CRITICAL ZONE CALCULATION FOR AUTOMATED VEHICLES USING MODEL PREDICTIVE CONTROL

by

Enimini Theresa Glasky

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Indianapolis, Indiana

May 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lingxi Li, Chair

Department of Electrical and Computer Engineering

Dr. Yaobin Chen

Department of Electrical and Computer Engineering

Dr. Brian King

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

To all the little girls around the world who have ever doubted their ability to achieve greatness.

ACKNOWLEDGMENTS

I'd like to thank my husband Christopher Glasky, for his endless support and for bringing the light back into my world. Thank you for being the best teammate anyone could ask for. To Dr. Lingxi Li, thank you for your guidance, wisdom, support, and constant encouragement. Thank you for consistently believing in me, teaching me, and guiding me on this path. Thank you for always cheering me on even when I struggled to see the finish line. This would not have been possible without you. Dr. Yaobin Chen, thank you for your wealth of wisdom and knowledge. Dr. King and Sherrie, thank you for always answering all my questions and guiding me on my next steps, I appreciate all your help.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

EV Ego Vehicle

TTC Time To Collision

TTR Time To React

MPC Model Predictive Control

SMC Sliding Mode Controller

FLC Fuzzy Logic Controller

CAS Collision Avoidance System

ADAS Advanced Driver Assistance System

ADS Automated Driving System

MMPC Multi-constrained Model Predictive Control

CPU Central Processing Unit

LTV Linear Time-Varying

DOF Degree of Freedom

MIMO Multi-Input Multi-Output

QP Quadratic Programming

SISO Single-Input Single-Output

ZOH Zero-Order Hold

MV Manipulated Variables

OV Output Variables

TSCC Lines of Time-Scaled Collision Cone

ABSTRACT

This thesis studies critical zones of automated vehicles. The goal is for the automated vehicle to complete a car-following or lane change maneuver without collision. For instance, the automated vehicle should be able to indicate its interest in changing lanes and plan how the maneuver will occur by using model predictive control theory, in addition to the autonomous vehicle toolbox in Matlab. A test bench (that includes a scenario creator, motion logic and planner, sensors, and radars) is created and used to calculate the parameters of a critical zone. After a trajectory has been planned, the automated vehicle then attempts the car following or lane change while constantly ensuring its safety to continue on this path. If at any point, the lead vehicle brakes or a trailing vehicle accelerates, the automated vehicle makes the decision to either brake, accelerate, or abandon the lane change.

1. INTRODUCTION

1.1 Thesis Background

An automated vehicle is a vehicle that is designed to function without the presence of a human driver. The design of automated vehicles has been noted to have begun in the 1920s and the development of the prototypes started in the 1980s as shown in Fig. 1.1. The technology behind automated vehicles was inspired by science fiction when automobile manufacturers wanted road systems with navigation systems that would be used to control vehicles. In 1939, the Futurama exhibit which was planned by Norman Bel Geddes imagined driver-less cars controlled on electrically charged roads.

Up until 1980, all proposals for automated vehicles included the modification of roads with radio, magnetic, or electrical control systems. In the 1980s, automobile manufacturers worked with universities and started the design of self-navigating automated vehicles, which use cameras, radar, and ultrasonic sensors. Research has shown that automated vehicles can reduce traffic accidents, provide economic and societal benefits, and improve traffic congestion and delays, as the underlying systems of these vehicles are less prone to distractions [1]. Automated vehicles also enhance driving comfort, optimize fuel consumption, decrease gas emissions, and improve driver's performance [2].

Fig. 1.2 expands on how there is vast research on various models about how automated vehicles can perform tasks such as lane change, car following, collision avoidance, adaptive cruise control, cooperative adaptive cruise control, lane-centering control, etc. while capitalizing on ensuring the safety of the driver and occupants.

Years	Features
1950-2000	Safety/convenience features Cruise control Seat belts Antilock brakes
2000-2010	Advanced safety features Electronic stability control Blind spot detection Forward collision warning Lane departure warning
2010 - 2016	Advanced driver assistance features Rearview video systems Automatic emergency braking Pedestrian automatic emergency braking Rear automatic emergency braking Rear cross traffic alert Lane centering assist
2016 - 2025	Partially automated safety features Lane keeping assist Adaptive cruise control Traffic jam assist Self- Park
2025- future	Fully automated features Highway autopilot

Figure 1.1. Features of Vehicles.

Levels of Automation	Definition
Level 0 – No Automation	Human driver is entirely in control
Level 1 – Driver Assistance	ADAS can assist human driver with steering, braking / accelerating
Level 2 – Partial Automation	ADAS can solely control steering, braking / accelerating
Level 3 – Conditional Automation	ADS can perform all aspects of driving alone under certain circumstances
Level 4 – High Automation	ADS can perform all aspects of driving and monitor driving environment
Level 5 – Full Automation	ADS can perform all driving tasks in all circumstances alone

Figure 1.2. Levels of Automation.

With the rapid development of automated driving technology, there has been an increase in the development of advanced systems that can perform various tasks that involve longitudinal and lateral motion. This thesis focuses on the study of car-following and lane-changing maneuvers of automated vehicles.

