DEGREE PROJECT IN INDUSTRIAL ENGINEERING AND MANAGEMENT,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Trajectory generation for safe overtaking maneuver in autonomous vehicles

Evaluated in lane merging scenario utilizing a trajectory planner

MOAAJ BHANA

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Master of Science Thesis 2018 : TRITA ITM-EX 2018:681

Trajectory generation for safe overtaking maneuver in autonomous vehicles

Evaluated in lane merging scenario utilizing a trajectory planner

Moaaj Bhana

Approved 22-10-18	Examiner Martin Törngren	Supervisor Lars Svensson
	Commissioner Alten	Contact person Detlef Scholle

Sammanfattning

Mer än 1,2 miljoner människor dör varje år på grund av trafikskador och för att minska antal olyckor som beror på mänskliga fel är autonoma fordon föremål för intensiv forskning [1]. För att förbättra körupplevelsen har bilföretag utvecklat Advanced Driver Assistance Systems (ADAS) som Adaptive Cruise Control (ACC) och Lane Keeping Aid (LKA) som syftar till att göra körningen säkrare samt mer bekväm. En särskilt intressant manövrering är filbyten. Denna manövrering är en av de mest riskfyllda manövrer som en förare måste utföra på en motorväg och kan uppfattas som utmanande eftersom det innebär förändringar både i sidled samt i fordonets färdriktning i förhållande till andra fordon.

Denna masteruppsats syftar till att utvärdera hur olika prognosmodeller för banplanerare kommer att påverka kollisionsrisken, komfort och resultera i en ökande grad av framgångsrika omkörningar. En ideal planeringsalgoritm är tillförlitlig och gör smarta beslut för att utföra en säker manövrering och skapar ständigt en diskret banprofil i förhållande till parametrarna för omkringliggande fordon. Framtida rörelser förutses med hjälp av prediktionsmodeller som kopplas till kontrollingångar, fordonets egenskaper och externa förhållanden för utvecklingen av fordonets tillstånd. Fordonet ska kunna undvika kollisioner vid tillfällen där en omkörning görs innan två filer övergår till en och därför beaktas endast rörelse i fordonets färdriktning för utvärderingen. Prediktionsmodellen som valts för denna avhandling är konstant acceleration (CA) och konstant hastighet (CV). Detta projekt ingår i ett stort EU-projekt kallad SafeCOP (Safe Cooperating Cyberphysical Systems) med Wireless Communication som syftar till att utveckla en komplett prototyp av ett intelligent transportsystem.

En stor mängd bana generationstekniker har undersökts och fjärdegradspolynomet är valt för bangenerering eftersom den har många fördelar med att ha en låg beräkningskostnad. Kontinuerlig sammankoppling av kurvor är möjlig och speciellt användbar vid utvärdering av komfort. Det är viktigt att banplaneraren kan avbryta banor som skulle vara dynamiskt omöjliga och resultera i en ökad risk för kollision med omkringliggande fordon. De två valda förutsägelsesmodellerna utvärderades för tre olika scenarier som de testas på och deras resultat jämförs. För de scenarierna som behandlas i denna avhandling gav prediktionsmodellen Constant Acceleration (CA) bättre resultat jämfört med prediktionsmodellen Constant Velocity (CV) och hade en lägre risk för kollision vilket ökar antalet framgångsrika omkörningar. Samtidigt är de dynamiska begränsningarna anpassade för att säkerställa att banan som genereras ligger inom passagerarnas komfortzon.

Abstract

More than 1.2 million people die each year due to road traffic injuries [1]. In order to reduce traffic accidents and human errors, autonomous vehicles is been the subject of intense research. To improve the driving experience, automotive companies have developed Advanced Driver Assistance Systems (ADAS) such as Adaptive Cruise Control (ACC) and Lane Keeping Aid (LKA) which aim to make driving safer and more comfortable. One particularly interesting maneuver is the lane change. Lane change maneuver is one of the riskiest maneuvers that a driver has to perform on a highway, and can be perceived as challenging since it involves changes in both the longitudinal and lateral velocities, direction and as well as movement in the presence of other moving vehicles.

This thesis seeks to evaluate how different prediction model of the trajectory planner will affect collision risk, comfort and result in an increasing rate of successful overtakes. An trajectory-planning algorithm will be reliable in making smarter decisions for performing a safe overtaking maneuver's and constantly generate discrete trajectory profile with respect to the parameters of the vehicle in front. Future motion is predicted using prediction models linking control inputs, vehicle properties and external conditions to the evolution of the state of the vehicle. The vehicle should be able to avoid collisions at the point of convergence where two lane road merges into a single lane road and therefore, motion only in the longitudinal direction is considered for the evaluation. The prediction model chosen for this thesis is constant acceleration (CA) and constant velocity (CV). The project is part of a large EU-project called SafeCOP (Safe Cooperating CyberPhysical Systems) using Wireless Communication which aims at developing a complete prototype of an intelligent transport system.

A great amount of trajectory generation techniques have been surveyed and quartic polynomial is selected for trajectory generation as it has many benefits of having a low computational cost and the continuous concatenation of curves is possible. It is important in the trajectory planner to cancel out trajectories which would dynamically not be feasible and result in an increase risk of collision with the surrounding vehicle. The two chosen prediction models were evaluated for three different scenarios on which they are tested and their results is compared. For the different scenarios addressed in this thesis Constant Acceleration (CA) prediction model gave better result when compared to Constant Velocity (CV) prediction model and had an lower risk of collision which increases the number of successful overtakes. While doing so the jerk dynamic constraints were always considered to ensure that the trajectory generated are within the comfort zone of the passenger.

Acknowledgements

This master thesis is one part of a larger project with six other master thesis students at Alten Sweden. I first and foremost want to thank each and every one of this group: Erik Sjölund, Louise Augustsson, Marcus Olsson, Rebecca Wikström and Robin Moreno Rinding. Many thanks to my supervisor at KTH Lars Svensson who was always there when needed and for being actively engaged in the project. I would also like to thank my parents, sister, Siddhant Tripathi and Arijit Sinha Roy for their constant support. Lastly I want to thank Detlef Scholle and Alten for the opportunity to conduct my master thesis at Alten.

Moaaj Bhana,
Stockholm.

Contents

List of Figures

List of Abbreviations

Abbreviation	Description
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AHS	Automated Highway System
CA	Constant Acceleration
CACC	Cooperative Adaptive Cruise Control
CCA	Constant Curvature and Acceleration
CSAV	Constant Steering Angle and Velocity
CTRV	Constant Turn Rate and Velocity
CTRA	Constant Turn Rate and Acceleration
CV	Constant velocity
EMC2	Embedded Multi-Core systems for Mixed Criticality
EU	European Union
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
HDL	Hardware Description Language
INS	Inertial Navigation System
IVHS	Intilligent Vehicle Highway System
LiDAR	Light Detecting and Ranging
LKA	Lane Keeping Aid
MCS	Mixed Criticality System
MPC	Model Predictive Control
OS	Operating System
RC	Radio-Controlled
RRT	Rapid-Exploring Random Tree
RTOS	Real Time Operating System
SAE	Society of Automotive Engineers
SafeCOP	Safe Cooperating Cyber-Physical System
SoC	System of Chip
TTC	Time To Collision
V2V	Vehicle to Vehicle
VHDL	Very High Speed Integrated Circuit Hardware Descriptive Language

Chapter 1

Introduction

This chapter will introduce the subject of trajectory planning and system architecture. The problem that exists in the field and purpose of the degree project.

1.1 Background

The World Health Organization reports that more than 1.2 million people die each year due to road traffic injuries. This makes road traffic accidents a leading cause of human death globally [1]. To improve the driving experience, automotive companies have developed Advanced Driver Assistance Systems (ADAS) such as Adaptive Cruise Control (ACC) and Lane Keeping Aid (LKA) which aim to make driving safer and more comfortable [3]. To further increase the capability of ADAS and eventually progress to fully automated highway driving, one particularly interesting maneuver is the lane change. This maneuver is one of the riskiest maneuvers that a driver has to perform on a highway, and can be perceived as challenging since it involves changes in both the longitudinal and lateral direction as well as movement in the presence of other moving vehicles [4].

The increased demand for transportation lead to increased interest in automated vehicles in order to reduce human errors. The driver-less vehicles, are vehicles which can sense their environment and move along the predefined path without driver intervention. Sensor data fusion plays an important role in current and future vehicular active safety systems as there are several vehicular applications that fusion of data coming from many different sensors is necessary to increase safety and reliability of the overall system.

Automated driving vehicles must be capable to perform many complex operations like changing lanes on a highway, leaving the road, or overtaking another vehicle on a two-way road. This thesis, will focus on the trajectory planning of this third maneuver. About 6 percent of accidents that occurred in Germany were due to overtaking maneuvers caused because of human error, but they cause approximately 9 percent of death and seriously injured people [5] .This shows how dangerous accidents can be caused by a risky execution of an overtaking maneuver.

The automated vehicle should be able to execute safe overtaking for different scenario such as, a two lane highway converging into a single lane road. The vehicle should be able to make smart decisions if it will be able to perform overtaking in time before reaching the point of convergence to avoiding collisions.

The project is part of a larger project conducted by Alten which aims at developing a complete prototype of an intelligent transport system. The vehicles will be fully autonomous and connected to the infrastructure. The project is part of a large EU-project called SafeCOP (Safe Cooperating Cyber-Physical Systems) using Wireless Communication [6].

The Alten MCS (Alten Mixed Criticality System) is a Mixed Criticality System consisting of both hardware and software components. The general idea is that it should be capable of running two operating systems of different criticality on the same hardware, separated via a hypervisor, where errors from the non-critical OS should not be able to propagate into the safety-critical OS. The Alten MCS has currently been built on two different development boards, the EMC2 Development Platform (EMC2DP) and the Zynq Evaluation and Development board (Zedboard). Both boards are equipped with a Zynq-7000 System on Chip (SoC). The trajectory planner will be implemented on the real time operating system of the demonstrator at Alten which is an 1/8 scale RC vehicle to demonstrate the reduction of collision risk in autonomous vehicle.

1.2 Problem statement

An ideal trajectory-planning algorithm will be reliable in making smart decisions for performing safe overtaking maneuver's and constantly generate a trajectory profile which are collection of discretized points and cancels out the colliding trajectories with respect to the parameters of the observed vehicle. Future motion is predicted using prediction models linking some control inputs, car properties and external conditions to the evolution of the state of the vehicle. This leads us to the research question:

- How does the prediction model and/or cost functions of the trajectory planner affect collision risk, comfort and successful overtakes ?

In the selected scenario's geometry of the path to be followed by the vehicle is already known and based on that a reference path for the overtaking maneuver is set. So, when a fast moving vehicle detects a vehicle in front with slower speed an overtaking command should be initiated and be completed before the two lane highway converges into an one lane road. Several methods have been developed to obtain trajectory between the initial state and final goal like RRT and Dijkstra algorithms [7] but they are not so reliable in simultaneously computing alternative goal states. The capability to set alternative goal states with high frequency is of importance when the vehicle in front have varying speeds to make smart decisions about maneuvers and reduce the rates of collisions. Therefore, an algorithm based on quartic polynomial as demonstrated by Werlings trajectory generation algorithm [8] will be implemented which is sensitive to traffic changes and calibrate final multiple states with high frequencies and will vary the motion model to assist in predicting future states and trajectory profile generation. The motion models which will be considered are Constant Velocity (CV) and Constant Acceleration (CA) in order to achieve an increase in rates of successful overtakes [9] which would result in decrease of collision rates.

