Master's Thesis Report for
Master in Automotive Engineering for Sustainable Mobility (AESM)

Academic Year 2019-2020

Maneuver decision for autonomous vehicles, considering vehicle dynamics and perception uncertainties

Presented By :
TUPE Dheeraj

Performed at :
Heudiasyc Laboratory,
UMR CNRS/UTC 7253,
57, Avenue de Landshut,
60200 Compiègne.

Academic Supervisor :
M. JOUANGUY Julien
Professor
ISAT,Univ. Bourgogne,
49 rue Mlle. Bourgeois,
58000 Nevers. France
Email : Julien.Jouanguy@u-bourgogne.fr

Research Supervisor :
M. ADOUANE Lounis
Professor
Université de Technologie de Compiègne,
Rue Roger Couttolenc,
60200 Compiègne. France
Email : lounis.adouane@hds.utc.fr

Mme. TALJ Reine
Chargée de Recherche CNRS
Université de Technologie de Compiègne,
Rue Roger Couttolenc,
60200 Compiègne. France
Email : reine.talj@hds.utc.fr

Acknowledgement

I would like to express my special thanks of gratitude to my Research Supervisor's ***M. ADOUANE Lounis*** and ***Mme. TALJ Reine*** here at Heudiasyc Laboratory who gave me this golden opportunity to do this research work on the topic "**Maneuver decision for autonomous vehicles, considering vehicle dynamics and perception uncertainties**", which helped me a lot to gain knowledge in this rapidly evolving field and gave an insight of how research work is carried out.

Also, I cannot express enough thanks to my Academic Supervisor ***M. JOUANGUY Julien*** at ISAT, Nevers, for his continuous support and constant guidance throughout the duration of my Internship.

I also extend a special thanks to **Heudiasyc Laboratory and UTC**, for their help by providing access and funding to carry out this research work without any restriction.

My completion of this internship could not have been accomplished without the support of my supervisors who were there to solve my doubts and correct me whenever necessary and gave direction to my research.

Finally I would like to express my gratitude to my family and friends for their continuous encouragement and motivation provided from time to time.

Abstract

Looking at the recent history of automakers, automation has been at the center of technological innovation aiming to prioritize life's of humans within it as well as pedestrians. Initially this research was focused on automating some specific driving tasks such as locking speed of the vehicle so that it can cruise at a constant speed on a long stretch of highway, or automating the steering system to keep the vehicle lane centered so that driver might get some relief from continuous driving on a long journey on an highway where much attention is not required.

More recently, a lot of research has been made both by industry and institutions to completely automate the driving process. This potential societal benefits of this fully automated driving are numerous, including safety on roads, easy mobility for disables and elderly, increasing human productivity. But to make this technology highly commercialized, this technology should be able to safely share the common space of mobility with human drivers. This intend to say that this technology should be able to interact with human drivers and also understand other drivers and predict their intentions as a normal driver does to avoid collisions or causing unnecessary chaos. For this purpose, autonomous vehicles must be capable of making decision while keeping all these factors into consideration.

The Objective of this internship is to make an autonomous vehicle navigate on a road, by avoiding collisions with other road occupants, in a highly dynamical situations and in the presence of uncertainties (such as perception uncertainties, unknown intentions of other road vehicles, etc). This report addresses the decision making approach by presenting a model which is testing on a common highway scenario.

Keywords : Decision-Making, Markov Decision Process, Partially Observable Markov Decision Process, Model Predictive Control, Vehicle Dynamics.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

Since the first competition of the DARPA Grand Challenge, the research & development in autonomous vehicles starting picking pace and set in motion the development of this new technology which has a potential to radically transform the transportation industry. This transformation was in consideration of improvements in the road safety, in which human error accounts for almost 94-96% of all the motor vehicles crashes ([1]). So the shift towards safer road conditions for both the vehicle occupants as well as other road users is more in favor of Autonomous vehicles, but the transition from human operated vehicles to autonomous vehicles is a little tricky task to achieve. Many Advance Driver Assistance System (ADAS) have been development and deployed in mass scale which can handle certain driving tasks under certain conditions. Some ADAS features to be named are Cruise Control (CC) , Adaptive Cruise Control (ACC), Lane Keeping Assist (LKA) etc. Which help the driver to achieve some of the driving tasks under controlled situations automatically. These ADAS systems certainly help in bridging the gap between the human driven vehicle and autonomous vehicle. Combining all these ADAS driving tasks will help us to get closer to creating an autonomous vehicle but decisions to make switch between these tasks will make the vehicle a truly autonomous one.

SAE has defined and classified automated vehicles in 6 levels based on their degree of autonomy and it is shown in Table 1.1 below. In level 0, the driver has the full control of the vehicle, along with this, some lane departure warning systems, blind spot warning systems, etc. may be used. Level 1, the driver can automate either longitudinal or lateral control under certain conditions. Adaptive Cruise Control, lane keeping assistant, etc can be few examples. Level 2, the vehicle can take full control to itself under controlled conditions. But, the driver is required to continuously monitor the vehicle and be ready to take back the control at any given time. Some examples like highway driving assistants, park assist systems come under this criteria. Level 3, the vehicle can drive in fully autonomous mode under certain conditions without the requirement of driver to continuously monitor the vehicle, but, must be ready to take back the control if required. Tesla have commercialized their autopilot technology under this level. Other examples include traffic jam autonomous driving systems and the human chauffeur system. Level 4 , requires no driver inputs during any part of the journey, these level vehicles are capable of driving in an Geo-fenced areas without any human intervention. The vehicle is capable of making its own decision under any circumstances in this Geo-fenced area. The approach used by Waymo which has deployed its Geo-fenced autonomous vehicles fleet in Phoenix, Arizona, USA. Level 5, this level represents the highest level of automation possible in driving. This level of vehicles will be able to navigate anywhere without having any constrains made by Geo-fencing and other things. This level represents unconditional full autonomy which every company is trying to achieve as their final objective.

Level of Autonomy	Name	Characteristics
0	No Automation	Driver has full control.
1	Driver Assistance	Longitudinal or lateral control can be controlled by the vehicle is possible.
2	Partial Automation	Longitudinal and Lateral control can be controlled by vehicle under certain conditions. But driver should be available to take control at any time.
3	Conditional Automation	Vehicle can assume full control under certain circumstances. Driver is not required to monitor the system continuously but must be ready to take control when required.
4	High Automation	Vehicle can assume full control in a specific Geo-fenced areas. No action need from driver.
5	Full Automation	Vehicle has full control at all times under all circumstances.

Table 1: Levels of driving automation (SAE International, 2016)

This thesis particularly focuses on the decision making part for the autonomous vehicle while taking into consideration the limitations of vehicle dynamics. For this purpose, a particular highway scenario has been taken into consideration to look out for various parameters to deal with while making a maneuver decision for a vehicle, considering uncertainties.

This manuscript starts a brief section 1.1, introduction about the laboratory, which gave me an opportunity to work on this internship under the guidance on my supervisors. Then in section 1.2, briefly stating the objectives of this thesis which were given to get started with. Section 1.3, describes the approach which was taken to achieve these objectives in brief, followed by section 2, literature review, in this section a brief summary of all the important papers which were useful for our research have been discussed. Next, section 3, describes the system architecture in details, followed by, section 3.1 where all these sub-systems will be described in brief for their selection purpose and other options which were looked for. Section 4, discusses the results and final conclusion of this thesis followed by some recommendations which would help the model to be get more refined.

1.1 Laboratory

Heudiasyc (HEUristique et DIAgnostic des Systemes Complexes) is a laboratory managed jointly by Universite de Technologie de Compiegne and CNRS (Centre National de la Recherche Scientifique). Heudiasyc's work is in the area of information and communication science, and more precisely in computer science, automatic control, robotics and artificial intelligence.

Heudiasyc laboratory mostly focuses on the investigative research and targeted research in the field of Mobility, Transport, Communication and Security. Research in this lab is organized around in three teams which are as follows,

- CID : Knowledge, Uncertainty, Data.
- SCOP : Dependability, Communication, Optimization.
- SyRI : Robotic systems in interaction.

In this organization, I was part of Team SyRI, this team mainly focuses on mini-UAVs and autonomous vehicles. The SyRI team develops embedded systems that enhance the ability of mobile robots to act autonomously in complex open environments, in some cases in interaction with human operators, and in other cases in mutual interaction with other robots.

In team SyRI, the major research topics for this team has been divided in 3 parts as per their objectives,

1 - Autonomy of mobile robots interacting with humans

2 – Multimodal ON- Board perception

3 – Multi-Robot systems in interaction

Heudiasyc laboratory has its own technology platform named PACPUS, whose objective is to provide tools and resources for experimenting on intelligent vehicles. Its specific intention was for the development, integration and testing of ADAS functions, particularly in relation to autonomous vehicles. This platform comprises of 5 experimental vehicles, each of which having a specific purpose on developing various components of an autonomous vehicle.

1.2 Internship Objective

As the title clearly suggest, this internship deals with the decision making section of autonomous vehicle. The object of the internship is to make a autonomous vehicle navigate on a road, by avoiding collisions with other road occupants, in high dynamical situations and in the presence of uncertainties (such as perception uncertainties, unknown intentions of other road vehicles, etc). This mostly deal with the decision making step where the vehicle decides to make a maneuver to overcome an obstacle and avoid collision while following a predefined global path for reaching its destination. At this level, the vehicle dynamics are not finely considered in general, they are either neglected or limited to some constraints. This fact reduces the spectrum of safe possible maneuvers that can be executed by the vehicle and do not integrate at all the passengers comfort. Also the uncertainty in the environment plays an important role in this decision making step.

So overall, in this context, this internship aims to deal with the decision making aspect in the presence of perception and modeling uncertainties, and while considering explicitly the vehicle dynamics in order to improve the safety and the fluidity of the vehicle movement.

1.3 Summary of proposed work

The internship was organized in following steps :

- Bibliographical study on vehicle dynamics, and, on maneuver planning and decision making in autonomous vehicles in presence of uncertainties.
- Consideration of a driving situation for testing the proposed approach.
- Development of a new maneuver planning approach that considers various uncertainties.
- Validation of the proposed approach on MATLAB/Simulink.

Based on these steps the work started with an in depth literature review based on a global architecture of previously build autonomous vehicles from the DARPA event which marked the true beginning for the development on autonomous vehicles. This study gave a basic platform on how exactly the initial Autonomous Vehicle were constructed and worked on planning and control parts of vehicle. Once an overview of required architecture was formed, then next step was to focus on deciding the part to be focused on for the work, that is decision making for the autonomous vehicle. As decision making can be made simple by studying of a simple scenario, for this purpose an highway overtaking scenario was taken into consideration as this maneuver contains all the aspects which were essential for testing of objectives of this thesis.

An highway overtaking maneuver contains decision making level where the vehicle needs to take multiple decisions to successfully complete this maneuver, also while considering the uncertainty of unknown intentions of the vehicle which is to be overtaken. Also, overtaking maneuver tests the vehicle dynamics limits of the vehicle taking into consideration the lateral movement and yaw rate changes for the steering of the vehicle. These characteristics of this maneuver were a good test for the testing of objectives for this internship.

Once a maneuver was decided, then this maneuver was then studied in details to get the information what all parameters are to be taken into consideration and can further be manipulated. An overtake maneuver can further be sub divided into 2 lane change maneuver and a lane keeping maneuver. One lane change maneuver at the start of overtaking maneuver, when the ego vehicle decides to start the overtaking process, then a lane keeping maneuver where the ego vehicle increases its speed or keep its constant depending on the speed of the obstacle vehicle, and then again a lane change maneuver to merge into the initial lane, which marks the end of overtaking maneuver. So, During this there are 2 main decisions which needs to be made by the vehicle to complete this maneuver, first is whether to initiate this overtake maneuver or not, and second if yes then which path needs to be followed to complete this maneuver.