1.2 Literature Review

There are different theories that have been used to study vehicle following and lane changes in automated vehicles, such as model predictive control, game theory, behavior models, and learning models. An ego vehicle (also an automated vehicle) should always have the ability to plan its trajectory, show its intention to make a lane change, and execute/abort its plan based on the responses from surrounding vehicles [3].

Lane changes are complex maneuvers that include both longitudinal and lateral motions. It also includes sub-maneuvers such as lane changing, lane keeping, and merging in a sequential manner. A safe lane change not only depends on the ego automated vehicle, but also relies on external factors such as road conditions, weather, relative velocity, relative distance, and many more [4].

For the design of Advanced Driver Assistance Systems (ADAS) and automated Vehicles, safety is one of the most important factors to consider. In the modeling of intelligent vehicles, safety and risk seem to go hand in hand. Risk can be thought of as the possible scenarios

where a vehicle could be damaged. To safely design vehicles, different mathematical models are used to predict possible scenarios [5]. There are mainly three types of models:

Physics-based motion models

Physics-based motion models (Fig. 1.3) assume that the motion of vehicles only depends on the laws of physics. It assumes a constant speed and orientation for vehicles. Dynamic and kinematic models are used to predict future motion with the use of control inputs like steering, acceleration, braking, and properties of the vehicle.

For road safety, physics-based motion models are mostly used for trajectory prediction and collision risk estimation. Evolution models which include dynamic models are based on Lagrange's equations and kinematic models that describe the motion of a vehicle using the relationship between parameters of movements such as position, velocity, and acceleration while excluding forces that affect motion. There are different methods for using these models for trajectory prediction, such as single trajectory simulation, Gaussian noise simulation, and Monte Carlo simulation.

Maneuver-based models

Maneuver-based models study the relationship between the future motion of a vehicle and the intended maneuver of the driver. They treat vehicles as independent maneuvering entities. A vehicle's motion on a road network is assumed to correspond to an independently executed series of maneuvers. Fig. 1.4 shows a Maneuver based motion model

Interaction-aware motion models

Interaction-aware motion models consider the inter-dependencies between vehicle maneuvers, which leads to a more accurate description of the motion profile, a better understanding of the traffic scenario, and more detailed risk analysis. These models in Fig. 1.5 are based on prototype trajectories or Dynamic Bayesian Networks (DBN). Prototype trajectories is a good method, however, it does not consider inter-vehicle influences during the learning

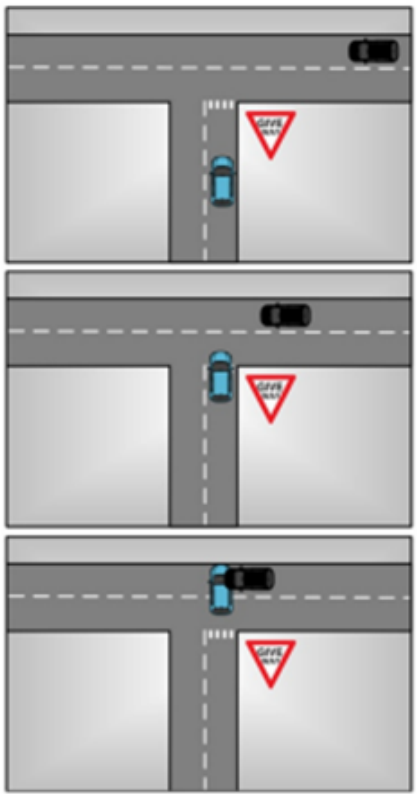

Figure 1.3. Physics-based motion model [5].

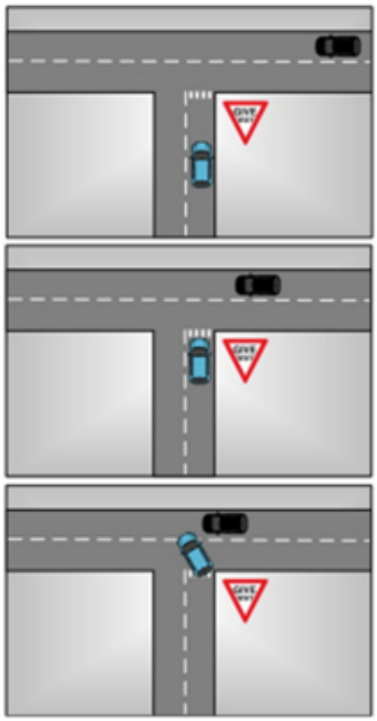

Figure 1.4. Maneuver-based motion model [5].

phase as the number of motion patterns becomes too complicated to manage. The majority of interaction-aware motion models are based on Dynamic Bayesian Networks (DBN).