1.3 Purpose

The purpose of this work is to allow autonomous vehicles make smart decisions in order to reduce the collision rates and in turn increase the ratio of completed versus cancelled overtakes. This thesis will be conducted at Alten where they have an scaled model of inbuilt track replicating a road scenario where the two lane highway merges into one lane. The vehicle will have to generate a trajectory at every instant of time that it will follow in order to make a successful overtake maneuver and if its not possible then it would apply brakes and slow down to follow the vehicle in front into an one lane road.

The demonstrator at Alten consists of an RC car which is equipped with a Zedboard. The Zedboard is using a Xilinx Zynq-7000 All programmable system on the chip. The chip includes an FPGA [10] and a dual ARM Cortex-A9 processor. One core of the ARM processor is used to run safety-critical processes and the other core is used to run safety-critical processes and the other core is used to run petalinux for entertainment applications. FPGA will be used for sensor data acquisition.

1.4 Goal

In this project there are six master thesis students working together on the same vehicle. This means that there are both individual goals and team goal.

1.4.1 Individual Goal

Individual goal is to generate trajectory at every instant of time and investigate how varying the prediction model can lead to a decrease in collision rate and increase in rate of successful overtakes. The algorithm will then be implemented on the RC car which will detect and communicate with the vehicle in front moving with varying speed and make decisions. It will execute an overtaking maneuver if its possible before the two lane highway converges into one lane road or will just cancel the overtaking operation and follow the vehicle in front into an one lane road if it would result in collision of the vehicles.

1.4.2 Team Goal

The team goal is to develop a demonstrator consisting of vehicles that can safely handle a number of commonly occurring traffic situations. The demonstrator consists of two vehicles whereof one follows the other using Cooperative Adaptive Cruise Control (CACC). The operations that will be performed are:

- The vehicle detects pedestrian and traffic signs and make decision for what maneuver to perform.
- The vehicle issues a warning for the above situations and conveys it to another vehicle.
- The vehicle performs an overtaking when it detects another slower vehicle in front.

1.5 Method

This study is divided into two phases namely Literature review and Implementation phase. Literature review will be performed to by gathering information of the subject from well-known and accepted sources. The methodology used for this phase is a qualitative approach as it will be done to assist in understanding and gaining a better insight into the problem.

The system is evaluated for the scenario when the ego car approaches a slow moving vehicle in front and perform a safe overtaking maneuver before the two lane highway merges into a single lane road and avoid any possible risk of collisions as depicted in figure 1.1. The lane change maneuver algorithm considers both the the aspect of longitudinal and the lateral motion planning but in this thesis only longitudinal motion planning problem will be considered in order to reach the desired final position preferably ending up ahead of the other vehicle with a pragmatic approach to reduce the risk of collision at the critical merging zone. In addition to constraints on the state of the ego vehicle during the motion planning, care should be taken in making sure the ego vehicle does not plan a motion which cannot be executed during the evasive maneuver due to the physical limitation of the vehicle like actuator saturation.

The ego car follows a longitudinal trajectory defined by a trajectory planner which will generate a set of trajectories and by using the prediction model to predict the motion of the observed vehicle perform collision check with the observed vehicle and this is carried out by constant communication between the vehicles. The planner needs to make discrete decision to either go ahead or follow the observed vehicle into the single lane road. Colliding reference trajectories are removed, and a cost function determines the best of the remaining ones. The observed vehicle has varying speeds, therefore the ego vehicle has to generate trajectory in every instant of time to make decision of either to follow the observed vehicle to a single road or accelerate and be the first one to reach the depicted merge area so the presented system should be able to handle the maneuver without experiencing any collisions.

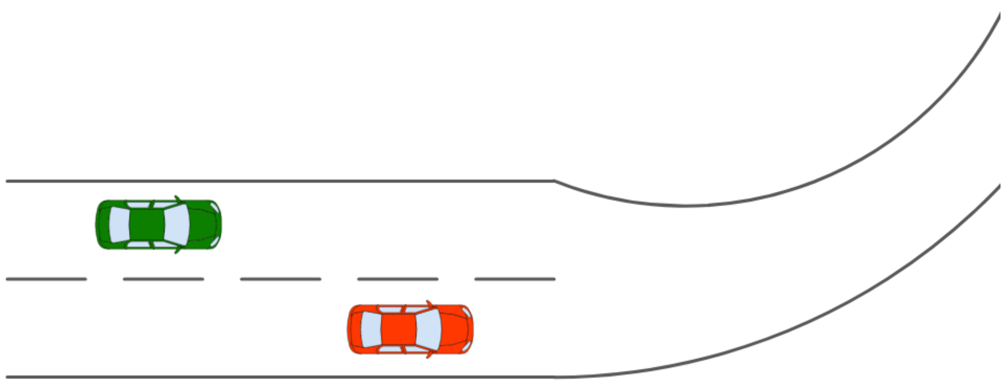


Figure 1.1: Traffic scenario with vehicles traveling on a two lane highway that merges to a single lane road. The vehicle in green wants to perform the overtaking operation. The vehicle in red is the leading slower vehicle.

An experimental research method will be used to investigate system performance. The methodology used for the implementation phase is a quantitative approach as the team will be comparing the data and facts to make a conclusion. It will mainly include simulations of the system which could be implemented on the RC cars.

1.6 Alten system

1.6.1 Hardware platform

Hardware imposes constraints during the design space exploration phase of the project. Therefore, in order to reach an optimal design solution, it is imperative to have a thorough understanding of the hardware architecture and its capabilities. This project is restricted to the used of the Zedboard as a hardware platform. The Alten MCS uses two Operative Systems (OS) to create temporal and spatial separation between safety-critical and non-critical applications using TrustZone. In its current setup the Real-Time Operative System (RTOS) FMP by TOPPERS [11] is used for safety-critical applications. For non-critical applications, the General Purpose Operative System (GPOS) Linux kernel 4.4 is used. The trajectory planner should be implemented on the RTOS of the demonstrator as shown in figure 1.2 . The steering function and the lateral control of the vehicle is executed on the Raspberry Pi which is responsible for the lane detection. The Raspberry Pi is connected to the Zedboard through UART connection.

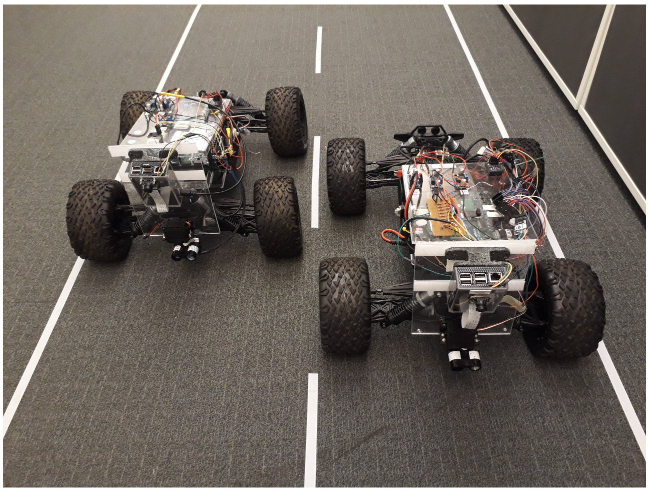


Figure 1.2: Demonstrator at Alten.

1.6.2 Field programmable gate array (FPGA)

Since their introduction in the 1985, Field Programmable Gate Arrays (FPGAs) have become increasingly important to the electronics industry. They have the potential for higher performance and lower power consumption than microprocessor. FPGAs are suitable for an extremely diverse number of applications including: sorting and searching; signal processing; audio, video and image manipulation; cryptography; packet processing; random number generation and logic emulation [10]. FPGA programming language is commonly called Hardware Description Language because it is actually used to describe or design hardware. The two major Hardware Description Languages are Verilog HDL and VHDL. The explicit difference between FPGA programming and software programming is the way that its instructions are executed. In terms of the execution of instructions, instructions in software programming (C, Ada, etc.) are sequentially executed while Verilog/ VHDL instructions in FPGA programming are mostly parallelly executed. The environment perception systems are located on the FPGA.

1.6.3 System architecture

The figure 1.3 shows the existing system architecture of the demonstrator at Alten. The focus on this thesis will be on the longitudinal and lateral component of RTOS system for the implementation of the trajectory generation algorithm. A simple trajectory planner is implemented on the demonstrator which upon detecting a slower vehicle in front executes an overtaking maneuver.

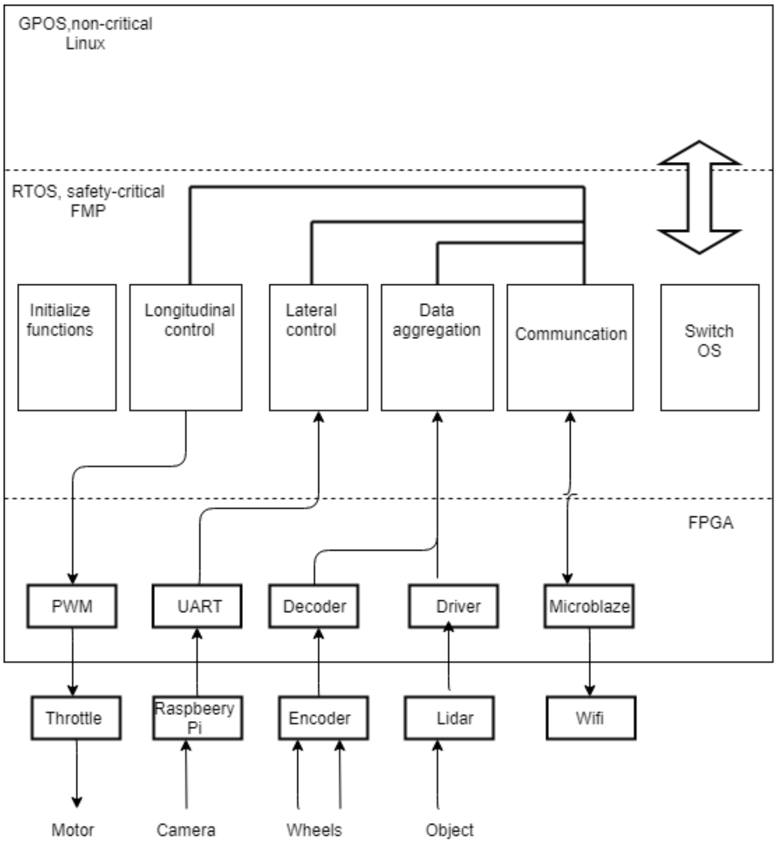


Figure 1.3: The system architecture of the demonstrator at Alten

1.7 Ethical considerations

The work performed in this thesis project is carried out in an as ethical and sustainable way as possible. Any type of misleading information, as well as representation of primary data findings in a biased way is avoided. The automated vehicles make decisions based on speed, weather, road conditions, distance and other data gathered by a variety of sensors, including cameras, LiDARS and radars. An automated car will calculate a course of action based on how fast it is traveling as well as the speed of an object in its path. The main challenge is in gathering and processing the necessary data quickly enough to avoid dangerous circumstances in the first place. The system has to be tested before it can be send for production. There is also risk of hacking the V2V communication between the vehicles which could result in wrong decision making.

1.8 Delimitations

The context of this scope is limited to two vehicles and strictly with respect to a highway scenario. It will be constrained to the Xilinx Zynq-7000. The scope of this work extends to investigate how prediction model affect the trajectory generation, collision risk and comfort. We will be considering the overtaking maneuver as a longitudinal motion planning problem and the result will be based on simulations. Only predefined parameters will be considered for the implementation of the trajectory planner.

Chapter 2

State of the art

This chapter will go through the relevant articles and research that has been done in the prediction model, trajectory planning, key definitions, automotive standards and system architecture.