For the first decision to be made by the ego vehicle that is, to initiate the overtake or not does not give us the exact maneuver to be followed by the ego vehicle. That is, this decision can be considered as a switch decision process, where as for the second decision that is, which path to follow exactly need to compute the complete maneuver which the vehicle should follow to successfully complete this maneuver without collision with obstacle vehicle. As per our objectives, the main focus need to be on the second decision making process which gives an exact maneuver path to follow. For the model which was prepared, the initial decision to overtake the slower vehicle in front of ego vehicle will be taken as soon as the TTC (Time-To-Collision) is detected to be less than 5 sec. This TTC was taken into consideration from the research mentioned in the paper, “A general formulation for time-to-collision safety indicator” in which they have mentioned that considering TTC between 4-5 sec gives sufficient time for the ego vehicle to react for avoiding the collision with other vehicle (2). For this initial decision whether to go for a overtake or not a study based on POMDP (Partially Observable Markov Decision Process) as illustrated in paper, “Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving” will rather be a good application (3) .

For the maneuver planning decision to be made, we used the MDP (Markov Decision Process), to step by step select all the best possible decisions for the vehicle to be taken for reaching the final goal position from initial position. Markov Decision Process is a discrete-time stochastic control process which provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under control of a decision maker. The driving task was first formulated as an MDP by defining environment state space, agent action space, which uses state transition model and a reward model to calculate and decide the most rewarding next action step for the agent. An detailed explanation of MDP will be provided in later section 3 of this report. Once the decision to overtake the slow moving vehicle is taken, then immediately MDP function is executed which calculates the goal point to reach based on the current speeds of both ego vehicle and the slow moving vehicle. Once the goal point is set, the MDP function then calculates and gives an output of all the available and most rewarding waypoints through which the ego vehicle must follow. To check whether the initially slow moving car has not changes its speed during the on going execution of overtaking maneuver, the ego vehicle continuously keeps on checking on the speed of the slow moving car. If it detects increase in its speed, then again this MDP function is called and once again new goal point is set for the ego vehicle to reach with modified speed of the obstacle vehicle.

As the waypoints are defined from MDP for maneuver, these waypoints are then given as an input to the Adaptive Model Predictive Controller which tunes these waypoints as per the set soft and hard constrains for the vehicle dynamics to follow so as to get a safe,comfortable and geometrically feasible set of inputs for the vehicle dynamics model. Model Predictive Controller has been given inputs of lateral position and the yaw angle from which smooth steering outputs are generated which is given as input to the 3 DOF vehicle Dynamics model.

Following is the snapshot of the successful simulation using this method.

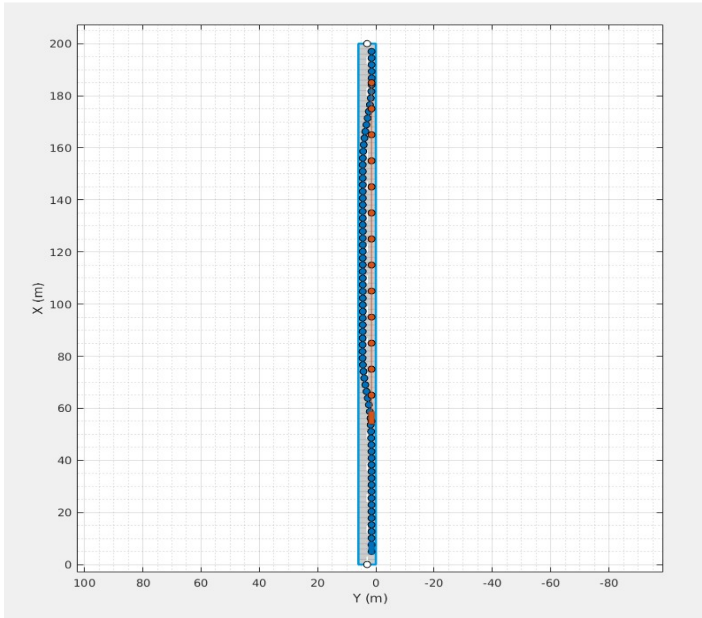

Figure 1 : Driving simulation final result

For Simulation purpose, Driving Scenario Designer application has been used. This application helps us to test different types of scenarios for different ADAS and Autonomous Vehicles. The Above figure is a snapshot from this simulation application user interface, which depicts the path followed by the 2 vehicles which are the results of algorithm implemented to avoid the collision between them. The scenario created was a highway scenario, in which a slow moving vehicle (orange car) is detected by the ego vehicle (blue car) in its path to be followed. The pre-defined path for both the vehicles was to follow same lane until they are at Time-To-Collision less than 5 sec. As soon as the ego vehicle detected the orange vehicle (speed = 15 m/s) to be slow and in TTC less than 5 secs, the ego vehicle (speed = 30 m/s) executed the MDP function to find the most feasible path to overcome this slow moving vehicle. In the figure 1, blue waypoints define the points calculated by MDP function for executing this overtake maneuver. In later section 3 we will have a detail discussion on how exactly the MDP function works and selects all best possible waypoints to execute the maneuver.

2. Literature Review

2.1 Architecture

Autonomous driving in a real urban settings was the important objective for the DARPA 2007 Urban Challenge, during which many teams participated and were able to finish this challenge. But the first to cross the finish line was team BOSS the entry of the Carnegie Mellon Team followed by Stanford University's entry "Junior". The vehicles which crossed the finish line just proved that they have a well proven software and hardware architecture which can perform all the tasks needed for the autonomous vehicle to go through Urban traffic scenario. So to understand the architecture of Autonomous Vehicles, we choose to study Team BOSS and Team Junior's vehicle architecture. This study helped us to understand how different departments such as perception, navigation, sensor interfaces interacted with each others. What all inputs and outputs are handled between the systems and how the flow of data is handled with each department making sense of the input data and adding some additional information of its own to help the vehicle achieve its goals. Following is the flow chart of System architecture of Team "Junior".(4)

Also while studying their architecture, how the decision making was approached by these vehicles was also studied. For example, team Junior used transition probability for making decisions for its next set of actions based on its current location and knowledge of its surrounding. The benefit of this probabilistic view of decision making was that it penalized plans that delay the maneuver to the very last moment. For example a lane change maneuver, Junior tends to execute the lane shift at the earliest possibility while compromising the speed gains, this made their actions safer but made the vehicle slow as speed gain was penalized.

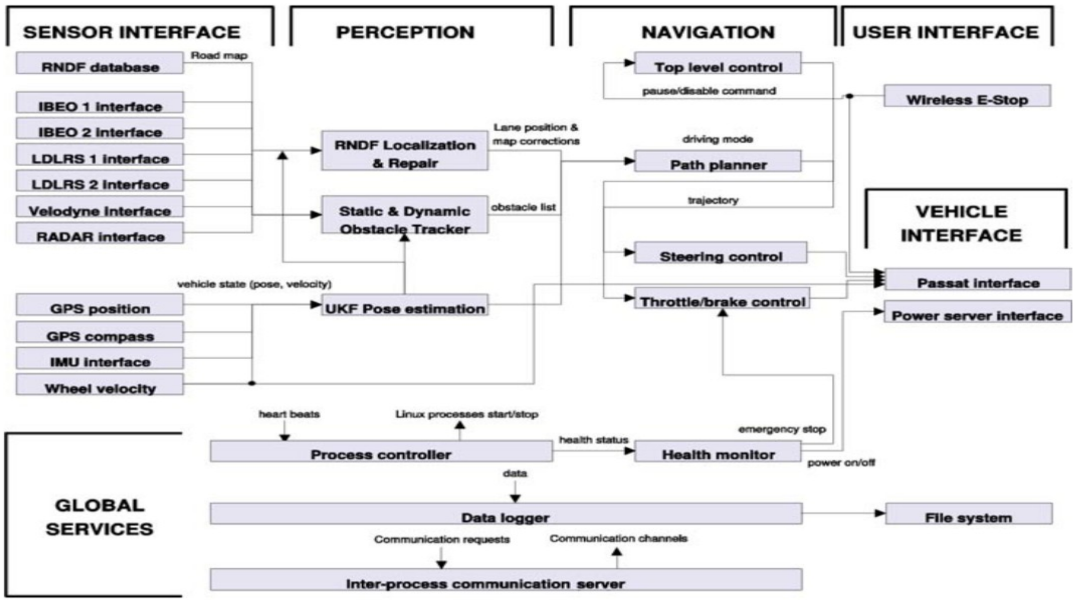

Figure 2: Flow Diagram of Junior Software (4)

This study also helped in finding out a clear differences of various planning terminologies such as path planning, maneuver planning and trajectory planning. Once differences were cleared, it was possible to now segregate Path planning section and the maneuver planning section. This also helped get a clear idea of what the input and output of the decision making part would be.(3)

2.2 Decision-Making

From DARPA Grand and Urban Challenge, a lot of research efforts have been invested for decision making. Every team which participated in Urban Challenge, has their own distinct approach for making decisions. Some of them implemented complex driving maneuvers requiring tactical decision making tasks, while others used some variant of a state machine. Team “BOSS” used analytic equations based on gaps between vehicles and used thresholding and binary decisions to switch between the tasks. While team “Junior” used a cost based approaches for global path planning and a finite state machine for making the switch between different maneuvers(4). Apart from these DARPA entries, many researchers have also tried using fuzzy logic for modeling lane change for decision making problems, this approach gave a simplicity and computationally efficient approach for decision making.

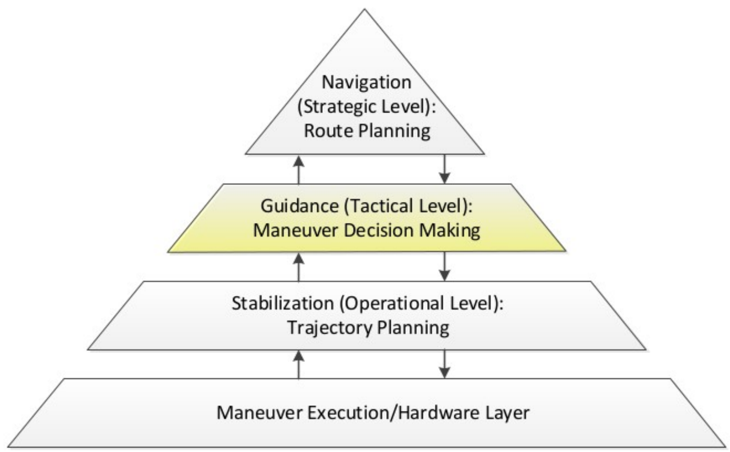

Fig. 1. Tactical decision making as a sub-problem of overall decision making for lane changes in automated driving

Figure 3: Tactical Decision making as mentioned in reference (3)

These early decision making approaches where mainly focused on some kind of cooperative behavior with the vehicles surrounding the ego vehicle. They used a set of analytic cost function for decision making in their architecture. They didn’t draw any particular attention towards uncertainties of any kind in them and were mostly based on rule base or human data driven methods. This issue was addressed by many other researches in their decision making approach. Y Gaun, S Eben Li, (2018) (5) addressed this issue by applying a probabilistic decision making Markov Decision Process for their simulation study of a highway scenario. Then a study by S. Ulbrich, M. Maurer, on Probabilistic Online POMDP approach on decision making for autonomous vehicles was also referred which took the uncertainties of perception model into consideration for lane change maneuver (3). In figure 3, we can see that how the authors sub-divided the tactical decision making as a sub-problem of overall decision making for a lane change maneuver. They have specifically used it for a lane change maneuver but this sub-division can be done in general to any kind of maneuver which a car can perform. These methods directly approached the problems of decision making been based on rule based decision approaches and not taking into consideration the uncertainties of the environment. They were based on decision process's state variables which were directly based on measured values like relative distances and velocities towards surrounding vehicles. This methods helped to keep the decision making directly based on physical quantities instead depending on per-defined rules.

The uncertainties from prediction originated from noisy perception data and the unknown intentions of human drivers cannot be measured by any system in autonomous vehicles, which is required for decision making process. For this purpose, study done by (6) addressed this problem as a Partially Observable Markov Decision Process with the intention of other vehicles as hidden variables and controlled the longitudinal acceleration of the vehicle. This gave the vehicle some time to make the decision whenever confronted with uncertain situations. Along with this situation- aware decision making was also studied. Study done by (7) in their journal considered using POMDP based algorithm which was extensively evaluated for various urban road scenarios, which includes leader follower collision avoidance and traffic negotiations at T-junction and roundabout.