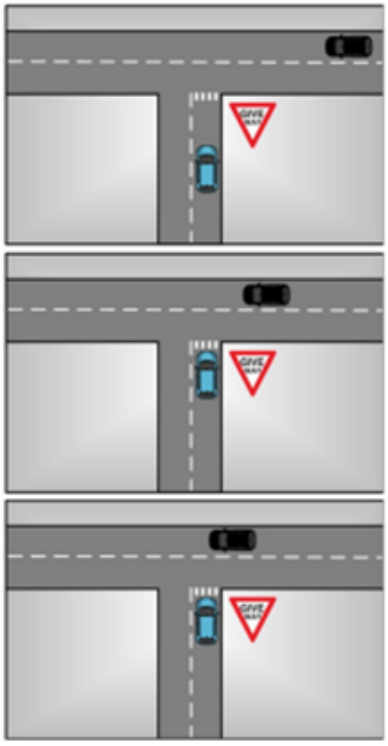

Figure 1.5. Interaction-aware motion model [5].

Another important aspect of motion planning for automated vehicles is collision prediction. if a possible collision can be predicted, the intended maneuver for an automated vehicle can be carefully planned to ensure it is avoided. This is done by first predicting the potential future path for all objects in the scenario, then detecting collisions between each pair of trajectories to determine the risk estimate based on the total possibility of a collision. Collision prediction can either be binary, where trajectories are discretized and a search for the possibility of a collision is performed at each time stamp; or probabilistic, where the risk is calculated in a probabilistic setting over possible future trajectories and detecting whether or not there is the presence of a collision. Other risk factors that complement collision risk exist as well. The impact of a possible collision could be influenced by certain indicators such as the velocity of the vehicles [6], the probability that both vehicles will be in the same

position at once, and the configuration of the collision [7]. They could be measured as a response to time such as Time-To-Collision (TTC) and Time-To-React (TTR).

Model Predictive Control Method

Lane change maneuvers in automated vehicles include decision-making and path planning while ensuring that sufficient inter-vehicular spacing is being maintained to avoid collision with other surrounding vehicles. The amount of space between vehicles can be used to determine when automated vehicles make changes by modeling critical zones. With the use of kinematic motion models to calculate the critical time gap, automated vehicles can decide whether to brake or steer. In order to model a critical zone; linear motion models are used to model the motion of all vehicles involved, the critical zone is calculated based on different scenarios at every time instant. The ego vehicle uses longitudinal and lateral acceleration to plan evasive maneuvers. The vehicle motion is modeled using a point mass model and is discretized using zero-order hold. Leading vehicles are always assumed to travel at a constant velocity while having the ability to come to an immediate stop at any time. Trailing vehicles are assumed to travel at a constant velocity until the ego vehicle signals its intention to begin a maneuver. Trailing vehicles are assumed to have three responses: decelerate to allow the ego vehicle to make a change, continue at its initial speed, or accelerate to inhibit the lane change. The motion planning which involves getting the vehicle from its initial position to the final desired position is written as a standard quadratic programming optimization problem, constraints such as speed are set to ensure the entire maneuver is within the limits of the critical zone [3].

Instead of critical zones, a hierarchical control system with an inner loop and outer loop can be used. The inner loop is an adaptive cruise controller which ensures the ego vehicle maintains a relative distance from the leading vehicle while maintaining its speed. The outer loop decides which preceding vehicle the automated vehicle will follow to minimize the cost function and the optimal amount of space between vehicles that will allow the ego vehicle to initiate a lane change. The driving decision entropy of drivers in the main lane is the cost function. The driving behavior model is also used to predict the decision of drivers in the

target lane. A driver can either accelerate thereby rejecting the intention of the ego vehicle, accept the cut in action, or remain undecided [8].

There are different driving scenarios that exist on highways. When it comes to lane changes, there are mandatory lane changes (Fig. 1.6), that occur when a lane is ending, and the driver has no option but to merge onto the closest lane. For simplicity, there is one vehicle in the merging lane and at least two in the target lane.

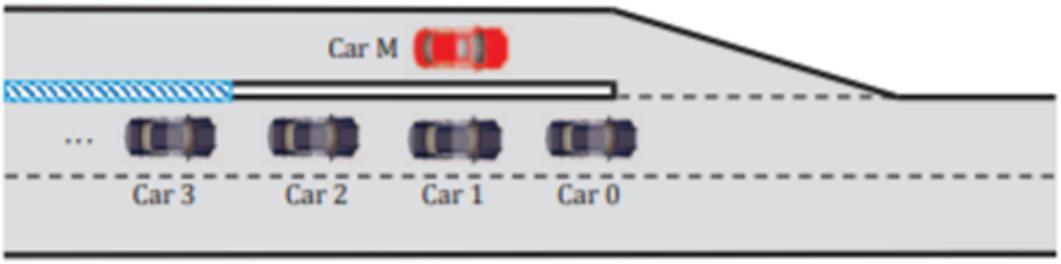

Figure 1.6. Mandatory lane change scenario [8].

During lane changes on highways, one of the biggest concerns is the harmonization between the ego vehicle and surrounding human drivers in a mixed traffic situation. The probability of the driver in main lane accepting the lane change can be quantitatively modeled and analyzed using a logistic regression model and a Bayesian network. In this thesis, the road has two straight main lanes and one merging lane which connects to the left lane of the main road.