2.1 SAE standards J3016

SAE Internationale J3016 provides a common taxonomy and definitions for automated driving in order to simplify communication and facilitate collaboration within technical and policy domains [12]. The six levels of driving automation that span from no automation to full automation are:

- No Automation (level 0) : The full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems.
- Driver Assistance (level 1): The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task.
- Partial Automation (level 2): The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task.
- Conditional Automation (level 3): The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene.
- High Automation (level 4): The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task,even if a human driver does not respond appropriately to a request to intervene.
- Full Automation (level 5): The full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver.

A lot of research has been focused towards level 4 and level 5 of driving automation to achieve full autonomy which is not lacking of technological and social challenges.

2.2 Theoretical preliminaries

This section introduces the key conceptual terms commonly used in the literature within the files of automated vehicles and in context of this thesis to formulate and solve the problem.

The set of independent attributes which uniquely define the position and orientation of the vehicle according to a fixed coordinate system is termed the configuration vector. Consequently, the set of all the configurations of the vehicle constitute the configuration space. The set of attribute values describing the condition of an autonomous vehicle at an instance in time and at a particular place during its motion is termed the "state" of the vehicle at that moment [13]. The most common set of attributes, defined as a vector, which are used to express the state of a vehicle are the position (x, y), the linear velocities (vx, vy) and angular velocities (xx, xy). Subsequently, state space represents the set of all possible states that a vehicle can be in. As will be seen in the sections 2.5, the mathematical representation of a state space differs from the approach taken by vehicle planning.

Given a configuration space or a state space, planning is a computationally intensive task, demanding high memory utilization. The lowest level of planning is concerned with planning a smooth trajectory adhering to vehicular dynamics and such a plan is chalked out on a small (local) search space of high dimensional states. Path is expressed as a continuous sequence of configurations beginning and ending with the boundary configurations, i.e. the initial configuration and the terminating configuration respectively. On the other hand, trajectory is represented as a sequence of states visited by the vehicle, parameterized by time. Trajectory planning is concerned with the real-time planning of the actual vehicle's transition from one feasible state to the next, satisfying the vehicle's kinematic limits based on vehicle dynamics and constrained by the navigation comfort, lane boundaries and traffic rules, while avoiding, at the same time, obstacles including other road users as well as ground roughness and ditches. Trajectory planning is parameterized by time as well as acceleration or velocity and is frequently referred to as motion planning. During each planning cycle, the path planner module generates a number of trajectories from the vehicle's current location, with a look-ahead distance, depending on the speed and line-of-sight of the vehicle's on-board sensors, and evaluating each trajectory with respect to some cost function to determine the optimal trajectory. Trajectory planning is scheduled at regular intervals; the length of which largely depends on the frequency of receiving fresh sensor data [14].

2.3 Decision-making hierarchy

Automated cars are essentially autonomous decision-making systems that process a stream observations from on-board sensors such as LIDARs, cameras and GPS/INS units. These observations, together with prior knowledge about the road network, rules of the road, vehicle dynamics, and sensor models, are used to automatically select values for controlled variables governing vehicular motion. The hierarchy of decision making system of a typical self-driving car is explored in the paper [15] and illustrated in figure 2.1. At the highest level a route is planned through the road network. This is followed by a behavioral layer, which decides on a local driving task that progresses the car towards the destination and follows by the rules of the road. A motion planning module then selects a continuous path through the environment to accomplish a local navigation task. A control system then reactively corrects errors in the execution of the planned motion. After the behavioral layer decides on the driving behavior to be performed in the current context, which could for example, an overtak-

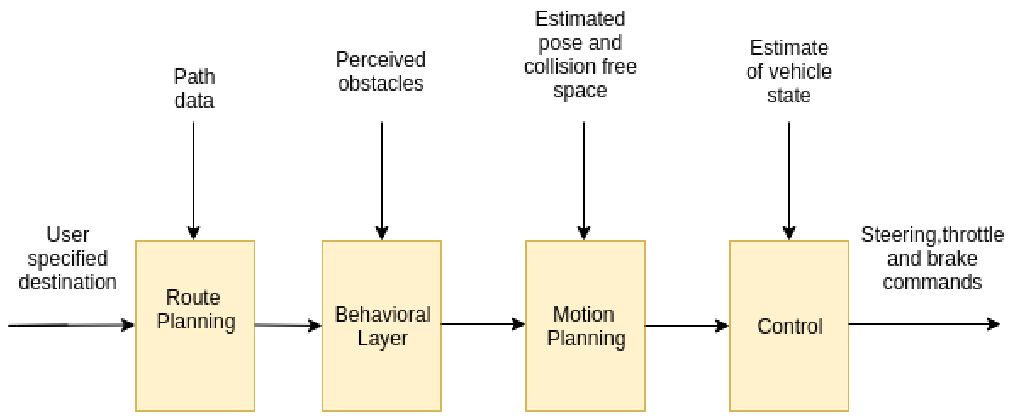


Figure 2.1: Illustration of the hierarchy of decision-making process in automated vehicles.

ing then that has to be translated into a path or trajectory that can be tracked by a low-level feedback controller. The resulting path should be free of collision. In this thesis we will be looking at the motion planning component since it is responsible for computing a safe, comfortable, and dynamically feasible trajectory from the vehicle's current configuration to the goal configuration.

2.4 Environment perception

In the automotive field there are several applications for which fusion of data of various sensors is necessary. For all around coverage and for supporting at the same time a lot of different applications, such as Adaptive Cruise Control (ACC), front/rear collision mitigation, parking aid, lane change and lane keeping support. However, as addressed in [16] this approach suffers from serious limitations such as the perception environment of the vehicle cannot go beyond the sensing range, the system is not able perceive the situation in time in order to warn the driver and suggest a corrective action etc. Therefore, there is a lot of ongoing research on cooperative vehicles.

Cooperative behavior refers to the ability of vehicles to cooperate with each other by means of communication, so that the intentions and positions of other vehicles are known to each other. Vehicles need to share the information they posses. V2V communication enhances the perception of environment not only through its own sensors, but through the sensors of other vehicles. Different applications of cooperative driving are closely highlighted in [17]. Different warnings refer to on-road incidents or traffic alerts in order to improve traffic fluency or even safety by avoiding accidents. Some applications are warnings of hazards, approaching emergency vehicles, slow vehicles, or traffic jams. In our case assisting with overtaking maneuver. Summarizing, they create new information or share existing information in a way that was not feasible before.

2.5 Vehicle Dynamics

The estimation of a vehicle's dynamic state is one of the most fundamental data fusion tasks for intelligent traffic applications. In order to increase the stability and accuracy of the estimation, the vehicles are mostly assumed to comply with certain motion models which describe their dynamic behavior. This approach has the ability to predict the vehicle's position in the future which for instance be used to calculate collision probability. A comparison of the different motion model, with different levels of abstraction, are discussed in detail in [2]. This section describes the motion models used in the implementation of the proposed algorithm.

The motion model and prediction approach can be classified into three levels of abstraction namely physics-based motion model, maneuver-based motion models and interaction-aware motion model [18]. Physics-based motion models consider that the motion of vehicles only depends on the laws of physics. Future motion is predicted using dynamic and kinematic models linking some control inputs, cars properties and external conditions to the evolution of the state the vehicle and are accurate for only short term prediction. Physics-based motion models for vehicles remain the most commonly used motion models for trajectory prediction and collision risk estimation in the context of road safety.

2.5.1 Kinematic physics-based motion model

The vehicle motion is described as a function of time only based on the kinematic relations between the parameters of interest like position, velocity and acceleration. The models proposed in the literature are numerous [2] and an overview is provided in figure 2.2. The most commonly used motion model are CV, CA, CTRV and CTRA. The simplest of different models based on low level of complexity are constant velocity (CV) and constant acceleration (CA). The advantage of using this model is the linearity of the state transition equation which allows an optimal propagation of state probability distribution but they assume straight motions and are not able to take rotation into the account.

Constant Turn Rate and Velocity (CTRV) and Constant Turn Rate and Acceleration (CTRA) model are sometime also referred to as curvilinear models. CTRV assumes that the vehicle will keep the same velocity and turning rate as measured at $t = 0$. CTRA combines the ability to predict turning

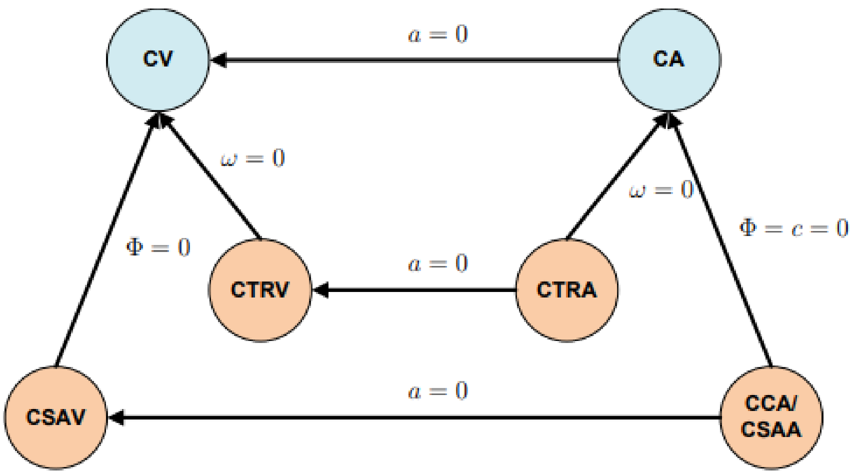


Figure 2.2: Overview about linear and curvilinear motion models [2].

motion from CTRA with the ability to capture accelerating or braking motion. Both CTRA and CTRV assume that there is no correlation between the velocity v and the yaw rate ω. In order to avoid this problem, the correlation between v and ω can be modeled by using the steering angle ϕ as constant and derive the yaw rate from v and ϕ. The resulting model is called Constant Steering Angle and

Velocity (CSAV). Again, the velocity can be assumed to change linearly, which leads to the Constant Curvature and Acceleration (CCA) model. CV and CA motion model are selected for this thesis as only the aspect of longitudinal trajectory generation is considered and not lateral trajectory generation since our scenario consists of reducing the collision probability when the two lane highway merges into a single road.

2.5.1.1 Constant Velocity

Constant Velocity (CV) is characterized by the equation as

$$x = \dot{x}\Delta t \tag{2.1}$$

It is to be noted that the motion model can be used to describe the motion of the vehicle in any direction, i.e., longitudinal and lateral and by using Δt as time step size state space model describing the longitudinal motion of the vehicle is given by

$$\bar{x}(t+\Delta t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \Delta t \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \tag{2.2}$$

2.5.1.2 Constant Acceleration

Constant Acceleration (CA) can be represented by equations of motion using Δt as time step size and state space model describing the longitudinal motion of the vehicle is given by

$$\dot{x} = \dot{x}_0 + \ddot{x}\Delta t \tag{2.3}$$

$$x = x_0 + \dot{x}_0\Delta t + \frac{1}{2}\ddot{x}\Delta t^2 \tag{2.4}$$

$$\bar{x}(t+\Delta t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} + \Delta t \begin{bmatrix} \dot{x} + \frac{\ddot{x}\Delta t}{2} \\ \ddot{x} \\ 0 \end{bmatrix} \tag{2.5}$$

Here, x, $\dot{x}$ and $\ddot{x}$ denote the position, velocity and acceleration respectively.

2.6 Trajectory generation

Trajectory generation, also called motion planning, generates a path as a function of time. Unlike a spatial path which is simply a sequence of coordinates (x_k, y_k) that can be followed with an arbitrary speed, the trajectory can be expressed as

$$(x_k, y_k, t_k), k = 0, 1....N \tag{2.6}$$

In many applications the trajectory functions as a time varying reference value for position and/or velocity being fed to a controller. There are several reasons for having a trajectory rather than a constant speed reference. In many cases a strong acceleration and jerk is undesired, which may be prevented by having the trajectory within the given limits of acceleration and jerk. In addition, the trajectory planner may also function as a path finder in order to avoid obstacles in the environment. Finally, a movement may have timing requirements for the position and speed which make trajectory planning necessary for avoidance of dynamic obstacle.