These studies which were based on Markov Decision Process took our attention as a possible candidate for the decision making approach to be used in our research and a well defined base of previously conducted research work in same direction.

2.3 Control System and Vehicle Dynamics

One of the main reasons in the development of Autonomous Driving was to make it more safe and comfortable for its passengers. Controlling velocity, acceleration, steering, etc gives a much better way for avoiding collisions with other vehicles and road obstacles. Also, sudden change in vehicles speed be it due to acceleration or braking causes discomfort among the passengers of the vehicle which is also not desirable from a good ride comfort perspective. A research done by (8), which focused on shaping of the Speed Profile by the use of Model Predictive Controller. The method used by them for controlling the speed profile showed promising results by use of soft and hard constrains for suppressing excessive and sudden acceleration for preventing collision with other vehicles during a maneuver.

As we had selected overtaking scenario to test our model, we had to decide various parameters required to complete this maneuver. A study done by (9) on minimum time required for overtake problem at various driver's control input was studied. This study proposed a novel method to obtain the driver control input during the overtake maneuver while also studying the safe overtaking distance and time to consider for completing the maneuver.

While deciding over a maneuver, we must need to first check whether the collision is avoidable or not. In order to improve vehicle safety, a interaction phase between primary and secondary safety systems has been defined which according to (10) provided by the primary safety systems to achieve the objective of avoiding the collision. The authors showed a method that improves on method to calculate the Time-To-Collision to provide a more accurate result for collision avoidance system. They showed many results of TTC which have been used to distinguish whether a collision is avoidable or not and also showed that with time the value of TTC went on decreasing as the reactions of primary safety systems increased significantly. We have taken the value or TTC to be equal to 5 sec in our experiment to be the initiating point for our maneuver. As per (2), a minimum TTC value of 3.5 sec is to be considered for the non-supported drivers. We considered 1.5 sec more than the minimum as a factor of safety to our system. This can later be manipulated after considering the reactive time of the complete system Incorporated into the vehicle.

To conclude our literature review, we decided to divide the decision making task into 2 parts as Tactical decision making and maneuver planning in decision making. For this we choose to go along with the Markov Decision Process, For tactical decision making the process specified by (3) which was based on Partially Observable Markov Decision Process is planned to be used. And for maneuver planning, we decided to approach it with Markov Decision Process. For control process Model Predictive Controller along with the basic 3 degrees of freedom vehicle dynamics model was chosen to implement various limitation of vehicle dynamics to the planned path during maneuver execution stage. In following section we will have a detailed description of the system architecture used.

3. System Architecture

3.1 Global Structure

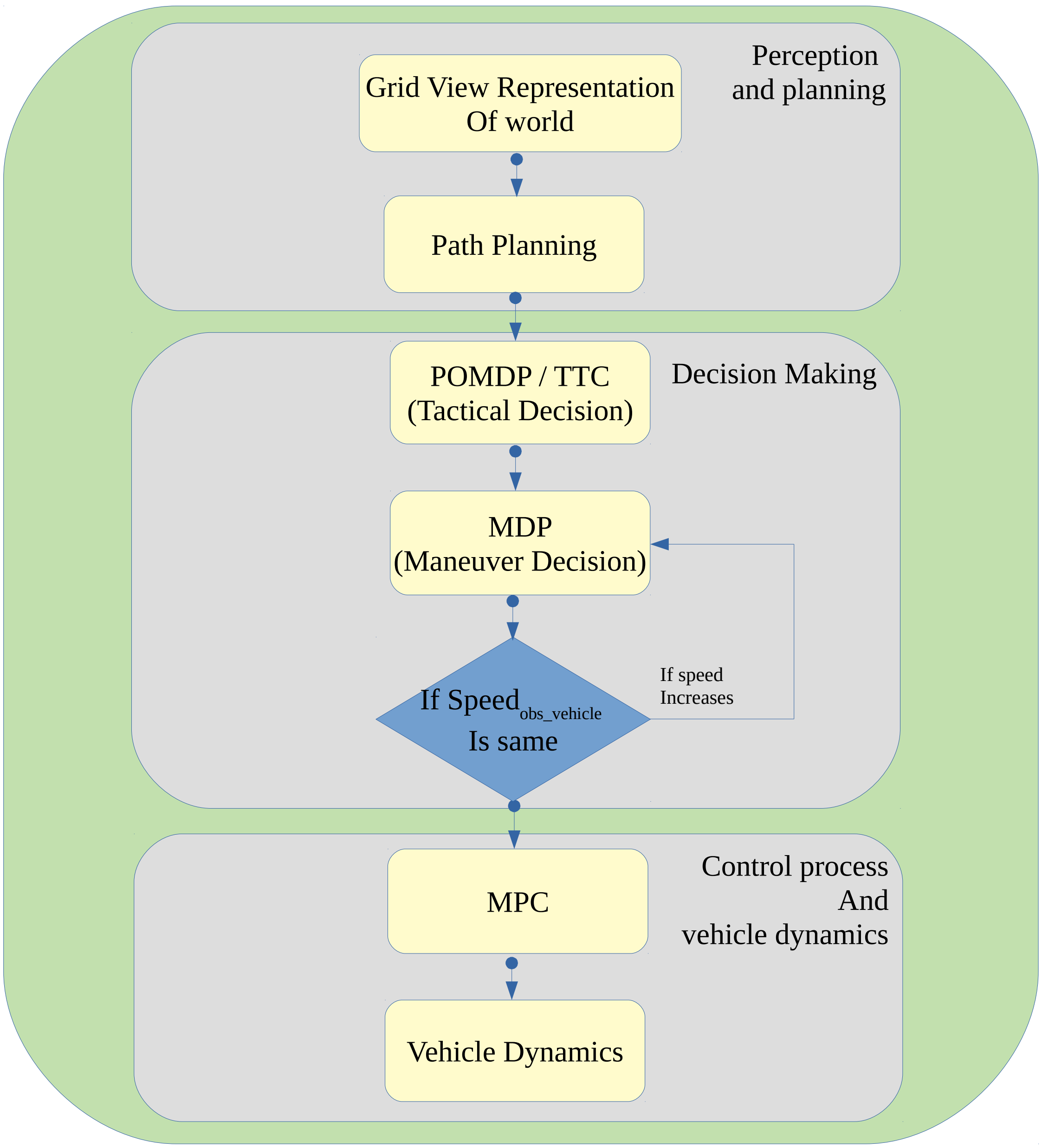

Figure 4: System Architecture block representation

This architecture can be sub-divided into 3 major parts :

1. Perception and Planning : In this block, the driving scenario to be tested is created, which depicts the data received from various perceptions sensors about the location and road environment for simulation assuming there is no noise or error in sensor data. Also, for planning a pre-planned route is given to the vehicles created in scenario which will result in a collision.
2. Decision making : This block is responsible for making the decision and creating a plan to execute so that the vehicle can avoid the collision from happening. This block is responsible for making maneuver level changes to the planned path in such a way that it ego vehicle will avoid the possible collision which has been detected by the system.
3. Control process and vehicle dynamics : This block in system architecture is responsible for constraining the defined path by the decision making block which will then be possible to execute by the vehicle taking into consideration its geometric limitations.

The working of this system starts with creation of driving scenario in a MATLAB code either by using a virtual application called Driving Scenario Designer or by directly using the inbuilt commands to create road segments such as road and lane dimensions. This mapping is done in a occupancy grid representation where in each block is of the dimension of 1*1 m. We choose occupancy grid for environment representation purpose while keeping in mind our actions which we have used in our MDP function. Once the road dimensions are specified, then ego vehicle and non-ego vehicle are added along with their characteristics such as their position in grid map, speed, and a default path to follow. This defines the initial 2 blocks of the global structure which are grid view representation of the world and path planning. While defining the driving scenario, we also define the sample time at which we would like to advance the scenario till the end of simulation.

So to summarize first 2 blocks, Grid view representation creates the environment and place the objects at their initial positions in that environment just before we try to start our simulation. Also, while defining the characteristics of ego vehicle and non-ego vehicle we have already defined a path which they must follow if there is no risk of collision between them.

Then the simulation is advanced in a while loop at the rate of sample time which is 0.1 sec till the end of simulation. Selecting a higher sampling rate usually lead to unnecessary computational overhead and choosing a low sampling rate leads to loss of data fidelity. Also, choosing a sampling time also depends on the clocking speed of the micro controller to process the given data within 2 sampling instants. A much smaller sampling time could have been better, but during some simulations with smaller sampling time, MATLAB crashed so to prevent this from happening frequently, decided to go with a much higher sampling rate. This selection was due to the limitation of hardware.

During this while loop, at each and every time step Time-To-Collision is calculated between the ego vehicle and non-ego vehicle. This condition is the first barrier of the decision making approach, that is, in our simulation this acts as the tactical level decision making. At this level the approach proposed by (3) for decision making at tactical level is what we propose to use. They have proposed a Probabilistic Online POMDP based Decision making for a lane change maneuver. In this approach they use POMDP to decide whether the lane change is possible or beneficial or not possible for the ego vehicle based on its current state and without knowing the intentions of the other slow moving vehicle. This approach address the uncertainty of other vehicles intentions. It makes these calculations based on relative distances, relative velocities and TTC with objects around the ego vehicle.

But for our experiment we didn’t implemented the POMDP method into our algorithm instead we used a simple TTC concept to trigger the decision of making the overtake maneuver. This will not let the ego vehicle execute the Markov Decision Process function (which is been discussed in detail in section 3.2.2 later in this report) until and unless the TTC is less than 5 sec. From study done by (M. Saffarzadeh, S. Naseralavi), minimum TTC to be considered for non-driver supported vehicle is 3.5 sec, so we considered an extra 1.5 sec of margin as a factor of safety in for our TTC value in our algorithm. Until TTC is not less than 5 sec, our ego vehicle will be following its normal pre-defined planned path. For calculating the TTC, we only considered the movement of vehicle in a longitudinal direction for which a simple formula is used which is as follows,

$$TTC=\frac{(non-ego\,vehicle\;position)-(ego\,vehicle\;position)}{(ego\,vehicle\;speed)-(non-ego\,vehicle\;speed)}$$

Once, the Time-To-Collision is less than the threshold, which is, 5 sec it now enters into the 2nd phase of decision making which is maneuver planning which is calculated by MDP.

For the execution of MDP function (a detailed description of MDP is presented in section 3.2.2), we require following inputs,

- Position of ego vehicle and non-ego vehicle in the occupancy grid space, this defines the initial position to start the overtake maneuver.

- Speed of ego and non-ego vehicle at that time,

- Based on their speed and position the overtaking distance is calculated while taking time as 2 times of TTC, this gives us the final goal position to reach for ego vehicle.

Based on all these parameters, MDP creates a reward map for the road which has different reward points for all the cells based on their state, that is, whether it is occupied or free, identifying the road boundaries and giving them suitable reward to prevent the vehicle going off the road boundaries. This reward map is explained in details in later section 3.2.2 in this report where sub-systems will be explained. After performing the MDP function we get a list of all the feasible points which ego-vehicle can travel through to reach the goal position. As soon as this maneuver path is defined by MDP, this path then replaces the pre-planned path so that the vehicle can go through these points as a reference rather than using the previous path which would had lead to a collision between the 2 vehicles. As soon as the pre-defined path is replaced by the new maneuver path, the variable named “Overtake” is changed to 0. This change of variable is very important as this will not let the algorithm execute the MDP function again to perform the maneuver planning because, while performing the overtaking maneuver, the TTC is going to decrease further as the gap between the 2 vehicles is going to decrease. Except for one case, at every increment of sample time, the algorithm checks for the change in the speed of the slow vehicle, If in case the speed of slow vehicle is increased, then again the “Overtake” variable is changed to 1. This scenario considers the uncertainty of the intentions of the slow moving vehicle. If during performing this overtake maneuver, ego vehicle detects increase in speed of the slow moving vehicle, this results in again planning the maneuver with new changed characteristic of both the vehicles. And finding the new goal point as set calculated by the MDP function.

