C \tag{15}$$

$$D_d = D \tag{16}$$

$$\tag{17}$$

In MATLAB,

$$B_d = (A_d - I) \cdot A \backslash B) \tag{18}$$

Careful with this one. The MATLAB function is *expm* and NOT *exp* (which is a different function in MATLAB). *expm*() is equivalent to the Taylor's series expansion below, which is similar to Euler forward, but a little more accurate

$$e^{(A \cdot T_s)} = I + A \cdot T_s + A^2 \cdot \frac{Ts^2}{2!} + A^3 \cdot \frac{Ts^3}{3!} + \ldots$$

7. Trapezoidal, also called bilinear transformation:

$$Trap\_term = (I - A \cdot \frac{T_s}{2}) \tag{19}$$

$$A_d = Trap\_term^{-1} \cdot (I + A \cdot \frac{T_s}{2}) \tag{20}$$

$$B_d = Trap\_term^{-1} \cdot B \cdot T_s \tag{21}$$

$$C_d = C \cdot Trap\_term^{-1} \tag{22}$$

$$D_d = C_d \cdot B \cdot T_s \tag{23}$$

A.2 Discretization of Non-linear Systems:

Non linear systems are represeted as:

$$\dot{X} = f(x, u, t) + w \tag{24}$$

$$\dot{Y} = h(x, u, t) + r \tag{25}$$

They cannot be put into the linear form A, B, C, D directly due to the non-linearity and hence discretization becomes tricky. An approximation is as follows:

1. For every time step, calculate the $f(\cdot)$ and $h(\cdot)$ functions as they are dependent on input, current state as well as time.

2. Linearize the non-linear system around the operating points (i.e. x_k) in continuous time using Taylor series expansion and then discretize. Taylor series expansion of the next time step is:

   $$f(x_{k+1}) = f(x_k) + F(x_k) \cdot T_s + f''(x_k) \cdot T_s^2/2 + \dots$$

   Where F is the jacobian, $f''(x)$ is the Hessian matrix

3. Hence, it requires finding the Jacobian (gradient) matrix for the system. Find this in continuous time function. Then use the current values of X to calculate F(X)

4. The discretized F_d can now be calculated from the linearized system F (jacobian matrix) using any of the above mentioned discretization techniques. For euler fwd or numerical integration, using *expm(*$F_d \cdot T_s$*)*

5. The same for H

6. Now the system can be propagated either using the linearized & discretized function or using the non-linear discretized function. For filters, the updates need the jacobian matrices.

Appendix B List of Publications

1. **Gill, J. S.**, Pisu, P., & Schmid, M. J. (2019). A Probabilistic Framework for Trajectory Prediction in Traffic utilizing Driver Characterization. 2nd IEE Connected and Automated Vehicle Symposium 2019

2. **Gill, J. S.**, Pisu, P., Krovi, V. N., & Schmid, M. J. (2019). Behavior Identification and Prediction for a Probabilistic Risk Framework. Dynamic System and Control Conference 2019

3. **Gill, J. S.**, Bhavsar, P., Chowdhury, M., Johnson, J., Taiber, J., & Fries, R. (2014). Infrastructure Cost Issues Related to Inductively Coupled Power Transfer for Electric Vehicles. Procedia Computer Science, 32, 545–552.

4. **Singh Gill, J.,** Tomaszewski, M., Jia, Y., Pisu, P., & Krovi, V. (2019). Evaluation of Navigation in Mobile Robots for Long-Term Autonomy in Automotive Manufacturing Environments. In WCX SAE World Congress Experience. SAE International.

5. Sauras-Perez, P., Gil, A., **Singh Gill, J**., Pisu, P., & Taiber, J. (2017). VoGe: A Voice and Gesture System for Interacting with Autonomous Cars. In WCXTM 17: SAE World Congress Experience. SAE International

6. Baskaran, S., Niaki, F. A., Tomaszewski, M., **Gill, J. S.**, Chen, Y., Jia, Y., … Krovi, V. (2019). Digital Human and Robot Simulation in Automotive Assembly using Siemens Process Simulate: A Feasibility Study. Procedia Manufacturing, 34, 986–994.