One approach is to use a controller based on Randomized Model Predictive Control (RMPC), where at every time step various speed profiles are generated for the ego vehicle. These speed profiles are used to predict potential trajectories for both the automated vehicle and surrounding vehicles in the main lane. The stochastic decision model for decision-making was used to define the decision entropy of drivers, it outputs a "1" or "0" to show if the merging decision has been accepted or rejected.

To find the optimal speed for merging, MPC is formulated as a nonlinear optimization problem. Two types of models are used, one for the decision of the human driver and the

other for whether the car accelerates or decelerates. The acceptance model works based on the logistic regression model; it estimates the state of acceptance. The motion model is based on a switched PD acceleration control law for the ego vehicle. The ego vehicle uses the PD control to keep a safe distance from the leading vehicle. The reference distances are set according to the average distance of the leading vehicle that has been collected when the state of acceptance is reject and undecided. The gains, k_p and k_d are set by trial and error. RMPC can be used when obtaining real-time solutions become an issue, it is promising when it comes to solving nonlinear and stochastic MPC problems. The Monte Carlo-based forward prediction of state trajectories with a sampling of random inputs is the highlight of RMPC [9]. [8] solves the same traffic scenario but uses speed profiles to determine optimal speed. It uses driver behavior models and two control loops to determine optimal space between vehicles. It focuses on reducing the computational time by using a Hamilton – Jacobi equation to solve the control problem.

Adaptive Cruise Control is one of the features of automated vehicles. The goal is to maintain the longitudinal velocity of a vehicle as well as maintain its relative distance to a leading vehicle. MPC is one of the most common approaches for designing an ACC algorithm because of its ability to handle various constraints. In [2], a controller that combines analytical optimal and predictive control theory is used. In order to get an explicit and low-cost function, analytical optimal control methods are applied although they are sensitive to modeling errors and uncertainties. A model-free RMPC can also be used to get an optimal solution but it involves tedious computations. The combination of both methods is proposed to design a nonlinear optimal controller for the ACC. The analytical solution uses the integral of tracking error to generate various input values while the RMPC selects the true optimal value. The ACC controller takes in the position of the target vehicle and its velocity then determines the appropriate position and velocity for the ego vehicle while satisfying constraints.

For lane changes to occur, the automated vehicle needs to be able to plan a path for how it will complete the task before initiating a maneuver. MPC with terminal set constraints can be used for this. [10] uses situational awareness and a trajectory framework planning, it combines potential field-like functions and reachability sets of a vehicle to iden-

tify safe zones that the automated vehicle can use. Lane changes can be treated as moving obstacle maneuver avoidance problems where RRTs are used but this approach led to large jerk which caused discomfort to the driver, MPC can be used to improve this due to its ability to formulate vehicle dynamics where constraints are set as finite horizon constrained optimization problems to prevent the collision. Collision avoidance constraints are usually non-convex, this results in a limitation of the feasibility and uniqueness of the solution. To solve this, researchers used convexification (a receding horizon control for automated vehicles in dynamic environments), change of the reference frame [11], and approximation. A mathematical framework that includes a target generation block, a trajectory generation block, and an artificial potential field is used.

The potential field maps the surrounding vehicles based on obstacle's position, orientation, and relative velocity. The target generation block searches for the safest point on the road which is compatible with dynamics of the surrounding vehicles and computes the reference state set. It combines the safe zones in the potential field with the capability of the vehicle dynamics of the surrounding vehicles derived from the reachable set. MPC is used to generate possible trajectories to steer the vehicle to the target state. [12] uses this approach to solve a reference tracking problem, it guarantees a closed-loop stability and persistent feasibility of the optimization problem. In addition to a coordinate frame, a vehicle frame (V-frame) is used, which is in the center of gravity of surrounding vehicles and uses Roll-Pitch-Yaw (RPY) convention. Obstacle frame (O-frame), which is in the center of gravity of the leading vehicle, and road frame (R-frame), which is the projection of the origin of v-frame onto the rightmost edge of the road with x-axis in the direction of travel, are used. This method has the advantage of being able to steer the state of a constrained systems toward any set-point regardless of whether it was in the terminal set or not and if an admissible state was targeted. It steers the system to the closest admissible state. With the increase of in consumers purchasing more self-driving cars, there is an increase in blended traffic situations involving human-driven and automated vehicles. The current safety requirements for self-driving cars make it overly conservative which makes it difficult to negotiate with human drivers in merge scenarios. In order to overcome this, game theory can be added to MPC to improve the control performance.

Due to the fact that there is a wide range of various decisions that can be made by different drivers in the same driving scenario, a controller that uses experimental validation and game theoretic learning can be used. An aggressiveness estimation algorithm is proposed, which is based on the construction of a heuristic utility function for human drivers in order to match different driving behaviors by varying the driver's aggressiveness. Previous studies of driver's aggressiveness are typically offline surveys which measures aggressiveness by frequency of candidate' aggressiveness behavior such horn honking, tailgating, etc. The results from this research are related to candidate's information such as age, gender, etc. However, processing such information online for automated vehicles is difficult. Some research shows the influence of driver's aggressiveness in a simulation environment, the behavior of each driver is different depending on whether they are classified as conservative, neutral, or aggressive. In [13] and [14], aggressive drivers operate at a higher speed, accelerate faster, maintain smaller following distance when following another vehicle and they have higher risk of collision when changing lanes. Driver's aggressiveness in this case is a parameter used to describe the driving utility including headway distance and vehicle speed.