Once all these maneuver waypoints, that is, waypoints refer to all the x-y coordinates which the vehicle should follow to reach the goal position, are collected, then these waypoints and the yaw angle of the ego-vehicle are given as an input to the Model Predictive Controller (see section 3.2.4). At the input the data of lateral position and yaw angles are not geometrically feasible for an vehicle to perform safely and comfortably. Prior to input the steering angle which was detected was in the limits of 45º for steering movement, which off course is not feasible under normal circumstances. For our current setup we have 1*1 configuration occupancy grid, we can also manipulate these waypoints by defining a smaller configuration occupancy grid so as to get a smaller angular steering movement between 2 waypoints decided by MDP. But as we smaller the size of occupancy grid, this will increase the computation requirement used my MDP function which is not feasible for real time working. Here to confine the steering movement of the ego vehicle within its geometric limits, MPC controller is used which follows the rule of various hard and soft constrains which we can set as per our vehicle specifications and comfort requirements. For this model we have used following constraints taking into consideration the geometric limits and considering comfort of passengers,

Steering angle – from -0.5236 to 0.5236 (rad) 30º hard constraint

Steering angle/rate – less than or equal to 0.2618 (rad) 15º hard constrain

Lateral Position – from -2 to 6 m soft constraint

Yaw Angle – from -0.2 to 0.2 (rad) soft constraint

As the inputs are passed through MPC block, these modified lateral positions and yaw angle are then given as an input to the steering input to the 3 DOF vehicle dynamics model (see section 3.2.5) which can be said as acceptable for the vehicle dynamic model to follow. The output from the Vehicle dynamics model is taken and given as an input to the simulation for getting the visual results of the total driving scenario representation.

Following is the Pseudo code representation of the complete algorithm of which explanation we just went through.

Algorithm 1 -

```
%Create Driving Scenario
1. Add road elements
2. Add ego vehicle
3.    - define speed, initial position and a pre-defined path
4. Add non-ego vehicle
5.    - define speed, initial position and a pre-defined path
6. variable Overtake = 1;
      /*Looping*/
7. while advance(scenario) for every 0.1 sec
8.    - calculate Time-to-Collision
9.    - If(TTC < 5 && TTC > 0)
10.         - If(ego-vehicle and non-ego vehicle are in same lane)
11.               - while (Overtake  == 1 )
12.                     - execute Markov Decision Process
13.                     - get the new maneuver waypoints to follow
14.                     - Overtake = 0
15.               end while
16.               - if(change in speed of non-ego vehicle is detected)
17.                     - Overtake = 1
18.               end if
19.         end if
20.  end if
21.end while
22.Input of lateral position and yaw angle for each sample step to MPC
23.Output of MPC after application of hard and soft lateral and yaw angle changes
24.Modified lateral and yaw angle changes as an input to vehicle dynamics model
25.Obtain position from vehicle dynamics output
26.Give this as an input to Driving scenario to obtain virtual results
```

3.2 Subsystems

3.2.1 Driving Scenario Application

Driving Scenario Application is a part of automated driving toolbox which is provided by MATLAB for testing simulation environment, driving algorithms, sensor modeling and synthetic data generation, etc. This application is used to create road and actor models using an interface, we can also configure vision and radar sensors mounted on the ego vehicle and use these sensors to simulate detection of actors and lane boundaries in created scenario.

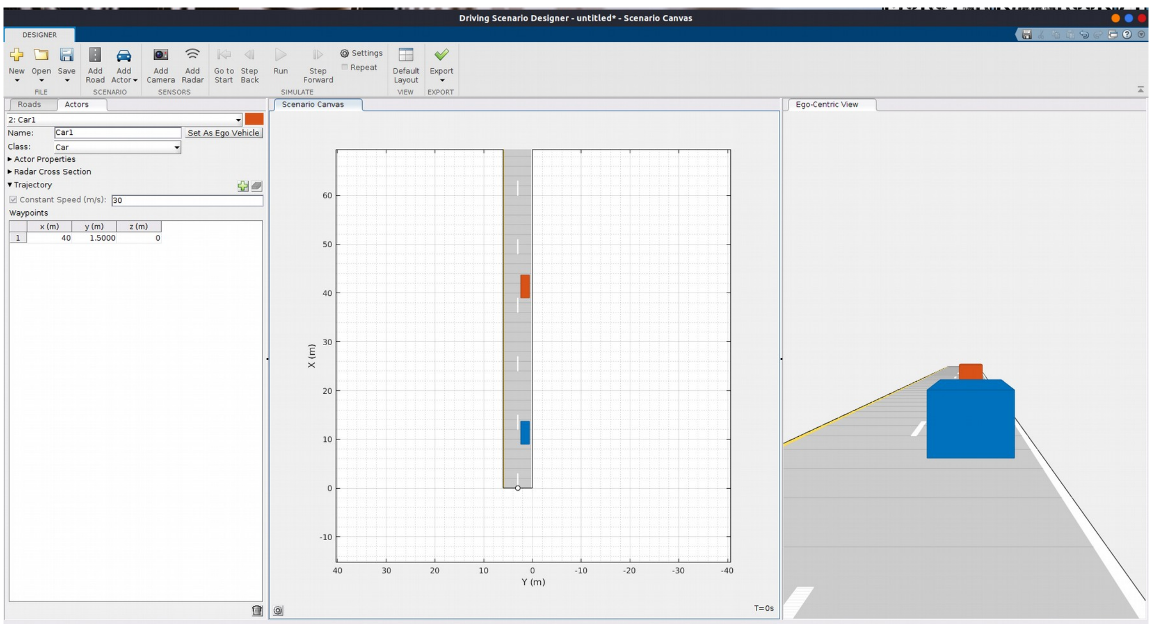

Figure 5 : Example of Driving Scenario Designer

Here in this example, we can see a road with 2 vehicles placed on their initial positions. This way we can also give them a predefined path to follow till some distance at different velocities. This way we can create any type of scenario which we want to test our algorithm for and get to know how exactly our algorithm is working for that situation.

3.2.2 MDP

Whenever we are faced with a decision of choosing an action from a set of actions and each action having its own consequences, then choosing the best action requires thinking not only about immediate effects but also to look for further consequences. Immediate effects of an action are easy to see, but long term effects are not always as easily predictable. Sometimes actions with poor immediate rewards have better long term consequences. So, choosing the best action not only based on immediate reward but also considering the future gains a Markov Decision Process is used.

There are 4 major components of an MDP model which are :

1) a set of states (S)

A state can be said as the way the world exists, and any action which we decide to take will have an effect to change the current state of the world. Every action will affect the current state differently, So if we think about the set of every possible way the world could be affected, then these all possible states of the world make a set of states in MDP.

2) a set of actions (A)

Actions are the set of possible options to make in the current state. An agent will have some limited number of actions which it can perform, the problem is to know which of these actions to take in the current state of world.

3) the effects of actions or Transition (p(s,a,s'))

Every action is going to change the current state of the world in its own way, when we decide a specific action we have an idea of how this action will affect the current state. This new changed state can be called as the transition state. Since, an action could have different effects, depending upon the state, we need to specify the action's effect for each state in MDP.

4) immediate value of actions or Immediate Rewards (r(s,a,s'))

If decision making is to be automated, then we must have some measure of action's value so that we can compare different actions transition states. We specify the immediate value of performing each action in each state.

As now we know a basic structure of MDP, we can now look for how we can solve an MDP, the solution to an MDP is called a 'policy' and it simply specifies the best action to take for each of the state. A policy π is a stochastic rule by which the agent selects actions as a function of states. The agent's objective is to maximize the amount of reward it will receive over time till it reaches a goal position. That is, finding an optimal policy π^* satisfying the following formula, where as $v_\pi(s)$ denotes the expected return from state s using policy π.

$$\Pi_* = \arg_\pi \max v_\pi(s) \tag{1}$$

For all s ϵ S.

The Bellman optimality equation (2), is a special consistency condition that the optimal value functions must satisfy and that can, in principle, be solved for the optimal value function. From this optimal value function, an optimal policy function is determined with the value iteration method.

$$\Pi_* = \arg_a \max \sum_{s',r} p(s',r|s,a)[r+\gamma v_\Pi(s')] \tag{2}$$

γ = discount factor can be any value between 0 and 1. for example, A reward R that occurs N steps in the future from the current state, is multiplied by γ^N to describe its importance to the current state. In our MDP we considered discount factor as 1.

MDP formulation for maneuver planning :

1. State space :

In our overtaking maneuver driving scenario, the state space is supposed to have a complete knowledge of all the objects in the environment along with their characteristics and properties of road. In our case, we calculate the MDP function only once, so the state space will contain all the information related to obstacle vehicle, ego vehicle, their precise location in the environment at the exact moment when the Time to collision is detected to be just below the mentioned threshold. In our example we have taken TTC = 5 sec. So, at that instant, MDP creates its own grid view map of environment with each grid size of 1*1 m^2. And place all the objects in the environment in that map at their corresponding places. This defines the state space for our MDP model.

2. Action :

As we know that during motion, a vehicle can move straight, left or right. For our model as we are going grid by grid to map a maneuver path, we also selected 3 actions for our ego vehicle, those are A = {Left, Straight, Right}. Also we know that a vehicle cannot move 90 degree left or right to its current position, So, this action space left and right are the grids which are diagonal I.e at 45 degrees to the current position grid. Below is a simple representation.

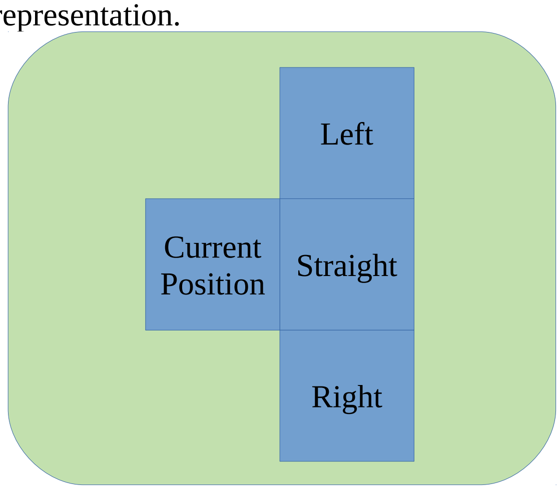

Figure 6: Action space representation

3. Transition state :

Transition model p(s, a, s') describes the probability by transition from state s to s' after performing action a. For calculating this, utility is used, here utility is defined as the sum of product of probabilities of reaching s' from s after preforming action a. So this transition state is calculated for each action possible in the current state. Once utilities of all the transition states for their respective actions are measured, the highest of them is selected and is considered the best possible action to perform at state s. This same process is performed at each an every best possible action from start position to goal position, this gives us a best possible and most rewarding path to reach our goal position.

In our algorithm described on page 33, we can see this selection of transition state is performed after assigning utility to each and every cell accessed. As we have 3 actions as described above, each of these actions (left, straight, right) are then performed at that current cell and the action which returns the most reward points is then selected for execution for that cell.

4. Reward :

The reward model r(s, a, s') is similar to that of transition model, but for rewards to be collected we first need to assign rewards to every block on the grid so that after performing the action our agent can collect it after performing the action. This rewards, can me assigned after complying it with road safety, efficiency, comfort and traffic rules which will create a complex reward system to perform with. In our model we worked on a fairly simple and basic reward system in which we determined the road boundaries, location of obstacle vehicle, giving a common reward for all the unoccupied grids, and a goal position. We set a large negative number for the grid where obstacle vehicle is detected (e.g reward = -100), and also we provided a no gain no loss reward for road boundaries to avoid ego vehicle going from off-road(e.g reward = 0). This allows ego vehicle to avoid going into that state, this ensures safety during the maneuver. A very high positive gain is given to the goal position so that ego vehicle will be tempted to be in that state(e.g reward = 1000). All the other free unoccupied grids were also given a reward. We tried 2 different reward system for this free spaces, one which gave a small gain (e.g reward = 1) and the other one was to penalized every action with a small negative gain so that this will force the vehicle to plan the maneuver in shortest possible path (e.g reward = -1).