The motion planning problems in dynamic environment or with dynamic constraints may be more suitably formulated in the trajectory planning framework, in which the solution of the problem is a trajectory, i.e., a time-parameterized function $\pi(t) : [0, T] \rightarrow \chi$ prescribing evolution of the configuration of the vehicle in time, where χ is the configuration space of the vehicle and T is the planning horizon [15].

2.6.1 Planning techniques

A great amount of trajectory generation techniques have been taken from mobile robotics and modified to face the challenges of road networks and driving rules. These planning techniques were classified in four groups, according to their implementation in automated driving: graph search, sampling, interpolating and numerical optimization [7]. The most relevant path planning algorithms implemented in motion planning for automated driving are:

Graph search

In automated driving, the basic idea is to traverse a state space to get from point A to point B. This state space is often represented as an occupancy grid that depicts where objects are in the environment. A path can be set implementing graph search searching algorithm which gives different solution by visiting states in the grid. The so called graph search methods discretized the configuration space χ of the vehicle and represent it in the form of a graph and then search for a minimum cost path on such graph [15]. For example, Dijkstra Algorithm [19] and A* [20]. The main problem associated with these algorithms are that the search is not heuristic and the resulting path is not continuous which is not good for real time application and evaluation of jerk.

Sampling based

Sampling based planning handle complex problems in high-dimensional spaces but usually operate in binary world, which aims to find out collision-free solutions rather than the optimal path [21]. The approach mainly consists of randomly sampling the configuration space or state space and looking for connectivity. For example, RRT [22]. The algorithm always converges to a solution if there is a enough time provided but the resulting trajectory is not necessarily optimal with respect to jerk.

Numerical optimization

Many optimization problems in aeronautics and astronautics, in industrial robotics and in economics can be formulated as optimal control problems. These methods aim to minimize or maximize a function subject to different constrained variables. There are many different techniques available to solve optimization problems, where the solutions are found by either analytic or numerical procedures. Numerical optimization methods obtain approximate optimal paths, or paths that are relatively close in the spatial domain to the ideal optimal path. For example, MPC [23] is useful when you want to take other users into consideration but time consuming since the optimization of function takes place at each motion state and that discrete decisions in the trajectory planning problem cause these problems to become non-convex and hard to solve in real time.

Geometric methods

In the geometric approach it is believed that since roads are designed with specific geometric models, the trajectory of the vehicle can be modeled with geometric features. A trajectory may be generated by using interpolation between different states at the given time instants. Interpolation is defined as the process of constructing and inserting a new set of data within the range of a previously known set (reference points). This means that these algorithms take a previously set of knots (e.g., a given set of way-points describing a global road map), generating a new set of data (a smoother path) in benefit of the trajectory continuity, vehicle constraints and the dynamic environment the vehicle navigates. In the presence of obstacles, it suffices to generate a new path to overcome the event and then re-entry the previously planned path [2]. For example, quintic polynomial [8] and Bezier curves [24] have a low computational cost.

2.6.2 Quartic polynomial

Trajectory planning for point to point motion maps position as function of time between specified points. Velocity and acceleration along the trajectory can be computed by differentiating position with respect to time, and for smooth path, a velocity cannot have discontinuities or the specified trajectory would require infinite acceleration. With a given set of via points and constraints, a smooth trajectory $(s(t))$ can be specified. The acceleration can be specified at each point using the quartic polynomial therefore it is preferred over the cubic polynomial trajectory method as discontinuity of acceleration makes the derivative of acceleration (jerk) infinite at each via point and causes an impulsive jerk in the motion of the vehicle. Therefore, a fourth order polynomial is required to define the trajectory between two via points. Three initial values of position, velocity and acceleration at the beginning of the maneuver and as well as three final values of maneuver time, velocity and acceleration in longitudinal direction form in total six constraints to generate the trajectory. The state of the vehicle at every time instant is denoted as

$$\bar{s}(t) = \left[x(t), \dot{x}(t), \ddot{x}(t)\right]^T \tag{2.7}$$

where x, $\dot{x}$ and $\ddot{x}$ denote the position, velocity and acceleration respectively. The quartic polynomial describe the vehicle in terms of time and it's formulated as below:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \tag{2.8}$$

Therefore, the constraints at points $q(t_0)$ and $q(t_f)$, can be defined as follows,

$$x(t_0) = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 \tag{2.9}$$

$$\dot{x}(t_0) = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 \tag{2.10}$$

$$\ddot{x}(t_0) = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 \tag{2.11}$$

$$\dot{x}(t_f) = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 \tag{2.12}$$

$$\ddot{x}(t_f) = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 \tag{2.13}$$

The parameters a_0 - a_f can be determined by solving the system of equations.

2.7 Vehicular motion

In recent years, a number of studies have been done on intelligent vehicle highway systems (IVHS) and automated highway system (AHS). There are two basic tasks for vehicle control within an IVHS and AHS. Longitudinal control mainly refers to vehicle speed regulation to maintain adequate spacing between vehicles and involves the vehicle's throttle and brake. On the other hand, lateral control includes automatic vehicle steering to follow a track reference, while maintaining good ride comfort.

2.7.1 Lateral motion

Lateral control keeps the vehicle in the center of the lane and steers the vehicle into an adjacent lane assisting in lane change maneuver, while maintaining good passenger comfort. Lateral control is concerned with lane keeping, turning, lane changing and avoiding objects that might appear in front of the vehicle. Although many parts of the vehicle can contribute to the lateral dynamics of the vehicle; the most important ones as mentioned in [13] are tires, steering system and suspension system. It should be noted that braking and acceleration can also significantly change the response of the vehicle since they result in a load transfer.

2.7.2 Longitudinal motion

The lane change maneuver algorithm considers both the the aspect of longitudinal and the lateral motion planning. Vehicle longitudinal motion control aims at ensuring passenger safety and comfort. It is an important aspect in dynamic collaborative driving i.e. when multiple vehicles should co-ordinate to share road efficiently while maintaining safety. The longitudinal references for merging necessitate the prediction of the observed vehicle especially in the case of collision check. For example, knowing the approximate position, velocity and acceleration of the vehicle in observation and by applying prediction models one can estimate its future positions by integration.

2.8 Collision prediction

The classic trajectory planning method is a close-loop between the trajectory generator and the collision detector. The planned paths not only should be safe but is also the optimal path with respect to predefined conditions such as the shortest length, the minimum acceleration jerk, and the minimum energy consumption. The Time To Collision (TTC) is widely used and studied for collision prediction and risk assessment. A low value of TTC indicates that a collision is bound to occur if none of the vehicles initiate any evasive action. The job of the collision avoidance system is to check the distance between the ego vehicle and the obstacle at every time instance through the maneuver and keep a safe distance to avoid collisions. [25] employs the method of representing the vehicles as a circle that encloses the entire vehicle. The collision detector, because of simplicity, the vehicle is modeled as one circle that enclose the entire vehicle with a radius R to demonstrate the idea of vehicle to obstacle distance. Also, extra spaces considered for the collision detector might be good because of uncertainty occurred from sensors, system modeling and so on which can at least ensure the safety of both vehicle. In the figure 2.3 each vehicle is represented by one circle with radius R which is half of the length of the vehicle.

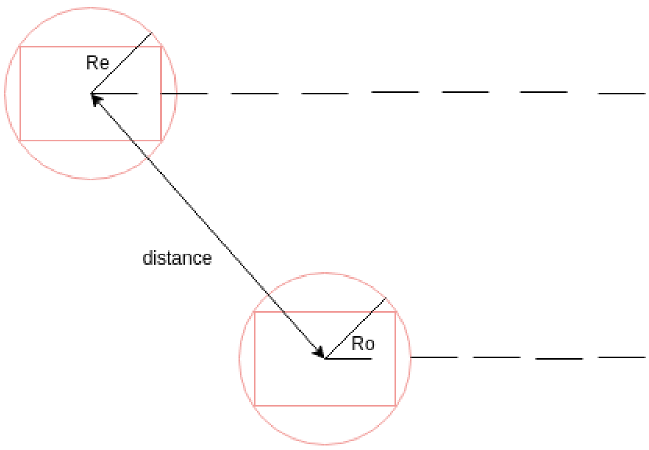


Figure 2.3: Distance between the vehicles.

The distance between two vehicles is :

$$distance = \sqrt{(x_e - x_o)^2 + (y_e - y_o)^2} \tag{2.14}$$

in order to ensure the safety of both vehicles, the following equation should hold true,

$$minimum\ distance > R_e + R_o \tag{2.15}$$

These equations can be intuitively explained from the geometry location of two vehicles. TTC is defined as the time that is needed to cover the distance between the lead and the following vehicle with the relative speed between the lead and the following vehicle [26] and is computed as:

$$TTC_i = \frac{X_{i-1}(t) - X_i(t) - l_i}{\dot{X}_i(t) - \dot{X}i - 1(t)} \tag{2.16}$$

with $\dot{X}_i$ denoting the speed of vehicle i, X_i the position of vehicle i, l_i the length of vehicle i and $i-1$ the vehicle ahead of vehicle i. Also, the position of both the vehicles should be taken at every sampling time instance from the beginning to the end of the maneuver. The rectangular shape can also be used to represent the original vehicle which are specified with a width W, length L and a height H which encloses the vehicle. The rectangle are replaced with a series of circles with the radius R as seen in figure 2.4. Therefore the minimum distance should be greater than the summation of radius of the two cars. This is the reason why prediction of the observed vehicle is needed. Based on collision check, the collision avoidance trajectory can be generated.

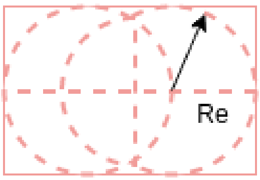


Figure 2.4: Vehicle represented as a rectangle.

2.9 Cost function evaluation

A cost function is a performance index which is minimized or maximized. For each state transition, a cost criterion is penalizing undesirable action effects. Trajectories are ranked against a cost function. The cost function can be decomposed into several sub-cost functions which are:

- Feasibility cost: With respect to collision avoidance and vehicular capabilities.
- Safety cost: Preferring to keep some buffer distance between vehicles, good visibility etc.
- Comfort cost: Minimizing the jerk in longitudinal direction by having some constraints.
- Efficiency cost: With respect to speed and time to goal.

After predicting a number of scenario sequences corresponding to the different strategies, a cost is computed for each of them. In the computation, all the predicted scenarios from $t = 0$ to $t = t_{horizon}$ are evaluated, as shown in equation 2.17:

$$C_{strategy(i)} = \sum_{t=0}^{t=t_{horizon}} C_{scenario(i,t)} \tag{2.17}$$

As a summary, after ranking up the candidate trajectories the trajectory that best match the driving policy is chosen via a set of weighted cost functions.

Chapter 3

Method and Implementations

This chapter deals with the methods used to solve the problem of trajectory generation during overtaking maneuver as described in the Chapter 2. This chapter also includes the problem formulation and description of the implementation of the entire system. MATLAB is used as the programming language to implement the proposed method and illustration of the results.

3.1 Algorithm demonstration

The proposed system only deals with trajectory generation in longitudinal direction and can be divided into several different modeled tasks. The complete overview of the complete algorithm is given in the Figure 3.1 where the information flow in the system is presented.