Bibliography

[1] Autonomous Vehicle Disengagement Reports 2018. `https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report_2018`.

[2] Clearpath Robotics: Mobile Robots for Research & Development.

[3] DARPA Grand Challenge 2004. https://archive.darpa.mil/grandchallenge04/media_gallery.htm.

[4] DARPA Grand Challenge 2005.

[5] DARPA Urban Challenge 2007. https://archive.darpa.mil/grandchallenge/gallery.html.

[6] OTTO Self Driving Vehicles for Automated Material Transport.

[7] The DARPA Grand Challenge: Ten Years Later. `https://www.darpa.mil/news-events/2014-03-13`.

[8] This Is How Many Autonomous Cars Will Be on the Road in 2025 - TheStreet. https://www.thestreet.com/technology/this-many-autonomous-cars-will-be-on-the-road-in-2025-14564388.

[9] Vecna Robotics.

[10] G Agamennoni, J I Nieto, and E M Nebot. Estimation of Multivehicle Dynamics by Considering Contextual Information. *IEEE Transactions on Robotics*, 28(4):855–870, aug 2012.

[11] A. Alin, M. V. Butz, and J. Fritsch. Incorporating environmental knowledge into bayesian filtering using attractor functions. In *2012 IEEE Intelligent Vehicles Symposium*, pages 476–481, June 2012.

[12] M. Althoff, O. Stursberg, and M. Buss. Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):299–310, June 2009.

[13] Angelos Amditis, Aris Polychronopoulos, Ioannis Karaseitanidis, George Katsoulis, and Evangelos Bekiaris. Multiple sensor collision avoidance system for automotive applications using an imm approach for obstacle tracking. In *Procedings of the Fifth*

International Conference on Information Fusion; Volume II, volume 2, pages 812–817, 2002.

[14] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How. Behavior classification algorithms at intersections and validation using naturalistic data. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 601–606, June 2011.

[15] Mohammad Bahram. *Interactive Maneuver Prediction and Planning for Highly Automated Driving Functions*. PhD thesis, Technische Universität München, 2017.

[16] Thomas Batz, Kym Watson, and Jurgen Beyerer. Recognition of dangerous situations within a cooperative group of vehicles. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 907–912. IEEE, 2009.

[17] H. Berndt, J. Emmert, and K. Dietmayer. Continuous driver intention recognition with hidden markov models. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 1189–1194, Oct 2008.

[18] Karl Berntorp, Björn Olofsson, Kristoffer Lundahl, and Lars Nielsen. Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres. *Vehicle System Dynamics*, 52(10):1304–1332, 2014.

[19] A. Berthelot, A. Tamke, T. Dang, and G. Breuel. Handling uncertainties in criticality assessment. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 571–576, June 2011.

[20] S. Brechtel, T. Gindele, and R. Dillmann. Probabilistic mdp-behavior planning for cars. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1537–1542, Oct 2011.

[21] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[22] Ashwin Carvalho, Yiqi Gao, Stéphanie Lefevre, and Francesco Borrelli. Stochastic predictive control of autonomous vehicles in uncertain environments. In *12th International Symposium on Advanced Vehicle Control*, pages 712–719, 2014.

[23] Howie M Choset, Seth Hutchinson, Kevin M Lynch, George Kantor, Wolfram Burgard, Lydia E Kavraki, and Sebastian Thrun. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[24] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2004.

[25] John L Crassidis and John L Junkins. *Optimal Estimation of Dynamic Systems, Second Edition (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science)*. Chapman & Hall/CRC, 2nd edition, 2011.

[26] Florian Damerow. *Situation-based Risk Evaluation and Behavior Planning*. PhD thesis, Technische Universität, Darmstadt, 2018.