Following is one small scale example of MDP function with smaller world grid to get an overview of how exactly it works. For this example, reward structure was as follows,

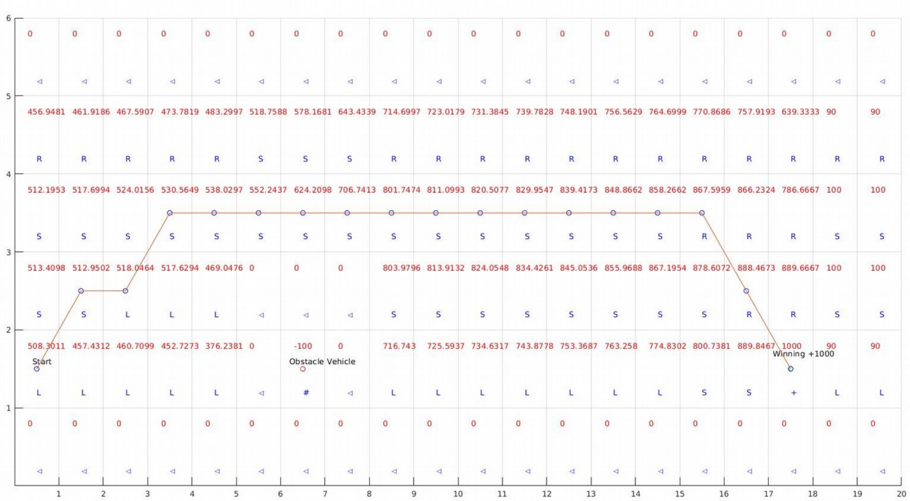

Figure 7 : MDP Reward example

In the figure 6, we can see a grid view of a small implementation of MDP function. As per in image we can see its a world of 20*6 grid, where start point is at (2,1), obstacle vehicle is placed at (7,2) which has a reward of -100 which is visible in the figure. All the numerical values which are displayed in each grid shows the cumulative reward which it gets if MDP function would had selected that option. And the connected points shown in the figure, represents the most rewarding path which the agent must take reach the goal position. Following are the reward structure which was followed,

Reward for unoccupied space = 1;

Goal position = 1000;

Obstacle vehicle position = -100;

road boundary = 0;

vehicle boundary = 0;

We have consideration the vehicle boundary because it gives the ego vehicle a safer distance to travel around the obstacle vehicle. This gave an evidence to the question that safety criteria based on distance gap between the 2 vehicles can be planned and executed in a MDP function using the reward system. This reward system can further me modified to follow traffic rules penalize the ego vehicle for traveling in wrong lane which some times becomes necessary for avoiding an collision.

Following is a pseudo code of MDP function which was implemented,

Algorithm 2 -

```
1. Define world grid in 1*1 m grids
2. Map all objects along with their rewards
3. Define road and vehicle boundaries
%loop to access each cell in created world
4. while (still accessing cells)
5.    for i = x-coordinate
6.          for j = y-coordinate
7.                update utility
8.                      performing 3 actions at each cell
9.                      Choosing best possible action and storing it
10.         end for j
11.  end for I
12. end while
13. making list of all the best actions
14. path created
15. end
```

3.2.3 POMDP

POMDP stands for Partially Observable Markov Decision Process, is a generalization of a Markov Decision Process (MDP). A POMDP models an agent's decision process in which it it is assumed that the system dynamics are determined by an MDP model, but the agent cannot directly observe the underlying state. Instead of which it maintains a probabilistic distribution over the set of possible states it can be in the environment on the bases of observations the agent receives. In our model, this method would have a good option for implementation instead of TTC as a decision making step whether to go or not to go for an Overtake maneuver. During our literature review we strumbled upon a research paper titled, "Probabilistic Online POMDP Decision Making for Lane Change in Fully Automated Driving" written by (3). In this paper they showed a decision making approach for performing a lane change maneuver in an urban environment, that is, they focused on tactical level of the decision making. A tactical decision making level can be defined as a system responsible for modifying the a-prior-planned lane level route in such a way that it fits well with the driving maneuvers of other traffic participants.

For a true system, states are not observable completely, Partially Observable Markov Decision Process helps to address this issue by the introduction of idea of a belief of being in a state x_t at time t. A POMDP is represented by the tuple (X,U,T,R,Z,O) where,

- X : set of all environment states x_t at time t.
- U : is the set of all possible actions u_t at time t.
- T : is the X*U*X $\rightarrow$ [0,1] in the transition function,
  where $T(x_t,u_{t-1},x_{t-1}) = p(x_t| u_{t-1} ,x_{t-1})$.
- R : is the X*U $\rightarrow$ R is the reward function, where r(x,u) is the reward obtained by executing action u in state x.

- Z : it is the set of all measurements or observations z_t at time t.
- O : is the X*U*X → [0,1] is the observation function,
  where $O(x_t,u_{t-1},z_{t-1}) = p(z|u,x)$ gives the probability of observing z if action u is performed and the resulting state is x.

In real time applications, POMDPs are often avoided because of their computational complexity. Significant research efforts have been spent on extending POMDP models and finding approximation methods to solve POMDPs (3).

As this section was not implemented in our model, all the details and experimental results are not available directly, but below are few important pieces of information which was mentioned in their paper by its authors S. Ulbrich, M. Maurer (3).

They provided their POMDP model with 8 states, which are as follows,

X = {(LcPossible’, LcInProgress’, LcBeneficial’),

(LcPossible, LcInProgress’, LcBeneficial’),

(LcPossible’, LcInProgress, LcBeneficial’),

(LcPossible, LcInProgress, LcBeneficial’),

(LcPossible’, LcInProgress, LcBeneficial),

(LcPossible, LcInProgress’, LcBeneficial),

(LcPossible’, LcInProgress, LcBeneficial),

(LcPossible, LcInProgress, LcBeneficial)}

LcPossible is a binary state variable which describes whether a lane change is possible or not, LcInProgress is a binary state variable which describes whether the agent is in lane change process or not and LcBeneficial is a binary state which describes whether a lane change is beneficial or not.

For these states, They have modeled a 3 actions set which the model can execute,

U = {‘Drive’, ‘InitiateLaneChange’, ‘AbortLaneChange’}

And for the reward function of POMDP, all elements of the reward matrix are set to zero except for the following one,

r(u = InitiateLaneChange, x = (:,LcInProgress’,:)) = -100;

r(u = InitiateLaneChange, x = (:,LcInProgress,:)) = -10000;

r(u = AbortLaneChange, x = (:,LcInProgress’,:)) = -10000;

r(u = AbortLaneChange, x = (:,LcInProgress,:)) = -200;

r(u = Drive, x = (:,LcInProgress’, LcBenefecial’)) = +5;

r(u = Drive, x = (:,LcInProgress’, LcBenefecial)) = -5;

r(u = Drive, x = (LcPossible,LcInProgress’, LcBenefecial’)) = -5;

r(u = Drive, x = (LcPossible,LcInProgress’, :)) = -60;

r(u = Drive, x = (LcPossible,LcInProgress, LcBenefecial)) = +50;

Every “:” denotes all the possibilities of that state in state space X.

The Authors finally concluded that their approach scaled remarkably well towards human like decisions for lane change scenarios which they tested. For testing their system they have had created a decision alert system through which they were able to come to this conclusion. From this results we can conclude that this approach would be a best fit for our 1^{st} tactical level decision where we actually used TTC based approach. This POMDP based approach will further enhance the overall decision making capability of our system.

3.2.4 Model Predictive Controller

To understand the use of Model Predictive Controller in our systems, first lets understand what in general MPC is, A MPC is an advanced method of process control that has been in use in the process industry for a very long time. Model Predictive Controller relies on the dynamic model of the process most often linear empirical models obtained by the system identification. MPC possesses many attributes that make it a successful approach to as a control design such as, Simplicity, Practicality, etc. The plus point of using a MPC over any other controller is its capability to handle large number if manipulated can controller variables, Constraints imposed on these variables and time delay. MPC models predict the change in the dependent variables of the modeled system that will be caused by changes in its immediate independent variables. Following figure shows the block diagram representation of a basic MPC structure on how MPC interacts with the plant model.

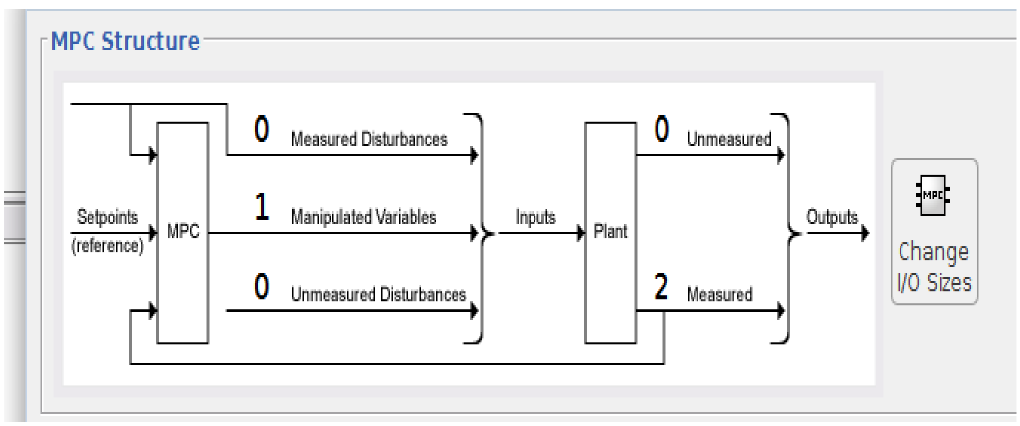

Figure 8: MPC Structure basic Input output

Following is the block diagram of Model Predictive Control and vehicle dynamics structure implemented in algorithm.

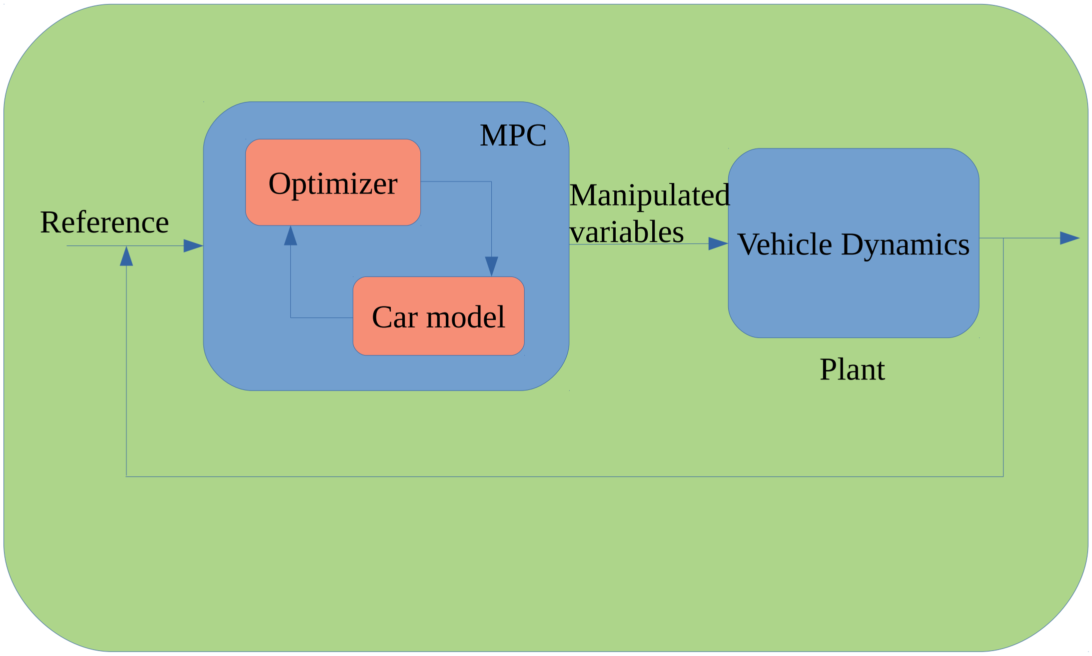

Figure 9: Block Diagram representation of MPC

Till now we had deal with all the path planning, decision making and maneuver planning aspects of an Autonomous vehicle. As we moved towards implementation of these calculated path, we first of all need to look for is whether these points are physically possible for the vehicle to execute the maneuver or not. Also taking into consideration the comfort and safety during to the physical limitations of the vehicle. Different maneuvers have different results depending on the speed of the vehicle at the time of execution. This mostly affects the comfort related problem for passengers, but sometimes may result in losing control of the vehicle if any action’s execution pushes the vehicle beyond its physical limitations. So the constrain these calculated maneuver waypoints as per vehicle’s physical limitations is much needed for safe and comfortable ride.