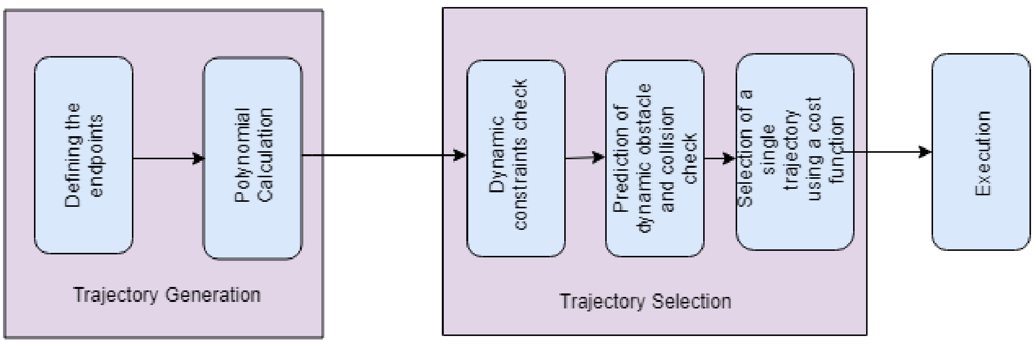


Figure 3.1: Overview of the proposed system.

3.1.1 Trajectory Generation

Definition of Endpoints

This component is responsible for providing the endpoints of the trajectories to be generated which are fed to the polynomial calculation component to generate a set of trajectories. The critical point now is to properly define the start and end conditions. Those constraints are displacement, velocity, acceleration and time of maneuver as seen from the Equation 2.16 in Section 2.6.2. The assumptions that are made are that the displacement at initial state is always zero, which is also known as the local displacement as it is the point on the car from which the set of trajectories will be generated and is represented by x_o. The final displacement is the output of polynomial calculation component which are collection of discretized points along each generated trajectory and is represented as x_f. The velocity at initial state is equivalent to the velocity of the vehicle at particular instant of time, which is represented as $\dot{x}_o$ and final state is equivalent to the desired speed of the vehicle represented as $\dot{x}_f$. The acceleration at initial and final state of the maneuver is longitudinal direction is assumed to be zero for simplicity of trajectory calculation and are represented as $\ddot{x}_o$ and $\ddot{x}_f$ respectively.

To ensure continuity at every time step for the complete maneuver, the position of ego vehicle which here is the global parameter and represented as X is the relative position of the ego vehicle with respect to the road, velocity and acceleration, the start points have to be placed somewhere on the previous generated trajectory where the position has not yet been reached in practice by the vehicle. This starting condition at every time step are decided depending on the number of discretized points consumed for every selected trajectory before the next iteration because before all the discretized points of the selected trajectory are consumed a new set of trajectories are generated as soon as new informations and data from sensors are received. This points for the implementation are selected through linear interpolation between consecutive time steps. The time at initial state is always assumed to be zero and represented as t_o. The final time is equivalent to the time required for the maneuver and represented as t_f. The sampling time was taken to be 0.1 s respectively.

Polynomial calculation

Using the endpoints as the input to the quartic polynomial 3.1 a set of trajectories are computed as solution to a simplified problem, where we consider only the ego vehicle generating trajectories without any constraints and cost function computation. The output from this component would result in x_f which is a collection of points along the trajectory. By putting the endpoints in the matrix form as in Equation 3.2 in Chapter 2 the quartic polynomial can be solved.

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \tag{3.1}$$

Equation 2.9 and 2.14 as in Chapter 2 can be put in the matrix form as,

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix} \tag{3.2}$$

Simulation of sample trajectory generation

A set of 100 trajectories were generated at every time step. For visualization purpose 5 sample longitudinal trajectories are simulated here using the quartic polynomial after defining the start and end conditions. In the Figure 3.2 you can see different possible end states for the ego vehicle are plotted. The Figure 3.2 also shows the corresponding velocity and acceleration profile associated with different end states. It can be seen that the initial velocity of the ego is $5\ m/s$ which increases to the desired velocity of $10\ m/s$

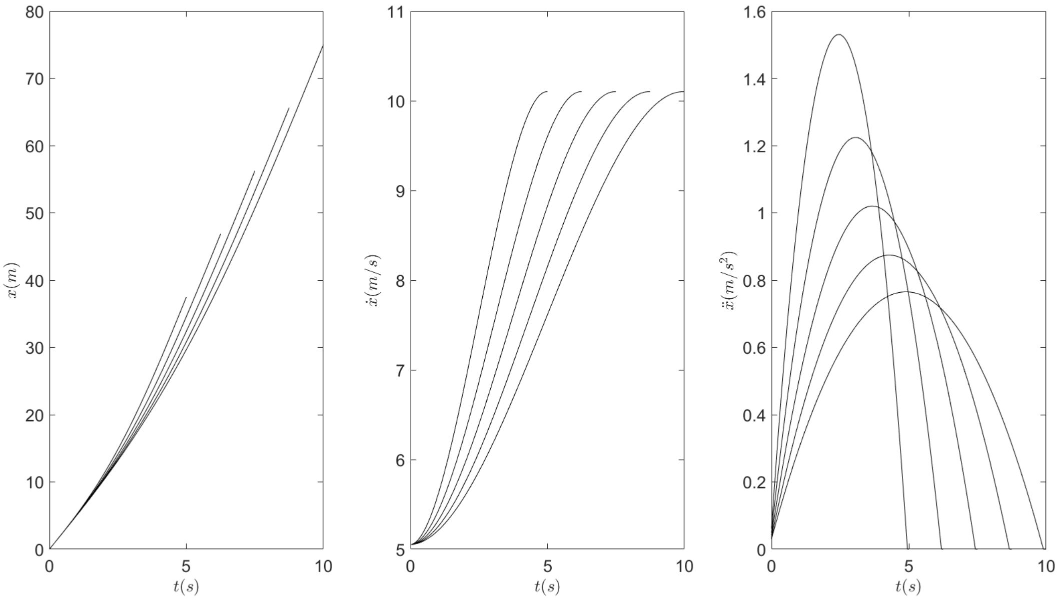


Figure 3.2: The set of possible longitudinal trajectory, velocity and acceleration profile is generated.

3.1.2 Trajectory Selection

The trajectory selection component has two main purposes. The first one is to ensure that all the dynamic constraints of the ego vehicle and the constraints associated with the interaction of the observed vehicle such as collision detection are respected. The second one is to implement a cost function for the selection of a single trajectory.

Dynamic constraint check

The longitudinal motion of the ego vehicle is modeled with respect to the feasibility of the vehicle and respecting the comfort of the passenger using the kinematic physics-based motion model as described in Section 2.5.1 of Chapter 2. Longitudinal velocity $\dot{x}$, longitudinal acceleration $\ddot{x}$ and longitudinal jerk $\dddot{x}$ are subjected to the set of constraints:

$$\dot{x}_{min} \leq \dot{x}(t) \leq \dot{x}_{max} \tag{3.3}$$

$$\ddot{x}_{min} \leq \ddot{x}(t) \leq \ddot{x}_{max} \tag{3.4}$$

$$\dddot{x}_{min} \leq \dddot{x}(t) \leq \dddot{x}_{max} \tag{3.5}$$

Prediction of dynamic obstacle and collision check

This component is based on the data received by the sensors stores the available information to determine how the ego vehicle will traverse in the longitudinal direction. After, removing trajectories based on vehicle dynamic constraints and ensuring comfort for the passenger, this component serves the purpose of removing trajectories which would result in collision with the observed vehicle at the merging point. This is done by checking the position of the ego vehicle with respect to the surrounding vehicle and maintaining a safe distance which was taken as $20\ m$ between them at every instant of time. As mentioned in Chapter 2 the two prediction model utilized to predict the future position of the observed vehicle are Constant Velocity (CV) and Constant Acceleration model (CA) which can be represented in its state transition form as,

$$\bar{x}(t+\Delta t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \Delta t \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \tag{3.6}$$

$$\bar{x}(t+\Delta t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} + \Delta t \begin{bmatrix} \dot{x} + \dfrac{\ddot{x}\Delta t}{2} \\ \ddot{x} \\ 0 \end{bmatrix} \tag{3.7}$$

Here, x, $\dot{x}$ and $\ddot{x}$ denote the position, velocity and acceleration respectively.

Equation 3.8 ensure that a safety gap is maintained between the ego and observed vehicle to avoid any risk of collisions.

$$20 > R_e + R_o \tag{3.8}$$

where R_e and R_o is the radius of the ego and observed vehicle respectively as in the Figure 2.3 in Chapter 2.

Selection of a single trajectory using cost function

This component is responsible for selecting a single trajectory which will be passed on the controller for the execution. After passing through the previous components, there are still more trajectories left. Hence, a cost function will be utilized to ensure that all the other trajectories are discarded and only a single trajectory is selected for execution. The cost function utilized here selects the trajectory that will ensure maximum distance transverse by the ego vehicle in the longitudinal direction to reach the merging point as quick as possible. The cost function implemented is shown in the Equation 3.9.

$$J = \max\left(\int_0^{t_f} \dot{x}\, dt\right) \tag{3.9}$$

After going through all the different components a single trajectory at a single time stamp is selected for execution as shown in the Figure 3.3

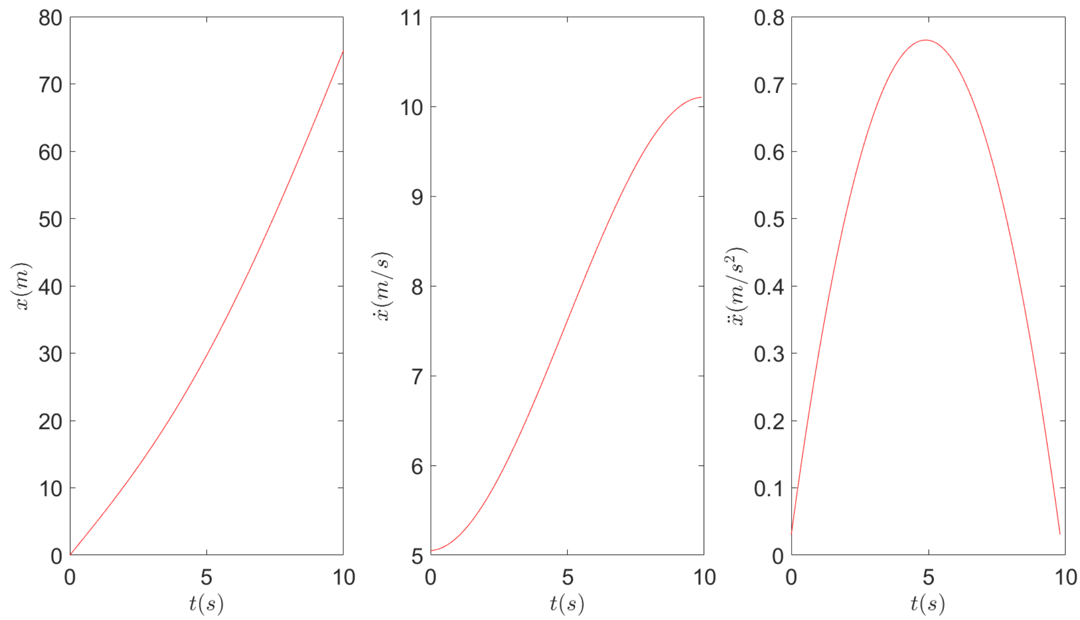


Figure 3.3: The final trajectory selected to transverse in the longitudinal direction. The selected velocity and acceleration profile are also shown in the figure.

Chapter 4

Results and Discussions

This chapter shows the main results that were obtained from simulations. The overtaking scenario for longitudinal movement of the vehicles is simulated in three different scenarios where the observed vehicle is moving with constant velocity, linear increasing and random velocity respectively. Both the Constant Velocity (CV) and Constant Acceleration (CA) prediction models were used for three different scenarios and their respective results obtained were compared. For scenario 1, the CV model will give perfect prediction, for scenario 2, the CA model will give perfect prediction and scenario 3, is the most realistic and will be challenging for both the models.