This approach uses an online aggressiveness estimation algorithm that uses a driver's real-time behavior. A mandatory lane change (MLC) is used here as there is limited time to make a decision and adjust vehicle's speed and position before the end of the merging area is reached. Research attempt to study MLC has shown that there is an assumption that a longitudinal controller is separated from the lane change process while the lane change strategy is developed in a rule-based manner. In Game Theory, vehicles are considered participants of a game, and traffic rules are typically considered to be implicit in different decision models. [15] describes it as a two-person non-zero-sum game which can be described as an interpretation of Hidas' driver courtesy scheme where vehicles share the payoffs or heuristics of the lane-changing process. GTMPC controls surrounding vehicle's longitudinal acceleration and lane decision and consists of a high-level and low-level controller. The high-level controller is the aggressiveness estimator and multiple games, where it observes the response of a GCV to the SV, then approximates its aggression. A Stackelberg game is then established between both vehicles. The high-level controller simulates the motion of both vehicles to determine the benefits and costs of all possible strategies according to

the Stackelberg algorithm. The solution of the game is to find the vehicles with the highest payoff. This selection becomes an input to the low-level controller. The low-level controller uses this result along with the IDM model to predict the motion of the trailing vehicle which is used as a reference for the MPC. The MPC outputs the optimal longitudinal acceleration and the optimal amount of time to initiate a lane change. At the high level, MPC is used to predict the potential trajectory of both players which is used to create the payoff of each player for each strategy combination of each game. In the low level, it controls the motion of the surrounding vehicle and tries to initiate the lane change in front of the trailing vehicle. In summary, it executes the lane change, merges into traffic, maintains its lane, and stays in the center of the lane while maintaining its speed. [16] uses a four-stage hybrid model predictive controller. This low-level controller regulates the longitudinal position and lane decision of the SV. It focuses on using MPC to handle constraints in different lanes while also generating a target gap. The low controller divides the lane change process into four stages: traveling in the current lane (TCL), leaving the current lane (LCL), crossing the lane marking (CLM), and traveling in the target lane (TTL). The purpose of the four stages is to identify what constraints the MPC deals with depending on what lane the SV is in. In TCL or LCL, SV only considers constraints associated with the current lane. In CLM, SV considers constraints in both lanes. Finally, in TTL, only constraints related to the target lane are considered.

For automated vehicles to be commercialized, the safety of the system has to be guaranteed. This requires a safe and robust algorithm with the ability to plan trajectories while responding appropriately in normal and high-risk driving scenarios. For automated vehicles to be able to handle various situations, objectives and constraints need to be set for trajectory planning that is based on numerical optimization. [10] creates an algorithm that plans trajectories that take into consideration miscellaneously shaped critical zones where automated vehicles can avoid collisions. Instead of using time gaps, it allows the critical zones to be defined by the longitudinal and lateral motion of the automated vehicle. To successfully navigate traffic situations, EV must do it in a self-assertive manner and not excessively conservative. The algorithm for path planning includes the determination of possible lateral motion trajectories of the maneuver, determination of the longitudinal safety

critical zones that correspond to the lateral trajectories, and the longitudinal motion. The lateral motion of the EV for determining the Scenario Model Predictive Control (SMPC) is another technique, the algorithm presents a safer way for detecting lane changes on highways and motion planning. Over the past decade, there has been an increasing number of studies in Advanced Driver Assistance Systems (ADAS) due to new sensor technology and a wider range of capabilities of computational hardware.

ADAS provides support to drivers by aiding in collision avoidance, lane changes, and others. ADAS can also estimate and predict traffic scenarios by measuring the vehicle's environment through laser scanners, radars, cameras, ultrasonic sensors, and vehicle-to-vehicle communication. Other approaches that use Physics-based models do not take into account the driver's control actions as they only apply laws of motion and friction limits, this in turn makes the accuracy of results only up to the human's reaction time. For longer prediction times, a maneuver-based model is used and for a higher level of realism, interaction-aware models are recommended. MPC is extremely useful for proper control of multi-variable systems with constraints on inputs and states. It also allows the integration of forward information and constraints. Deterministic Model Predictive Control (DMPC) which is based on deterministic traffic predictions can be used for path-following models with constraints in scenarios where there are no other vehicles. Stochastic MPC is a more advanced approach that applies probabilistic traffic prediction to MPC using stochastic uncertainty. Its key feature is the inclusion of chance constraints. These allow a systematic tradeoff between control performance and probability of state constraints in a stochastic setting [17].