[27] Gabriel R de Campos, Adam H Runarsson, Fredrik Granum, Paolo Falcone, and Klas Alenljung. Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 649–654. IEEE, 2014.

[28] Y. Dou, F. Yan, and D. Feng. Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 901–906, July 2016.

[29] Zvi Drezner. Computation of the bivariate normal integral. *Mathematics of Computation*, pages 277–279, 1978.

[30] J. Eggert, F. Damerow, and S. Klingelschmitt. The foresighted driver model. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 322–329, June 2015.

[31] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal. Moving obstacle avoidance for large, high-speed autonomous ground vehicles. In *2017 American Control Conference (ACC)*, pages 5568–5573, May 2017.

[32] Luke Fletcher, Seth Teller, Edwin Olson, David Moore, Yoshiaki Kuwata, Jonathan How, John Leonard, Isaac Miller, Mark Campbell, Dan Huttenlocher, et al. The mit–cornell collision and why it happened. *Journal of Field Robotics*, 25(10):775–807, 2008.

[33] M. Galvani, F. Biral, B. M. Nguyen, and H. Fujimoto. Four wheel optimal autonomous steering for improving safety in emergency collision avoidance manoeuvres. In *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*, pages 362–367, March 2014.

[34] Jasprit Singh Gill, Pierluigi Pisu, Venkat N. Krovi, and Matthias J. Schmid. Behavior Identification and Prediction for a Probabilistic Risk Framework. In *ASME 2019 Dynamic Systems and Control Conference, DSCC 2019*. Under review, http://arxiv.org/abs/1905.08332, 2019.

[35] Jasprit Singh Gill, Pierluigi Pisu, and Matthias J. Schmid. A Framework for Characterizing the Behaviors of Traffic Participants. In *2019 2nd IEEE Connected and Automated Vehicle Symposium*. Under review.

[36] Tobias Gindele, Sebastian Brechtel, and Rüdiger Dillmann. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1625–1631. IEEE, 2010.

[37] Peter G Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111, 1981.

[38] Peter G Gipps. A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological*, 20(5):403–414, 1986.

[39] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[40] J Huang and H Tan. Error Analysis and Performance Evaluation of a Future-Trajectory-Based Cooperative Collision Warning System. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):175–180, mar 2009.

[41] C G Keller, T Dang, H Fritz, A Joos, C Rabe, and D M Gavrila. Active Pedestrian Safety by Automatic Braking and Evasive Steering. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1292–1304, dec 2011.

[42] Arne Kesting, Martin Treiber, and Dirk Helbing. General lane-changing model mobil for car-following models. *Transportation Research Record*, 1999(1):86–94, 2007.

[43] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[44] A. Lambert, D. Gruyer, G. S. Pierre, and A. N. Ndjeng. Collision probability assessment for speed control. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 1043–1048, Oct 2008.

[45] Steven M LaValle. *Planning Algorithms*. Cambridge university press, 2006.

[46] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1, 2014.

[47] Stéphanie Lefèvre, Dizan Vasquez, Christian Laugier, and Javier Ibañez-Guzmán. Intention-aware risk estimation: Field results. In *ARSO*, pages 1–8, 2015.

[48] X Rong Li and V P Jilkov. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, oct 2003.

[49] X Rong Li and V P Jilkov. Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, oct 2005.

[50] Yaxin Li, Weiwen Deng, Bohua Sun, Jinsong Wang, and Jian Zhao. A maneuver-based threat assessment strategy for collision avoidance. Technical report, SAE Technical Paper, 2018.

[51] Peng Liu, Arda Kurt, Keith Redmill, and Umit Ozguner. Classification of highway lane change behavior to detect dangerous cut-in maneuvers. In *The Transportation Research Board (TRB) 95th Annual Meeting*, volume 2, 2015.

[52] Jerome M Lutin. Not If, but When: Autonomous Driving and the Future of Transit. *Journal of Public Transportation*, 21(1):10, 2018.

[53] P. Lytrivis, G. Thomaidis, M. Tsogas, and A. Amditis. An advanced cooperative path prediction algorithm for safety applications in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):669–679, Sep. 2011.