4.1 Scenario 1: Constant velocity

The first scenario assumed that the observed vehicle have a constant velocity of $10\ m/s$ and the ego vehicle has an initial velocity of $5\ m/s$. In figure 4.1 simulation of the first scenario is shown. The scenario is set up such that if both vehicles keep their setpoint velocities, there would be collision at the merge point, and trajectory planner makes sure that it does not happen, while the cost function is minimized. The position, velocity and acceleration of the ego vehicle is plotted in blue and of the observed vehicle in red respectively. The merging point is chosen to be $220m$. We see here that the ego vehicle reaches the merging point before the observed vehicle. It can also be noted that the ego vehicle crosses the observed vehicle at $13\ s$ and the vehicle is continuously decelerating to reach the desired velocity of $10\ m/s$.

Then a mean error was calculated by the averaging the errors over the total simulation time. Constant velocity (CV) prediction model gave us an mean percentage error equal to $zero$ with respect to the actual position of the vehicle at every instant of time which shows us that Constant Velocity (CV) model gives the desired result since the observed vehicle is moving with a constant velocity. Constant acceleration (CA) prediction model is not applicable.

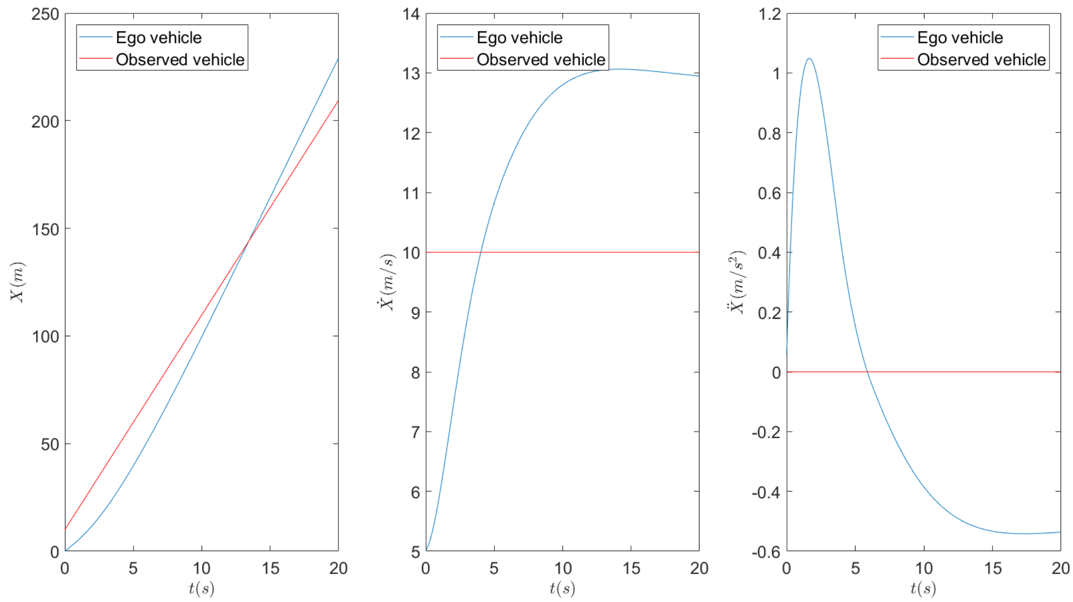


Figure 4.1: The distance traveled, velocity and acceleration of the two vehicles in longitudinal direction before reaching the merging point is shown. The ego vehicle is plotted in blue and of the observed vehicle in red respectively. Constant velocity (CV) prediction model was utilized

4.2 Scenario 2: Constant acceleration

The second scenario assumed that the observed vehicle have a linear increasing velocity with initial velocity of $5\ m/s$ and acceleration of $0.5\ m/s^2$ and the ego vehicle has an initial velocity of $5\ m/s$. In figure 4.2 a simulation of the second scenario is shown. The position, velocity and acceleration of the ego vehicle is plotted in blue and of the observed vehicle in red respectively. We see here that the ego vehicle reaches the merging point before the observed vehicle. Here both the prediction model were applied. It can be also seen that the dynamic constraints imposed on acceleration of the ego vehicle of $\pm\ 1.5$ is followed. The Constant velocity (CV) prediction model gave us an mean percentage error = 0.1123 which was higher than the mean percentage error = 0 given by the Constant acceleration (CA) prediction model. The results shows that Constant acceleration (CA) prediction model worked better for the vehicle with linearly increasing velocity.

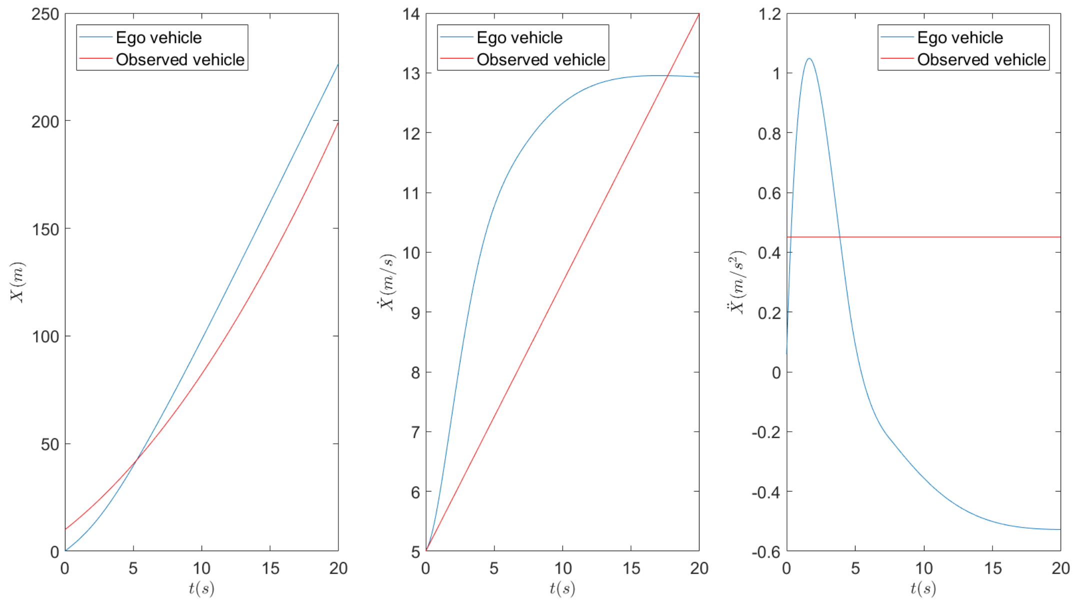


Figure 4.2: The distance traveled, velocity and acceleration of the two vehicles in longitudinal direction before reaching the merging point is shown. The ego vehicle is plotted in blue and of the observed vehicle in red respectively. Constant acceleration (CA) prediction model was utilized

4.3 Scenario 3: Random Velocity

The third scenario shows that the observed vehicle have a random velocity and random acceleration with respect to time. The ego vehicle has an initial velocity of $5\ m/s$. The three different outcomes that can occur here is that the ego vehicle reach the merging point before the observed vehicle both staying within their respecting dynamic constraints, both the ego vehicle and observed result in collision and third outcome is that if its not possible for ego vehicle to overtake and it decides to brake and is successful in braking without resulting in collision. Here both the prediction models were applied.

In figure 4.3 a simulation is shown where the position, velocity and acceleration of the ego vehicle is plotted in blue and of the observed vehicle in red respectively. It can be seen that the ego vehicle is able to easily overtake the observed vehicle without resulting in any collisions. The vehicle overtakes the observed vehicle at $6\ s$. The same velocity vector was used for both Constant velocity (CV) and Constant acceleration (CA) prediction model.

The second outcome in figure 4.4 in which the ego vehicle is plotted in blue and of the observed vehicle in red respectively. It can be seen that the velocity of the observed vehicle suddenly increases too high and not within the dynamic feasibility of the ego vehicle to overtake the observed vehicle. Hence, here the ego vehicle choses to follow the observed vehicle into the single lane road by canceling the overtaking maneuver.

The third outcome in figure 4.5 it can be seen that the ego vehicle and the observed vehicle would result in collision at the merging area as the safety gap of $20\ m$ between the vehicles is not maintained and it is not able to brake or overtake safely due to the dynamic constraints imposed on the vehicle

In order to mimic a physical scenario, piecewise random velocities were chosen for the observed vehicle. Due to this, continuity could not be ensured between the different segments of piecewise velocities leading to sudden changes in acceleration and kinks in jerk profile at the time-stamp common between two segments.

Modeling of observed vehicle

For modeling of observed vehicle traveling at random velocity equations 4.1, 4.2 and 4.3 were used. The dynamic constraints were given according to the equations 3.3, 3.4 and 3.5 in Chapter 3.

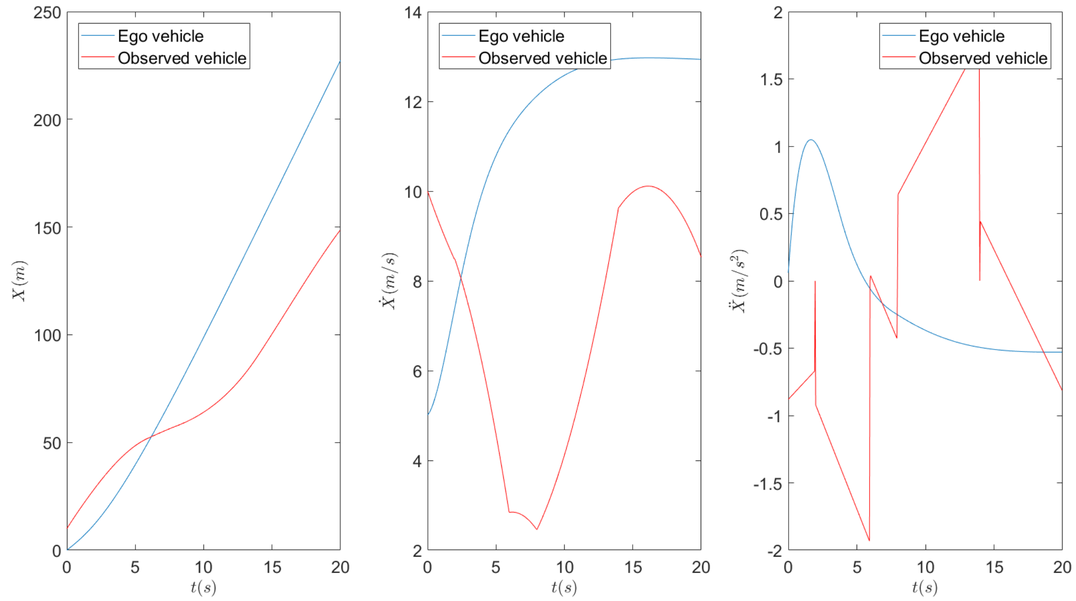
$$\ddot{X}_0 = \ddot{X}_0 + \dddot{X}_0 \Delta t \tag{4.1}$$

$$\dot{X}_0 = \dot{X}_0 + \ddot{X}_0 \Delta t + \frac{1}{2}\dddot{X}_0 \Delta t^2 \tag{4.2}$$