For this purpose we decided to use the Model Predictive Controller. MPC is said to be an advanced method of process control that is used to control a process while satisfying a set of constrains in the form of hard and soft constrains. MPC controller has been in use in various other industries to control the processes. In Automotive, we use it to control certain inputs which we have to give to plant, that is, the vehicle dynamics model of our vehicle, while following various constraints. MPC uses an inbuilt model to predict the plants behavior and an optimizer which ensures that the predicted future plant output tracks the desired reference.

We used an Adaptive MPC in our model to control the steering inputs given to our vehicle dynamics block because a traditional MPC controller is not effective at handling the varying dynamics, as it uses a constant internal plant model. For this, we gave an input of Lateral positions planned by our MDP maneuver planning and the yaw angle to MPC. We used an MPC designer which is an interactive tool provided by MATLAB as a part of Model Predictive Control Toolbox. First we had to specify all the parameters in the MPC toolbox such as number of inputs and outputs along with sample time and prediction and control horizons, constrains and weights. Following are the details which we used to set the parameters and tuned the MPC controller,

Sample Time : 0.1 seconds

Prediction Horizon : 10 secs

Control Horizon : 3 secs

Steering angle – from -0.5236 to 0.5236 (rad) 30º hard constrain

Steering angle/rate – less than or equal to 0.2618 (rad) 15º hard constrain

Lateral Position – from -2 to 6 m soft constrain

Yaw Angle – from -0.2 to 0.2 (rad) soft constrain

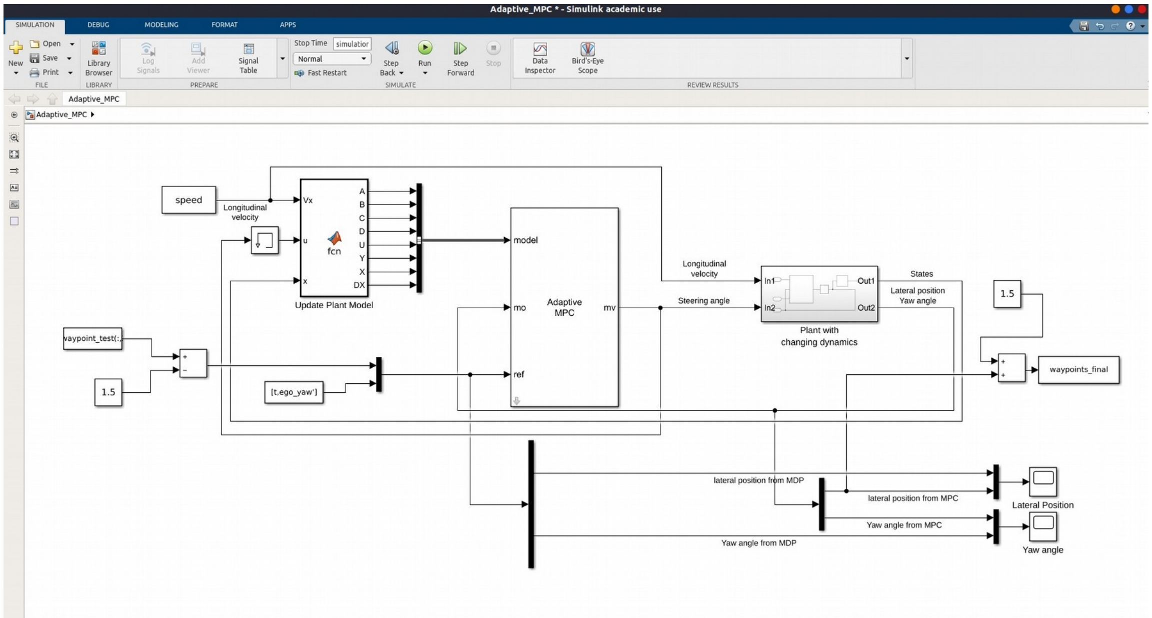

Figure 10: Model Predictive Controller Block diagram

In the above figure 7, we can see the connections made to and from Adaptive MPC controller with the plant model. And following are the control signal results from a test run for lateral position and yaw angle.

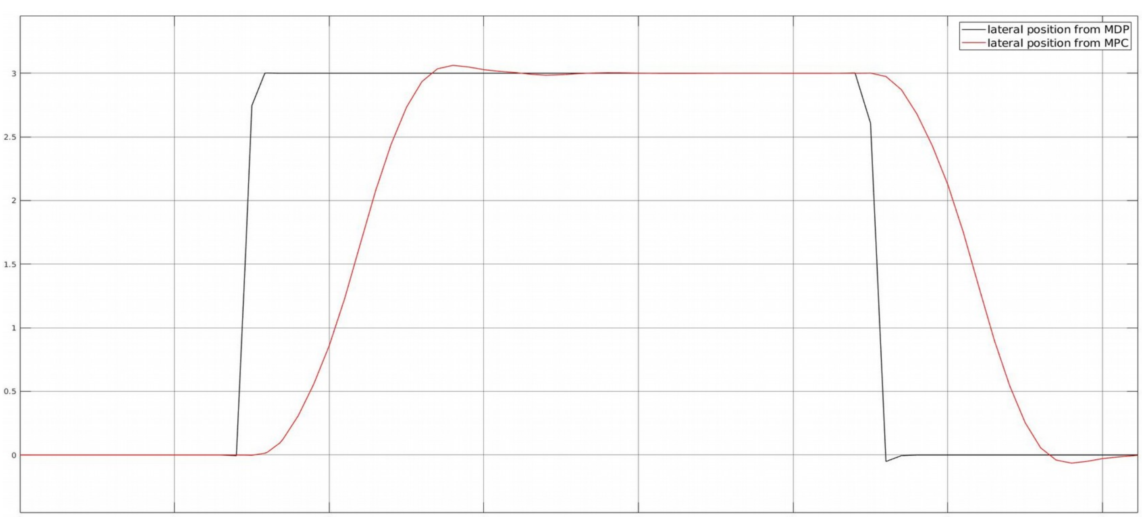

Figure 11: Lateral Position before and after MPC

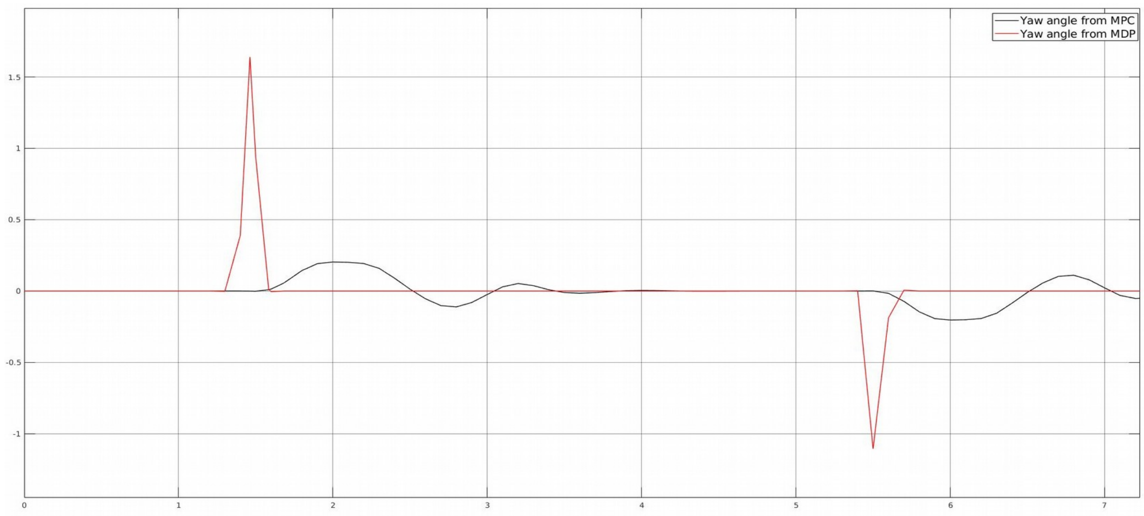

Figure 12: Yaw angle before and after MPC

3.2.5 Vehicle Dynamics

We used the state space continuous-time model which represented the lateral vehicle dynamics in them, with inputs of steering angle and longitudinal velocity and output as lateral position and yaw angle along with the states which is then used as input to the "update plant model" block for the Adaptive MPC controller. This "updated plant model" takes states as input from the plant model and operates at each time step to update the MPC model with the current states of plant model as MPC requires a discrete plant model. The "update plant model" first calculates the state space matrices as from the inputs, and then computes the discrete model and updates the nominal conditions according to current operating conditions. These nominal conditions are then given as input to the Adaptive MPC controller. Following are the vehicle parameters which we took into consideration,

Signification	Symbol	Value	Unit
Vehicle Mass	Mass	1575	Kg
Vehicle yaw inertia	Iz	2875	$kg.m^2$
Distance of COG – front axle	Lf	1.2	m
Distance of COG – rear axle	Lr	1.6	m
Cornering stiffness of front tires	Cf	19000	N/rad
Cornering stiffness of rear tires	Cr	33000	N/rad

Table 2: Vehicle Parameters

Equations used in Plant model are as follows,

% Continuous-time model

Ac = [$\frac{-2*Cf+2*Cr}{m.Vx}$, 0 , $\frac{-Vx-(2*Cf-2*Cr*lr)}{m*Vx}$, 0;

0, 0, 1, 0;

$\frac{-(2*Cf*lf-2*Cr*lr)}{Iz.Vx}$, 0 , $\frac{-(2*Cf*lf^2+2*Cr*lr^2)}{Iz*Vx}$, 0;

1, Vx, 0, 0];

Bc = [$\frac{2*Cf}{m}$, 0, $\frac{2*Cf*lf}{Iz}$, 0]’;

Cc = [0 , 0, 0 , 1;

0, 1, 0, 0];

Dc = zeros(2,1);

xdot = A*x + B*u;

The Difference between “plant model” and the “Updated plant model” is that the nominal conditions need to be calculated in discrete time model which is different from plant model.

Following are the equations used for converting continuous-time model to discrete-time model,

```
% Generate discrete-time model

nx = size(A,1);

nu = size(B,2);

M = expm([[A B]*Ts; zeros(nu,nx+nu)]);

A = M(1:nx, 1:nx);

B = M(1:nx,nx+1:nx+nu);

C = Cc;

D = Dc;

%Nominal condition for discrete-time plant

X = x;

U = u;

Y = C*x + D*u;

DX = A*x + B*u-x;
```

3.3 Results

We have divided this results section in 2 parts, first section discusses various types of MDP reward functions which we tried and the second part discusses the final model in which we can see how if we the use of MPC smooths the maneuver process as compared to directing using the MDP defined waypoints. For results related to POMDP, we were not able to perform it within the duration of internship, but, as results found performed and discussed by (3) in their journal, decision making by using POMDP approach gives results which were much similar to that of decision’s made by an human operator. If interested in more details on results its recommendation to refer to their journal to get a much clear idea.