1.3 Thesis Contributions

This thesis is an expansion of [3], which calculates and models critical zones to determine the optimal space for an ego vehicle to perform a car following and a lane change. It applies the theory to different traffic scenarios. The paper [3] only covers one scenario. This thesis expands the theory and technology to different scenarios. It introduces a lane following scenario, a lane change scenario, and then a second lane change scenario where the lead vehicle brakes. These scenarios were modeled and simulated in Matlab using the autonomous

driving, Simulink, and MPC toolboxes. The results of the simulation experiments validated the performance of the algorithm.

1.4 Thesis Organization

Chapter 1 focuses on the literature review. The literature review explores the various approaches that MPC can be applied to the lane change maneuver. Every method has an advantage or shortcoming while some methods are designed to be an improvement of a previous theory such as RMPC and GTMPC. This thesis focuses on the use of critical zones to determine the optimal time for an ego vehicle to begin a car following or lane change.

In Chapter 2, the problem is formulated. The development of critical zones is explained based on if it's a leading or trailing vehicle. It also highlights the assumptions that are used to determine the critical time gap that the ego vehicle uses to plan its trajectory. This chapter also discusses acceleration profiles and how they are used to generate equations for calculating critical zones. The critical time gap varies depending on if it is a leading or a trailing vehicle.

Chapter 3 discusses the motion planner and controllers used in the thesis. It introduces different scenarios to be simulated, the test benches used for simulation, and related parameters.

Chapter 4 discusses the results from the simulation experiments and summarizes the thesis.

2. PROBLEM FORMULATION

2.1 Critical Zone Modeling

This section introduces the problem and proposes the solution by defining the calculation of critical zones. An automated vehicle has to maintain its lane while following a lead a vehicle without entering the critical zone of another vehicle. When it has the intention of performing a lane change, it has to be able to plan a and execute one while paying attention to the behavior of other vehicles on the road. A critical zone is an area around a vehicle where the autonomous vehicle can either brake or steer to reduce the chances of a collision. A kinematic model that calculates the critical time gap needed to safely maneuver within the limits of the critical zone is used. To safely steer, the critical zone depends on the relative lateral position between vehicles. The main goal of the automated vehicle is to successfully complete a maneuver from its original lane to another lane. In this study, critical zones are depicted as shaded regions around each vehicle, to prevent a collision, the EV is never allowed to be in any of the critical zones throughout the maneuver. To calculate critical zones, a time gap is used as the safety indicator based on the capacity of a vehicle to brake to a stop.

The amount of time it takes for a vehicle to reach a reference position is defined as the time gap between a vehicle and the reference position. Reducing margins of critical zones makes it easier for an automated vehicle to complete a lane change in dense traffic but there still needs to be sufficient margin that allows for evasive action in worst case scenarios. An ego vehicle is expected to have the ability to plan the maneuver, signal to other vehicles to show its intention and complete or abandon the lane change based on the response of surrounding vehicles. This gives the EV a lot more freedom while planning its trajectory in an efficient and self assertive way [10].

The goal is to determine the critical zone around surrounding vehicles of importance, seek desired final position, and plan the motion to reach the position without collision. The relative lateral or offset and relative velocity between an ego vehicle and a surrounding vehicle is used to calculate whether the EV should brake or steer to avoid a collision. The following assumptions are used to model a critical zone: All motion is based on linear motion models.

For the ego vehicle, a point mass model established by kinematic relations and the triple integral of jerk defines displacement. The motion of surrounding vehicles is represented using a constant velocity model. Assumed worst case scenario is used to calculate the critical zone. Longitudinal and lateral acceleration profiles that are based on the specifications for comfort are used by the ego vehicle to plan evasive maneuver.

The equations used to calculate the critical zones depend on if the vehicle of interest known as the target vehicle is a leading or trailing vehicle. A leading vehicle is the target vehicle in front of the ego vehicle and a trailing vehicle is behind a leading vehicle. If it is a leading vehicle, the ego vehicle plans a maneuver under the assumption that the ego vehicle could brake at any time. If it is a trailing vehicle, the driver could either decelerate to accept the lane, accelerate to reject the lane change, or remain indecisive. The behavior of drivers in other vehicles are always of interest as it determines how the ego vehicle should plan its motion [16], [18]. For the purpose of this thesis, the assumption is that the trailing vehicle always accepts the intention of the ego vehicle to initiate a lane change.

2.2 Acceleration Profiles

The longitudinal and lateral motion of the vehicle determine the shape and size of the critical zones. The longitudinal and lateral acceleration profiles are used to derive equations for the critical zones. A longitudinal model explains the relationship between the acceleration of a vehicle and the different forces impacting the motion of the vehicle lengthwise [19]. The lateral profile describes the relationship between the position of a vehicle and the forces influences its motion horizontally. Longitudinal profiles are considered in reference to the velocity of the vehicle while the lateral profile is for the steering behavior. The best steering or braking maneuver is used to decide on which longitudinal and lateral profiles should be used. For the braking profiles, longitudinal jerk is considered as constant until the maximum deceleration is achieved and maintained for some time. In the main setup, the scenario in Fig. 2.1 was considered.