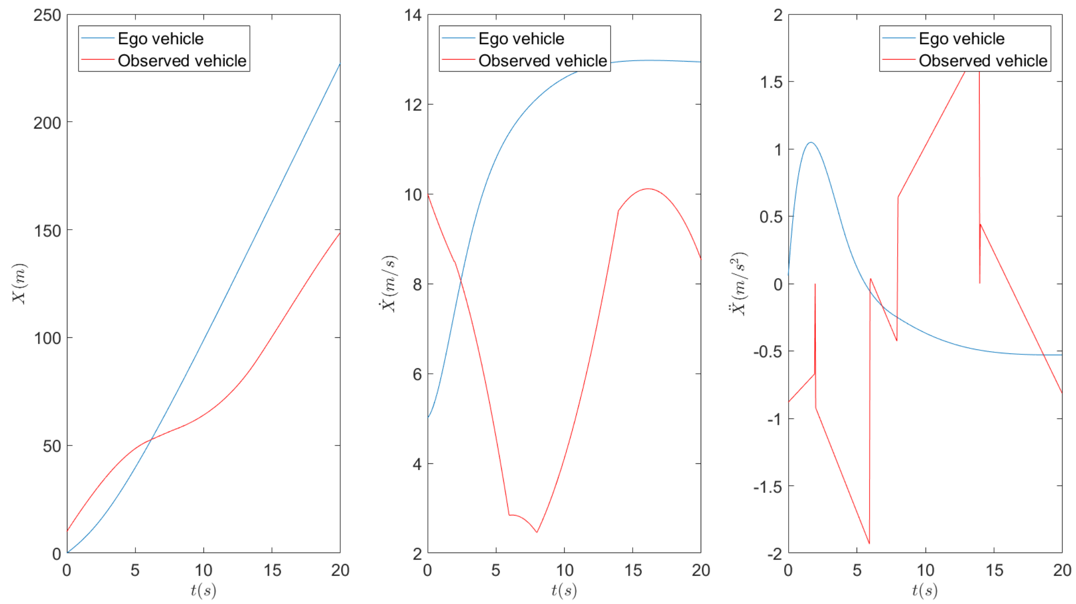
$$X = X_0 + \dot{X}_0 \Delta t + \frac{1}{2}\ddot{X} \Delta t^2 + \frac{1}{6}\dddot{X}_0 \Delta t^3 \tag{4.3}$$

where $\dddot{X}_0$ is the constant jerk and $\ddot{X}_0$ is the initial acceleration. The constant jerk model can be used to model vehicle motion where the acceleration is not constant and it should be noted that a zero jerk value in the constant jerk model results in the constant acceleration model.

Here both the prediction model were applied. The algorithm was executed for a number of iterations for both the prediction models. Because the time steps taken for both the models $(0.1\ s)$ is 200 times smaller than the total simulation time which is $20\ s$. Therefore, the displacement plots resulting from both the models have a similar shape and final position. This makes the difference between the models not immediately obvious. However, if the temporal resolution of the simulation is reduced, the CV model (first order approximation) performs worse than the CA model (second order approximation) due to new trajectories being generated less frequently. Thus, $0.1\ s$ was chosen as the temporal resolution for this simulation through trial and error method.

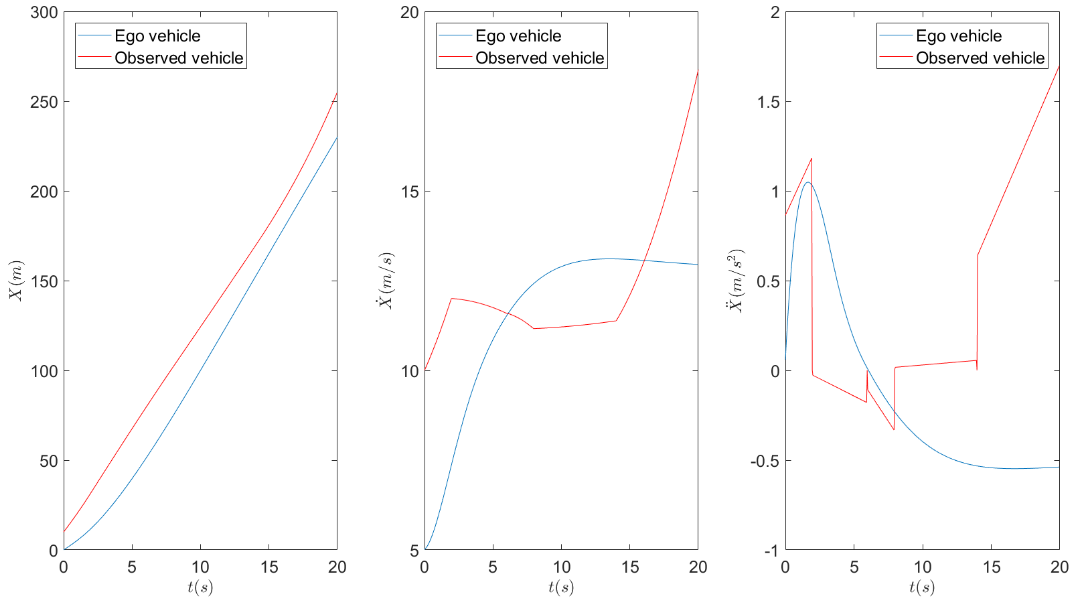


(a) Constant velocity (CV) prediction model.

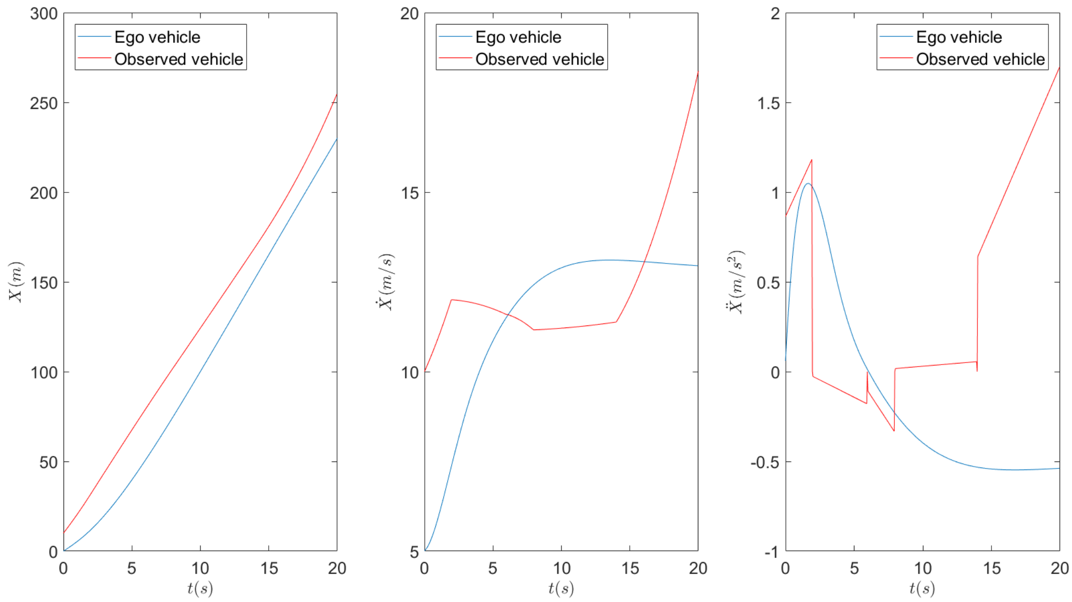


(b) Constant acceleration (CA) prediction model.

Figure 4.3: The distance traveled by the two vehicles in longitudinal direction before reaching the merging point. The ego vehicle is plotted in blue and of the observed vehicle in red respectively. It can be seen that ego vehicle reaches the merging point before the observed vehicle.

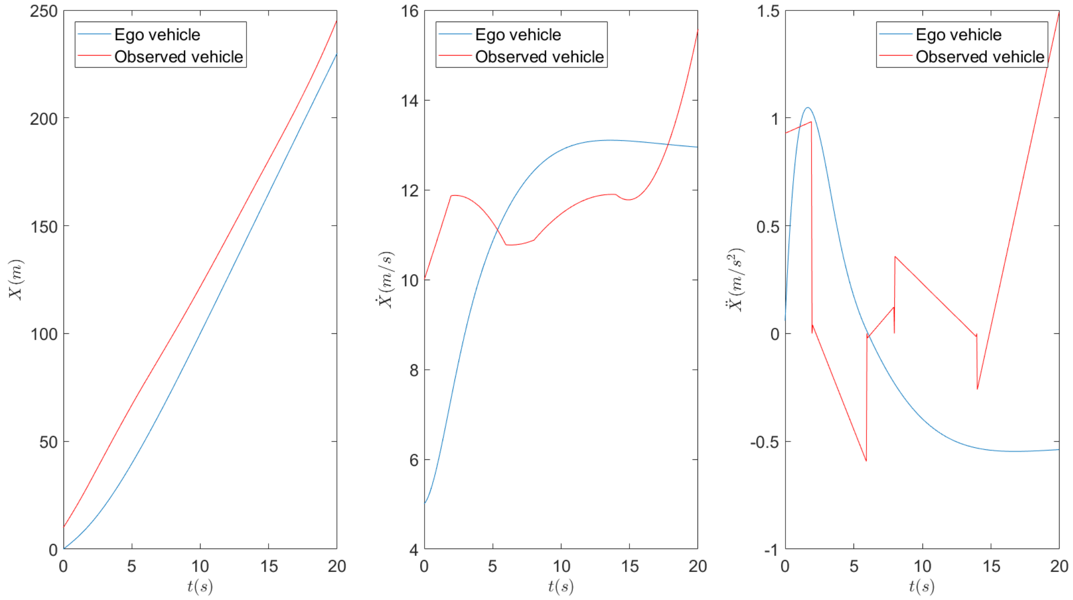


(a) Constant velocity (CV) prediction model.

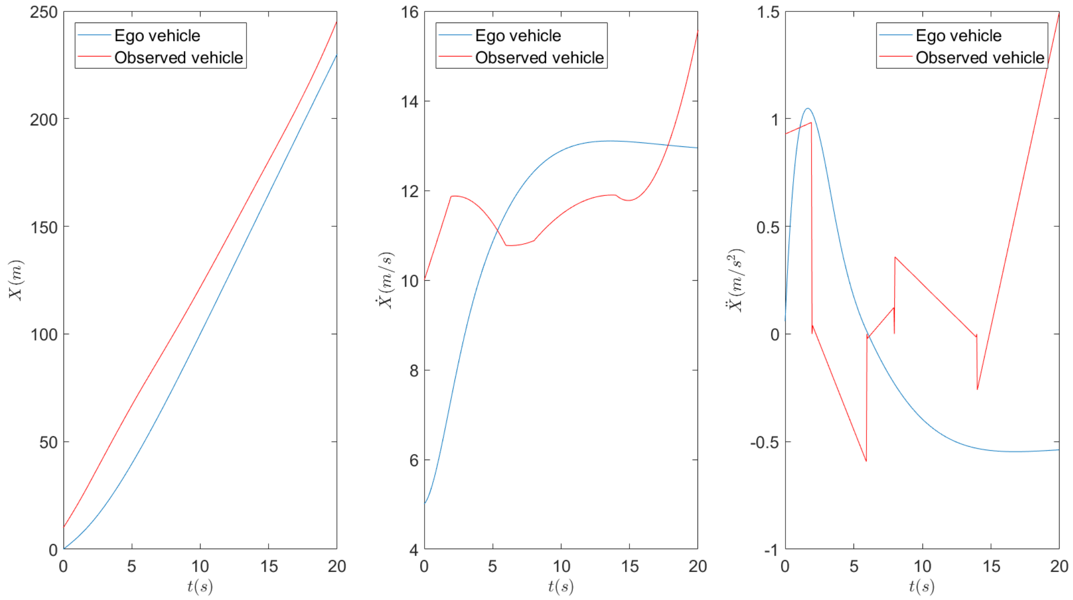


(b) Constant acceleration (CA) prediction model.

Figure 4.4: The distance traveled by the two vehicles in longitudinal direction before reaching the merging point. The ego vehicle is plotted in blue and of the observed vehicle in red respectively. It can be seen that the ego vehicle applies brake and chooses to follow the the observed vehicle into the merging point.