3.3.1 Different types of reward matrix which was tried in MDP :

Test 1:

- Reward at free space = 1;
- goal position = 100;
- obstacle position = -100;
- road boundary = 0;
- vehicle boundary = 0;

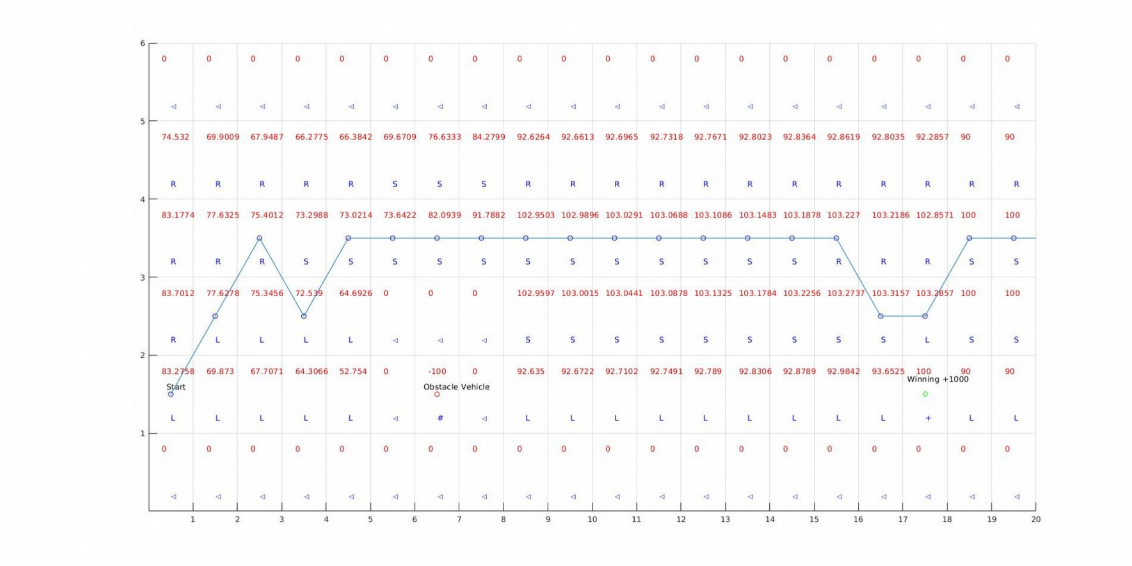

Figure 13: Reward test 1

In this test, agent missed to reach the goal position. This happened due to the fact that, by the time the agent reached near goal position, it has already collected more reward points than the reward available at goal position. This resulted in missing the goal position and going out of contest. This issue can be solved by giving reward at goal position more generously.

Test 2 :

- Reward at free space = 1;
- goal position = 1000;
- obstacle position = -100;
- road boundary = 0;
- vehicle boundary = 0;

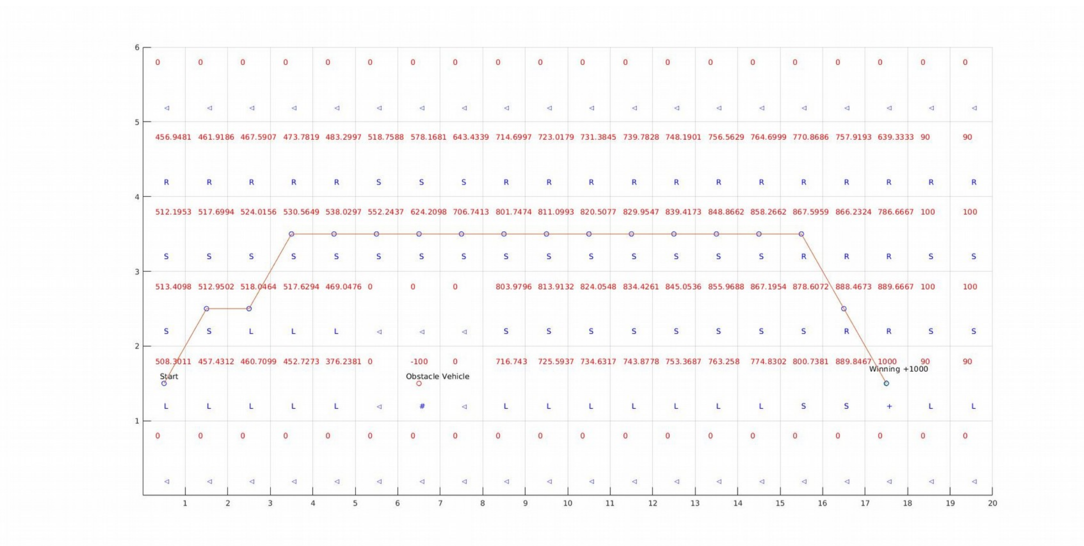

Figure 14: Reward test 2

Here we can see as we increased the goal reward, agent was successfully able to reach the goal position. Also we can see that due to vehicle boarder, agent is able to move at a safe distance from the obstacle vehicle.

Test 3 :

- Reward at free space = -1;
- goal position = 100;
- obstacle position = -100;
- road boundary = 0;
- vehicle boundary = 0;

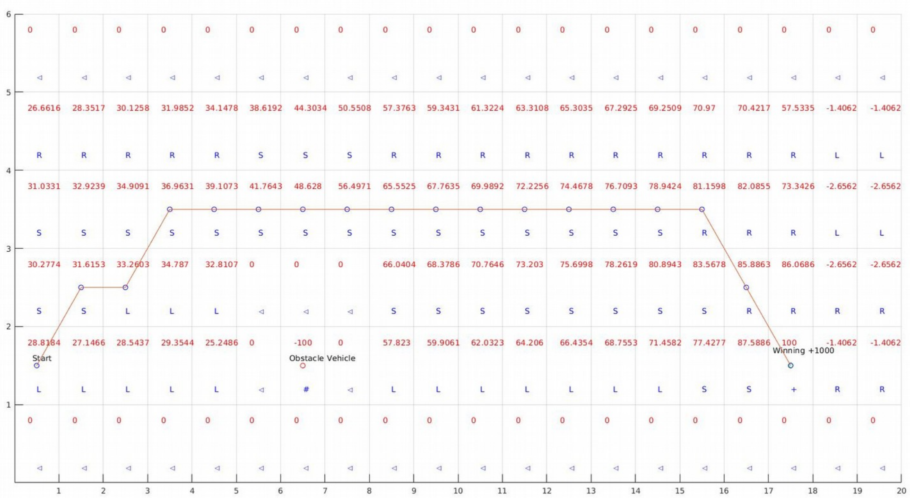

Figure 15: Reward test 3

This test is similar to test 1, instead it has a small change, the reward given to free space or unoccupied cells is given as -1. We can see that there is no major change in the path taken by the agent to reach the goal position. Agent is successfully able to reach the goal position because it has not collected more reward by the time it reaches the goal position.

Test 4 :

- Reward at free space = -1;
- goal position = 1000;
- obstacle position = -100;
- road boundary = no road boundary;
- vehicle boundary = 0;

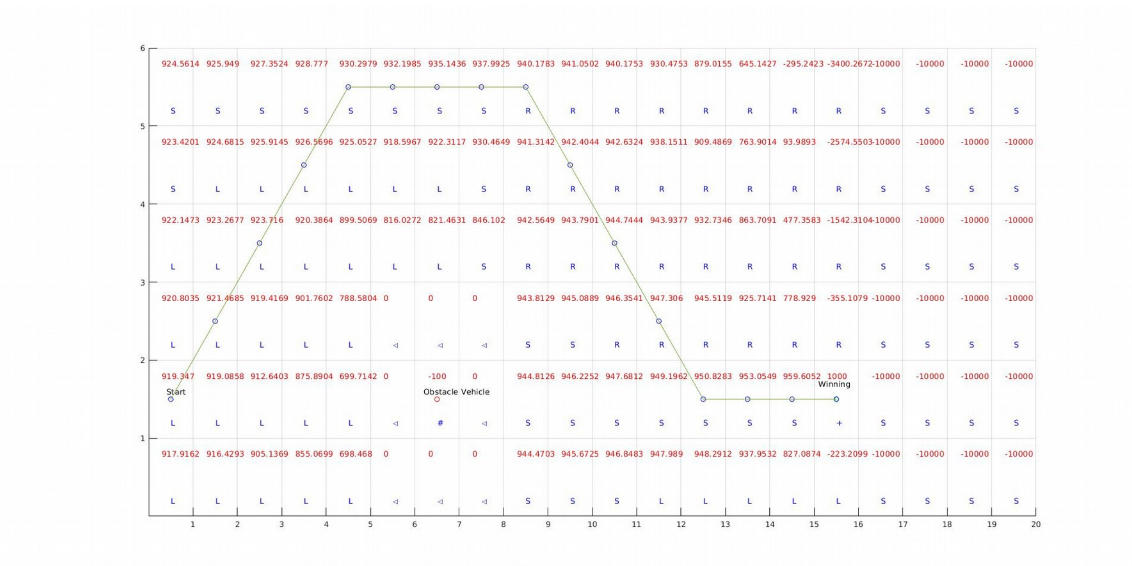

Figure 16: Reward test 4

This test shows that, how MDP functions tries to avoid obstacle or no reward positions in the environment, this condition forced agent to go at the road boundary. Till now in 3 test we had road boundaries marked a 0 which forced agent to stay within the road boundaries, mostly on the center line as it was equidistant from both the boundaries, or we can say no reward region.

3.3.2 Final Model results :

In above section, we saw different test cases which were done to find a good set of reward parameters to be assigned to our MDP function. In this section we will apply this MDP function with a fix set of rewards parameters and tried to carry out a simulation to see how exactly it reacts to created scenario. Following is the result of MDP function been applied to overtake scenario.

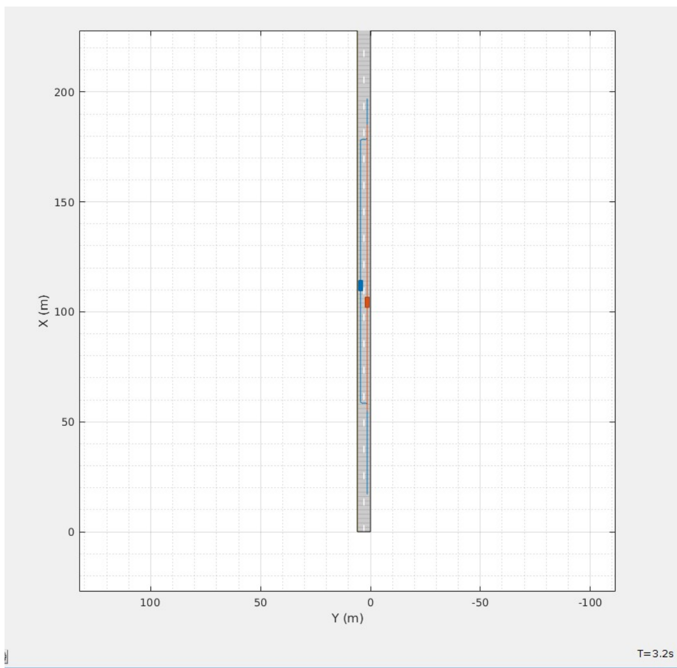

Figure 17: MDP pathway

Here we can see the results of a MDP function defined path, from this path we can clearly see that during lane change maneuver, the waypoints are places in such a way that it is almost expecting a 90º turn from the vehicle which is not possible by any physical vehicle to perform. Due to this MPC controller helps to steer the vehicle in a controlled and physically possible maneuver. Following is the output of same scenario, but results after MPC controller model.