[54] Panagiotis Lytrivis, George Thomaidis, and Angelos Amditis. Sensor data fusion in automotive applications. In *Sensor and Data Fusion*. IntechOpen, 2009.

[55] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *International Conference on Robotics and Automation (ICRA), workshop on open source software*, 2009.

[56] Mizanur Rahman, Mashrur Chowdhury, Yuanchang Xie, and Yiming He. Review of microscopic lane-changing models and future research opportunities. *IEEE transactions on intelligent transportation systems*, 14(4):1942–1956, 2013.

[57] T. Rehder, W. Muenst, L. Louis, and D. Schramm. Learning lane change intentions through lane contentedness estimation from demonstrated driving. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 893–898, Nov 2016.

[58] SAE On road Automated Vehicles Standards Committee et al. Sae j3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE J3016, Sep*, 2016.

[59] M. Schreier, V. Willert, and J. Adamy. An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2751–2766, Oct 2016.

[60] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka. Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1467–1474, Nov 2018.

[61] Nayabrasul Shaik, Thomas Liebig, Christopher Kirsch, and Heinrich Müller. Dynamic map update of non-static facility logistics environment with a multi-robot system. In Gabriele Kern-Isberner, Johannes Fürnkranz, and Matthias Thimm, editors, *KI 2017: Advances in Artificial Intelligence*, pages 249–261, Cham, 2017. Springer International Publishing.

[62] Jasprit Singh Gill, Mark Tomaszewski, Yunyi Jia, Pierluigi Pisu, and Venkat N. Krovi. Evaluation of Navigation in Mobile Robots for Long-Term Autonomy in Automotive Manufacturing Environments. In *WCX SAE World Congress Experience*. SAE International, apr 2019.

[63] Lennart Svensson. Evaluating the bayesian cramér-rao bound for multiple model filtering. In *2009 12th International Conference on Information Fusion*, pages 1775–1782. IEEE, 2009.

[64] H Tan and J Huang. DGPS-Based Vehicle-to-Vehicle Cooperative Collision Warning: Engineering Feasibility Viewpoints. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):415–428, dec 2006.

[65] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[66] Tomer Toledo, Haris N Koutsopoulos, and Moshe Ben-Akiva. Integrated driving behavior modeling. *Transportation Research Part C: Emerging Technologies*, 15(2):96–112, 2007.

[67] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, 62:1805–1824, Aug 2000.

[68] S. Wagner, K. Groh, T. Kuhbeck, M. Dorfel, and A. Knoll. Using time-to-react based on naturalistic traffic object behavior for scenario-based risk assessment of automated driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1521–1528, June 2018.

[69] M Werling and D Liccardo. Automatic collision avoidance using model-predictive online optimization. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 6309–6314, dec 2012.

[70] Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. Fetch and freight: Standard platforms for service robot applications. In *Workshop on Autonomous Mobile Service Robots*, 2016.

[71] C. Wissing, T. Nattermann, K. Glander, and T. Bertram. Interaction-aware long-term driving situation prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 137–143, Nov 2018.

[72] C. Wissing, T. Nattermann, K. Glander, and T. Bertram. Trajectory prediction for safety critical maneuvers in automated highway driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 131–136, Nov 2018.

[73] Guotao Xie, Hongbo Gao, Lijun Qian, Bin Huang, Keqiang Li, and Jianqiang Wang. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *IEEE Transactions on Industrial Electronics*, 65(7):5999–6008, 2018.

[74] DoHyun Daniel Yoon and Beshah Ayalew. Social force control for human-like autonomous driving. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V003T01A003–V003T01A003. American Society of Mechanical Engineers, 2018.

[75] B Zhou, W Schwarting, D Rus, and J Alonso-Mora. Joint Multi-Policy Behavior Estimation and Receding-Horizon Trajectory Planning for Automated Urban Driving. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2388–2394, may 2018.