The leading vehicles are labeled as L and the trailing vehicles are labeled as T. This shows the relationship between the EV and other target vehicles as well as the critical zones.

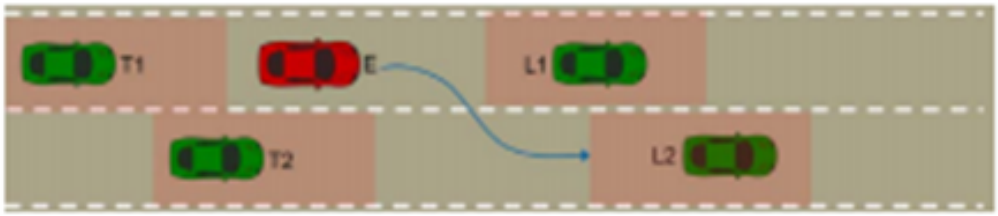

Figure 2.1. Lane change scenario [3].

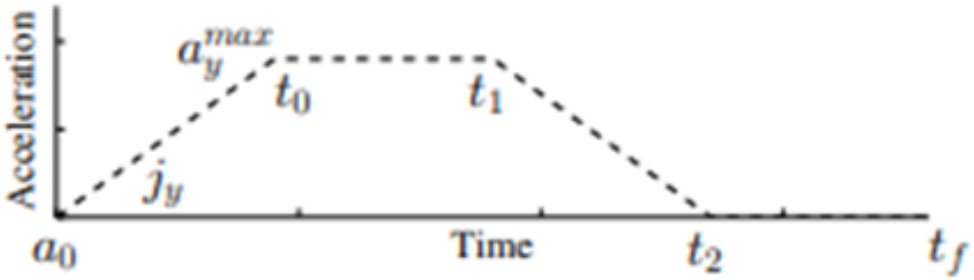

Figure 2.2. Longitudinal acceleration profile for braking [3].

The longitudinal acceleration model in Fig. 2.2 depicts how a vehicle begins with an acceleration a_0 and then at time t_0, it accelerates to a_max. It stays at that value of acceleration until t_1, then it begins to decelerate at t_2 as it slows down to stop for a total travel time of t_f.

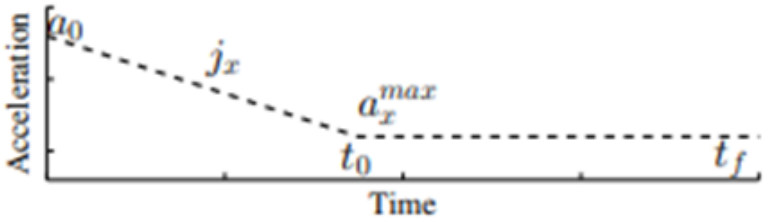

Figure 2.3. Lateral acceleration profile for steering [3].

In Fig. 2.3, the lateral acceleration profile models the vehicle's behavior. It shows how as time increases, the vehicle maintains its lane and stays in a straight line.

2.3 Critical Zone for Leading Vehicles

To model a critical zone for a leading vehicle, the assumption is that the leading vehicle can abruptly stop at a point in time. The critical time, T_{crit}, in regards to a leading vehicle is the minimum amount of time the automated vehicle needs to initiate an emergency action such as braking or steering. T_{brake} is the critical time gap the ego vehicle must maintain to avoid colliding with the leading vehicle by braking. V_{ego} is the velocity of the ego vehicle, S_{ego} is the braking distance based on the relevant acceleration profile, and S_{min} is the minimum longitudinal safety gap that must be maintained by two vehicles after it has stopped. Instead of braking, the ego vehicle has the option of steering to avoid a collision as well. T_{steer} is the critical time the ego vehicle must maintain to prevent hitting the leading vehicle by steering the vehicle away. With reference to the lateral acceleration profile, $S_{latsafe}$, is the required time for the automated vehicle to travel a safe distance. W_s, W_l, and W_e are the minimum lateral distance that must be maintained between vehicles, the width of the leading vehicle, and the width of the ego vehicle, respectively. $Offset$ is the relative velocity and relative lateral distance between the ego vehicle and surrounding vehicles.

$$T_{brake} = \frac{S_{ego} + Smin}{V_{ego}} \tag{2.1}$$

$$S_{latsafe} = (((1/2) * W_{lead}) + ((1/2) * W_{e}go) + (W_{min}) - (offset)) \tag{2.2}$$

$$V_0 = \frac{(((\mathrm{j}_y^{max} * t_{init}^2))}{2} \tag{2.3}$$

$$S_0 = \frac{((\mathrm{j}_y^{max} * t_{init}^3))}{6} \tag{2.4}$$

$$V_1 = ((V_0) + ((a_y^{max}) * (t_1 - t_{init}))) \tag{2.5}$$

$$S_1 = S_0 + (V_0 * (t_1 - t_{init})) + \frac{(a_y^{max}((t_1 - t_{init})^2)}{2} \tag{2.6}$$

$$S_2 = S_1 + (V_1 * (t_2 - t_1)) + \frac{(a_y^{max}((t_1 - t_{init}^2)}{2} - \frac{J_y^{max} * ((t_2 - t_1^3)}{6} \tag{2.7}$$