(a) Constant velocity (CV) prediction model.



(b) Constant acceleration (CA) prediction model.

Figure 4.5: The distance traveled by the two vehicles in longitudinal direction before reaching the merging point. The ego vehicle is plotted in blue and of the observed vehicle in red respectively. It can be seen that the ego vehicle and the observed vehicle result in a collision.

The performance of both the models was checked by computing the error in predicted position at every time step. Then a mean error was calculated by the averaging the errors over the total simulation time. This process was repeated 10 times with piecewise random velocity as input. Finally, the mean error from the 10 iterations were used to check for robustness of the model and calculate a final error. The Constant velocity (CV) prediction model gave an final mean percentage error for a number of iterations equal to 0.1466. The final mean percentage error given by the Constant Acceleration (CA) prediction model for a number of iterations was equal to 0.0159. The Constant Acceleration (CA) prediction model worked better for the vehicle traveling with random velocity. The figure 4.6 shows mean percentage error for 10 iterations with random velocities.

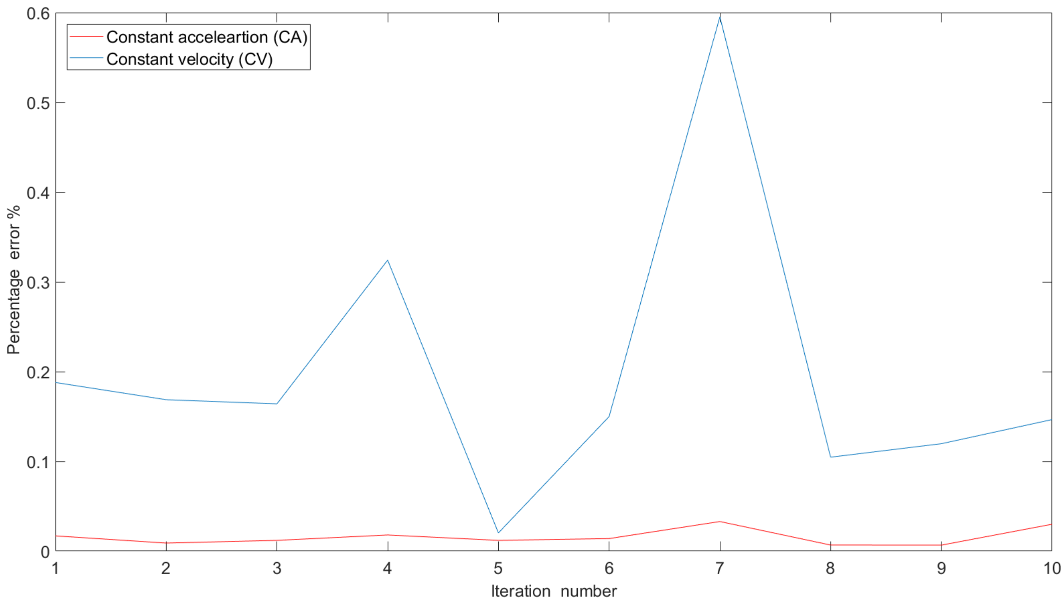


Figure 4.6: The percentage error of 10 iterations with random velocities.

The table summarizes the results obtained in this chapter. It can be seen that using Constant acceleration (CA) prediction model will give us better results when compared to the Constant velocity (CV) prediction model to detect the future motion of the surrounding vehicle and reduce the risk of collisions for the considered scenarios.

	Scenario		
Prediction model	Constant velocity	Linear increasing velocity	Random velocity
Constant Velocity(CV) error	0	0.1123 %	0.1466 %
Constant Acceleration(CA) error	N.A	0 %	0.0159 %

Table 4.1: Mean percentage error given by Constant Velocity (CV) and Constant Acceleration (CA) prediction model for different scenarios

Chapter 5

Conclusion

This chapter presents the conclusions drawn from the project. The chapter is divided into two sections and the conclusions drawn from the literature study are presented first and then the conclusions drawn from the implementation and results are presented.

5.1 Literature study

Overall conclusion drawn from the literature study are that the trajectory planning problem, is complex, mainly when it comes to the detection of the surrounding environment. There are many aspects to consider when selecting a trajectory to be executed by the vehicles in both the longitudinal and lateral direction for different type of scenario encountered by the vehicle. Many different aspects of the problem have been investigated in previous studies and many different working solutions have been proposed for different scenarios. There are different prediction model illustrated in the literature for different purposes and for this thesis Constant velocity (CV) and Constant acceleration (CA) prediction model were chosen as we considered the overtaking maneuver only in the longitudinal aspect and this model work well-enough for simple linear motion. Many different ways to check for collision have also been discussed in the literature. FPGA constitute a promising solution because of parallelism so different sensors can be executed in parallel to improve the safety of the system and reduce the risk of collisions.

5.2 Implementation and results

The conclusion drawn from the implementation and result section was quartic polynomial had many benefits of having a low computational cost and continuous concatenation of curves are possible. It is important in trajectory planning to cancel out trajectories which would dynamically be not feasible and result in an increase risk of collision with the surrounding vehicle. The research question was to compare different prediction model and compare their affect on collision risk and result in successful overtakes. For the scenario addressed in this thesis Constant Acceleration (CA) prediction model gave better result when compared to Constant Velocity (CV) prediction model and had an lower risk of collision which increases the number of successful overtakes and while doing so the jerk dynamic constraints were always considered to ensure that the trajectory generated are within the comfort zone of the passenger.

Chapter 6

Future Work

It can be seen from the literature study that there are different prediction models available. In this thesis only two prediction model were utilized but different prediction model might give better results for different scenarios and reduce the risk of the collisions. The algorithm can be extended for lateral motion of the vehicle also instead of only considering the aspect of longitudinal motion. The study can be extended to study with a few different initial speeds for each scenario and/or another cost function that would be more conservative. It would be also interesting to look at different aspects which results in collisions of the vehicles and how can it be avoided.

Only the first scenario was considered for the implementation on the demonstrator where the observed vehicle was considered to be traveling at a constant velocity since cooperative communication between the vehicles was not enabled. Therefore, that should be addressed and enabled between the vehicles, so that the observed vehicle can be driven manually with random velocity and depending on which the ego vehicle executes a selected trajectory and compare the results of different prediction model in real-time application.

Trajectory generation algorithm in the future can implemented on the FPGA to benefit from its parallelism and reduce the computational cost and increase its efficiency.

Bibliography

[1] W. H. Organization, "Global status report on road safety," *World Health Organization 2015*, 2015.

[2] G. W. Robin Schubert, Eric Richter, "Comparison and evaluation of advanced motion models for vehicle tracking," *Chemnitz University of Technology, Germany*, 2008.

[3] T. V. . M. P. . I. Spyropoulou, "Intelligent transport systems and effects on road traffic accidents: state of the art," *Institute of Transport Economics, Gaustadalle´en 21, NO-0349 Oslo, Norway. VTT Transport and Logistics Systems P.O. Box 1000, VTT FI-02044, Finland. National Technical University of Athens, 5, Iroon Polytechniou str, Zografou 15773, Greece*, 2007.

[4] V. A. B. . P. Ioannou, "Personalized driver/vehicle lane change models for adas," *University of Southern California, Los Angeles, CA 90089 USA*, 2014.

[5] S. B. (DESTATIS), "Individual data enquiry about overtaking accidents," *Wiesbaden*, 2009.

[6] SafeCOP, "Safecop," *http://www.safecop.eu/*, 2016.

[7] V. M. David González, Joshué Pérez and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems, Volume: 17 , Issue: 4 Pages: 1135 - 1145*, 2017.

[8] J. Z. Moritz Werling, Sören Kammel and L. Gröll, "Optimal trajectories for time-criticalstreet scenarios using discretizedterminal manifolds," *The International Journal of Robotics Research*, 2012.

[9] D. V. Stephanie Lefevre1 and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH JOURNAL*, 2014.

[10] P. H. Leong, "Recent trends in fpga architectures and applications," *The Chinese University of Hong Kong, Hong Kong*, 2008.

[11] T. T. project, "Introduction to the toppers/fmp kernel," *https://www.toppers.jp/en/fmp-kernel.html*, 03/03/2018.

[12] S. INTERNATIONAL, "On-road automated driving (orad) committee, "j3016 taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," 2018.

[13] A. Eskandrian, "Handbook of intelligent vehicles," *Center for Intelligent Systems Research The George Washington University 801 22nd Street, NW, Phillips Hall 630 Washington, DC 20052 USA*, 2012.

[14] W.-H. C. L. D. Christos Katrakazas, Mohammed Quddus, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *School of Civil and Building Engineering, Loughborough University, Loughborough LE11 3TU, United Kingdom.School of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, United Kingdom*, 2015.

[15] S. Z. Y. D. Y. Brian Paden, MichalCap and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles,Massachusetts Institute of Technology, Cambridge, MA, USA , Pages 33-55*, 2016.

[16] G. T. Panagiotis Lytrivis and A. Amditis, "Sensor data fusion in automotive applications," *Sensor and Data Fusion, Institute of Communication and Computer Systems Greece*, 2009.

[17] M. A. R. Radu Popescu-Zeletin, Ilja Radusch, "Vehicular-2-x communication," *State-of-the-Art and Research in Mobile Vehicular Ad hoc Networks, Chapter 2*, 2010.

[18] D. V. Stephanie Lefevre1 and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH JOURNAL*, 2014.

[19] Y. Y. Huijuan Wang, Quanbo Yuan, "Application of dijkstra algorithm in robot path-planning," *2011 Second International Conference on Mechanic Automation and Control Engineering, North China Institute of Aerospace Engineering Langfang, China*, 2011.

[20] R. W.ZENG, "Finding shortest paths on real road networks:the case for a*," *International Journal of Geographical Information Science ,Faculty of Information Engineering,China University of Geo sciences, Department of Geography,University of California, USA*, 2008.

[21] J. C. Leonard Jaillet and T. Simeon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics (Volume: 26 , Issue: 4 , Pages:635 - 646)*, 2010.

[22] A. L. Romain Pepy and H. Mounier, "Path planning using a dynamic vehicle model," *2006 2nd International Conference on Information Communication Technologies, Institut d'Electronique Fondamentale*, 2006.

[23] H. N. Morteza Farrokhsiar, Graham Pavlik, "An integrated robust probing motion planning and control scheme: A tube-based mpc approach," *Robotics and Autonomous Systems Volume 61, Issue 12, Pages 1379-139, School of Engineering, University of British Columbia, Kelowna, Canada*, 2013.

[24] M. Fixter, "Fast trajectory generation using bezier curves," *2007 IEEE Intelligent Vehicles Symposium Pages:505 -510, School of Computer Science and Software Engineering, The University of Western Australia*, 2003.

[25] Y. O. Guoqing Xu, Li Liu and Z. Song, "Dynamic modeling of driver control strategy of lane-change behavior and trajectory planning for collision prediction," *Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Sciences/The Chinese University of Hong Kong, Shenzhen 518055, China*, 2012.

[26] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Analysis Prevention Volume 35, Issue 3, May 2003, Pages 427-433, Swedish National Road and Transport Research Institute, S-581 95 Linköping, Sweden*, 2003.

[27] G. N. Tripathi1 and V.Rihani2, "Motion planning of an autonomous mobile robot using artificial neural network," *www.arxiv.org, India*, 2012.

[28] F. Mohseni1 and M. Morsali, "Decoupled sampling based planning method for multiple autonomous vehicles," *2018 IEEE Intelligent Vehicles Symposium (IV),Pages: 1 - 6 Changshu, Suzhou, China*, 2017.

TRITA ITM-EX 2018:681