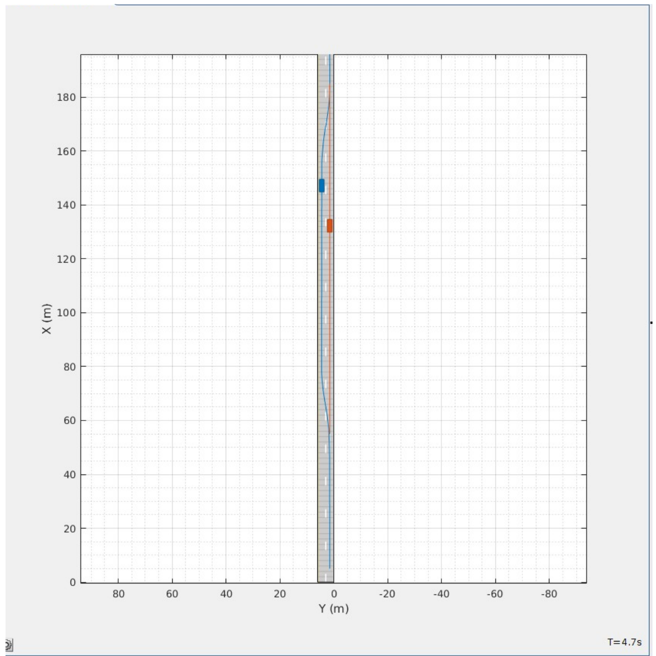

Figure 18: MPC pathway

Here in the MPC pathway we can see a clear difference in the path of the ego vehicle, the maneuver path is much more smoother than the one defined by MDP function. I have compiled the video results of these simulations in a video for your reference. (video link - https://youtu.be/bRG1PzwcVsE)

To see this difference following is the graph plot of lateral position and yaw angle for the 2 scenarios of MDP and MPC outputs,

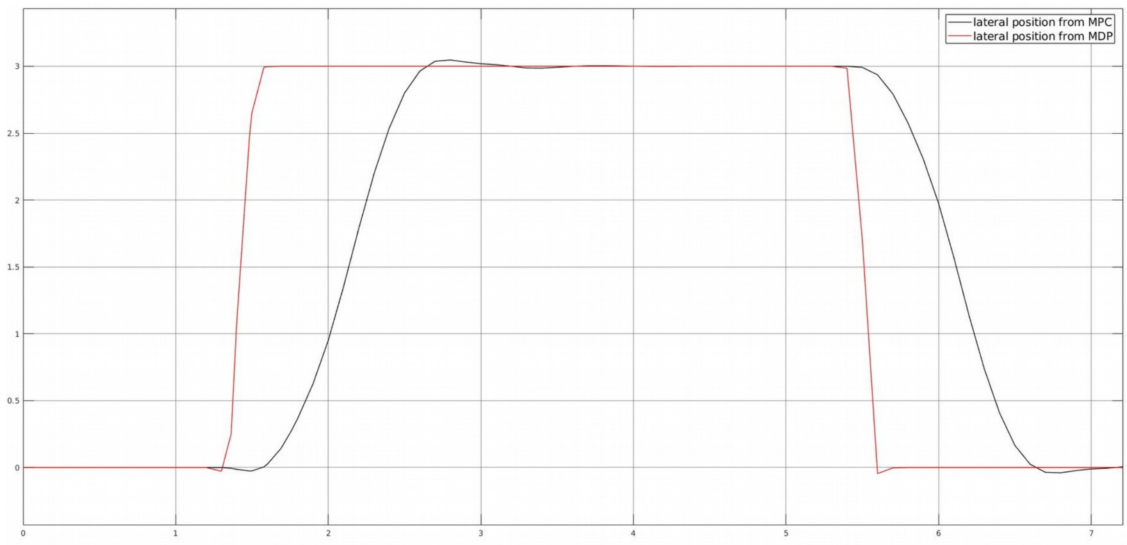

Figure 19: Lateral Position

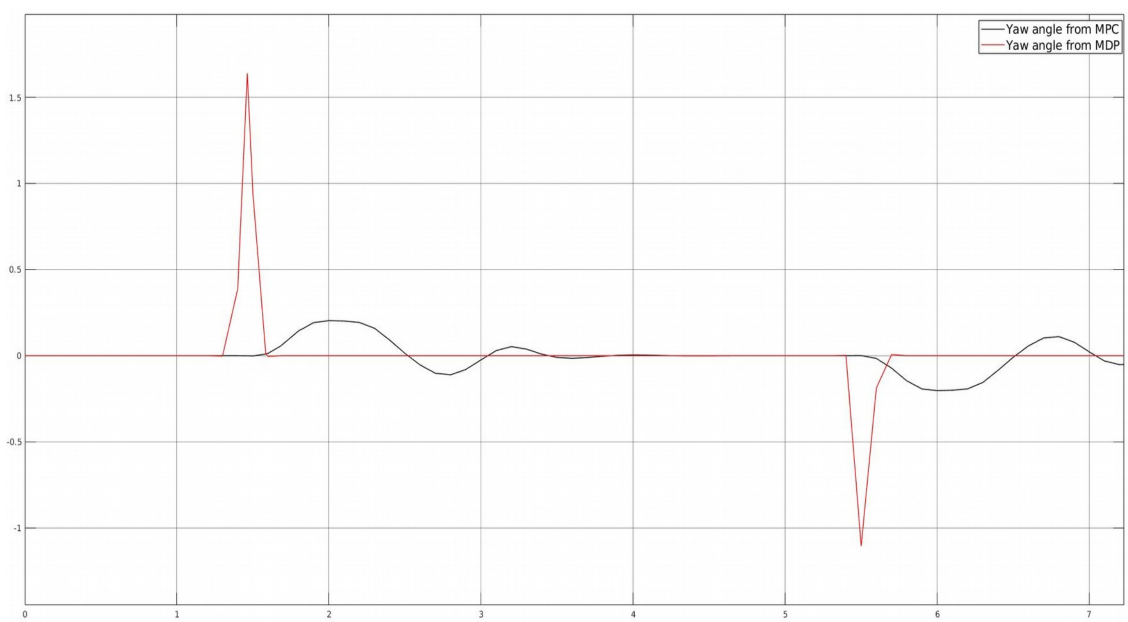

Figure 20: Yaw angle

Here we can see a remarkable improvement in the yaw angle between the MDP and MPC controller outputs. This is the result of hard and soft constrains applied in MPC controller to control the steering angle of vehicle dynamics model. Here the yaw angle represented with black is the one which will be given to simulator to execute the motion of the vehicle. The yaw angle represented with red is like a raw input created my MDP function without having much vehicle dynamic constrains.

4. Conclusion and Prospects

4.1 Summary

Before concluding this thesis report, we will have a small summary of the material which has been presented in the thesis. Then, we present the conclusions on which we arrived followed by the potential future perspective of future work.

In section 1, we introduced the our objectives and motivation to work on the given topic. Our main goal was to work on the decision making aspect of the autonomous vehicle. We first of all decided to clear the objectives and tried to have a clear focus on which areas to focus on to achieve our objectives.

Once decided with this part we moved to section 2, where we reviewed multiple documents stating various methods currently being used in academic and industrial research. We distinguished various methods which meet our objectives. As we were proceeding further we went on selecting one method for each step as per requirement for our architecture. From literature review we narrowed our focus on MDP/POMDP for decision making, Model Predictive Controller for processing control.

Once, all the components were developed, it was time to integrate these components into architecture with inputs and outputs as required by system. During building the architecture, each component was thoroughly studied and tried to manipulate to get the most out of that method being used. Various different combination were tried in MDP function for its reward matrix which gave varying results for each small change made to them.

Integrating these components was a challenging task as may time the outputs of one system didn’t provided the data in the manner which was acceptable to other components. But after investigating the issue and trying various different methods, we were finally successful to implement these components into a global running structure.

4.2 Conclusion

During the development of this thesis, we can conclude that :

- In this report, we tried to present a method using Markov Decision Process to define a maneuver path for the ego vehicle which need to modify the current path which it is following to avoid colliding with other vehicles or to gain some time to reach the destination.
- The results show that MDP can be used as an effective maneuver planning function which we showed by implementing it for an highway overtake scenario, but as we have defined a general set of actions for MDP, it can be used for various different scenarios.
- MDP/POMDP are an excellent decision making process which can take into consideration the varies problems associated with uncertainties be it with uncertainty in perception or the unpredictable behavior of other road occupants around the ego vehicles.

4.3 Prospects

To conclude this thesis, we propose some prospects which could help improve this model and extend the work :

- Reward function in MDP can further be modified to include traffic based reward so that it may penalize the ego vehicle to break the traffic rules such as driving on wrong side of the road.
- The horizon for the MDP function can be set to a specific limit which then can be computed for each sample time.
- Implementation of POMDP for in tactical decision for the consideration of environmental uncertainties.

Bibliography

1. https://www.nhtsa.gov/press-releases/usdot-releases-2016-fatal-traffic-crash-data, published on October 2017. (cit. On pg. 4)
2. M. Saffarzadeh, S. Naseralavi, "A general formulation for time to collision safety indicator" , ResearchGate. (2013) (cit. On pg. 10)
3. S. Ulbrich, M. Maurer, "Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving" , IEEE. (2013) (cit. On pg. 10, 14, 15, 16, 29, 30, 44)
4. M Buehler, K. Iagnemma, S. Singh, "The DARPA Urban Challenge" , Springer. (2009) (cit. On pg. 13, 15)
5. Y Guan, S Eben Li, "Markov probabilistic decision making ofself-driving cars in highway with random trafficflow: a simulation study" , Intelligent and connected vehicles. (2018) (cit. On pg. 15)
6. C. Hubmann, M. Becker, D. Althoff, D. Lenz, C. Stiller, "Decision Making for Autonomous Driving considering Interaction and Uncertain Prediction of Surrounding Vehicles" , IEEE. (2017) (cit. On pg. 16)
7. W. Liu, S. Kim, S. Pendleton, M. Ang, "Situation-aware Decision Making for Autonomous Driving on Urban Road using Online POMDP" , ResearchGate. (2015) (cit. On pg. 16)
8. Y. Mizushima, I. Okawa, K. Nonaka, "Model Predictive Controller for Autonomous Vehicles with Speed Profile Shaping" , ScienceDirect. (2019) (cit. On pg. 17)
9. Y. Zhao, X. Zhang, "Minimum Time Overtaking Problem of Vehicle Handling Inverse Dynamics Based on Two Kinds of Safe Distances" , chinese journal of mechanical engineering. (2018) (cit. On pg. 17)
10. F. Jimenez, J. Naranjo, F. Garcia, "An Improved Method to Calculate the Time-to-Collision of Two Vehicles" , Springer. (2013) (cit. On pg. 17)
11. M. Saffarzadeh, S. Naseralavi, "A general formulation for time to collision safety indicator" , ResearchGate. (2013) (cit. On pg. 10)
12. L. Adouane, Reactive versus cognitive vehicle navigation based on optimal local and global PELC*. Robotics and Autonomous Systems (RAS), (2017).
13. L. Adouane, HYPERLINK "https://www.crcpress.com/Autonomous-Vehicle-Navigation-From-Behavioral-to-Hybrid-Multi-controller/Adouane/9781498715584" Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures, Book.
14. A. Chebly, R. Talj et A. Charara, “Coupled longitudinal/lateral controllers for autonomous vehicles navigation, with experimental validation”, Control Engineering Practice, Elsevier, (2019).
15. A. Chebly, R. Talj, A. Charara, “Maneuver planning for autonomous vehicles, with clothoid tentacles for local trajectory planning”, IEEE 20th International Conference on Intelligent Transportation ITSC 2017, Yokohama, Japan, (2017).

16. D. Iberraken, L. Adouane, and D. Denis, "Multi-level Bayesian decision-making for safe and flexible autonomous navigation in highway environment", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'18), Madrid-Spain, (2018).
17. D. Iberraken, L. Adouane, and D. Denis, "Safe Autonomous Overtaking Maneuver based on Inter-Vehicular Distance Prediction and Multi-Level Bayesian Decision-Making", IEEE International Conference on Intelligent Transportation Systems (ITSC'18), Maui-Hawaii-USA (2018).
18. H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun et F. Guillemard, “Evidential based approach for trajectory planning with tentacles, for autonomous vehicles”, IEEE Transactions on Intelligent Transportation Systems, (2019).
19. H. Mouhagir, R. Talj, V. Cherfaoui, F. Guillemard, F. Aioun, “A Markov Decision Process-based approach for trajectory planning with clothoid tentacles”, IEEE Intelligent Vehicles Symposium (IV), Götenborg, Sweden(2016).
20. C. Philippe, L. Adouane, A. Tsourdos, H-S. Shin and B. Thuilot, Probability Collectives Algorithm applied to Decentralized Intersection Coordination for Autonomous Vehicles, 30th IEEE Intelligent Vehicles Symposium (IV’19), Paris-France, (2019).
21. C. Philippe, L. Adouane, B. Thuilot, A. Tsourdos and H-S. Shin, Safe and Online MPC for Managing Safety and Comfort of Autonomous Vehicles in Urban Environment, IEEE International Conference on Intelligent Transportation Systems (ITSC'18), Maui-Hawaii-USA, (2018).
22. Schubert, R. Evaluating the utility of driving: Toward automated decision making under uncertainty. IEEE Transactions on Intelligent Transportation Systems, (2012).
23. G. Tagne, R. Talj, A. Charara, “Design and comparison of Robust Nonlinear Controllers for the Lateral Dynamics of Intelligent Vehicles”, IEEE Transactions on Intelligent Transportation Systems, (2016).
24. G. Tagne, R. Talj, A. Charara, “Design and Validation of a Robust Immersion and Invariance Controller for the Lateral Dynamics of Intelligent Vehicles”, Control Engineering Practice, Elsevier, (2015).