$$S_3 = (S_2) + V_y^{max}(t_f - t_2) \tag{2.8}$$

$$T_{crit} = min(T_{brake}, T_{steer}) \tag{2.9}$$

$$S_{crit} = T_{crit} * V_{ego} \tag{2.10}$$

2.4 Critical Zone for Trailing Vehicles

For trailing vehicles, critical time is how long it takes the ego vehicle to abort the lane change and move away from the initially intended path of motion. In order to properly model critical zones for trailing vehicles, all possible scenarios for what behavior the vehicle

could exhibit has to be considered. There are three different options that a trailing vehicle could respond to: decelerate and allow the ego vehicle to proceed with the lane change, fail to notice while continuing at its current speed, and the worst-case scenario where the vehicle accelerates to prevent lane change. A driver accelerating to cut off the ego vehicle is typical in dense traffic where there is a preference of reduced time headway and small inter-vehicle gaps. An assumption of a constant acceleration model is used for this. When the automated vehicle avoids approaching the vehicle by safely traveling the minimum lateral distance, the amount of time needed is known as $T_{critical}$. $S_{latsafe}$ is calculated the same way as it is done for leading vehicles. The critical time gap in regard to trailing vehicles only involves steering. To calculate critical zone, the safety critical distance is needed. This is determined by finding the distance traveled by each of the participating vehicles by using

$$S_{ego} = V_{ego} * T_{crit} \tag{2.11}$$

$$S_{trail} = V_{trail} + \frac{(a_{trail} * T_{crit}^2)}{2} \tag{2.12}$$

$$S_{crit} = max((S_{trail} - S_{ego}), min(S)) \tag{2.13}$$

where V_{ego} and V_{trail} are longitudinal velocities of the ego vehicle and trailing vehicle, respectively, a_{trail} is assumed constant acceleration for the trailing vehicle, S_{min} is the minimum safety distance the EV must maintain ahead of the trailing vehicle to determine the critical zone, $S_{critical}$ has to be calculated for every offset in order to create a critical zone.

3. MOTION PLANNING - CONTROLLER

An ego vehicle has to be positioned at the center or as close as possible to the center of the target lane during the planning of a lane change maneuver. A safe trajectory has to be one that observes actuator, comfort, and safety constraints [17]. The algorithm has to be able to plan the motion in the longitudinal and lateral directions to reach the desired final position while ensuring that an evasive maneuver is always possible. Automated vehicles have constraints such as actuator saturation and lane width, which makes using Model Predictive Control with the receding horizon a good idea for motion planning. Point mass model is used for the motion of the vehicle, it is discretized using zero-order hold. The states of the ego vehicle are represented by the following:

- x, y represent the position.
- V_y, V_x represent the velocity.
- a_x, a_y represent the acceleration.
- J_x, J_y represent jerk which is the input to the system.

For the EV to go from its initial position to a desired final position, the longitudinal motion planning is represented like a standard QP optimization problem.

$$minJ = \sum_{\mathrm{i}=0}^{N-1} (X(\mathrm{i})^T Q X(\mathrm{i}) + u(\mathrm{i})^T R u(\mathrm{i})) + X(N)^T P_f X(N) \tag{3.1}$$

Subject to

$$X(K+1) = A_d X(k) + B_d U(k), k = 0, , N \tag{3.2}$$

$$X(0) = [x^{initial}, V_x^{(initial}, 0]^T \tag{3.3}$$

$$X(N) = [x^{final}, V_x^{final}, 0]^T \tag{3.4}$$

$[x^{min}, V_x^{min}, a_x^{min}]^T \leq X(k) \leq [x^{max}, V_x^{max}, a_x^{max}]^T$

$$A_d = \begin{bmatrix} 1 & t_s & \frac{t_s^2}{2} \\ 0 & 1 & t_s \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_d = \begin{bmatrix} \frac{t_s^3}{6} \\ \frac{t_s^2}{2} \\ t_s \end{bmatrix}$$

U is the control input to the systems with weight R as the cost. Constraints are set to position to ensure that the limits of the critical zones are always respected. To include actuator limitations or a smooth maneuver, the constraints for velocity, acceleration and jerk can be adjusted. Planning the lateral trajectory is similar to that for the longitudinal except that in this case, the lateral position constraints change with longitudinal position. The constraints for critical zones are a function of critical time $T_{critical}$, this can be calculated for the entire horizon using the derived longitudinal trajectory. To calculate lateral position constraints;

The longitudinal position $x(t)$ for the entire prediction horizon is determined by longitudinal motion planning.

The critical zone limits for leading and trailing vehicles are evaluated then used to find $T_{critical}$ with respect to the surrounding vehicles for a given value of position.

The value of $T_{critical}$ is substituted in the lateral expressions for the critical zone limits then used to determine the lateral constraints.

Solving the MPC helps to find the lateral trajectory which is done at a regular sampling interval.

3.1 First Scenario: Car Following

The first scenario is an example where the ego vehicle follows the lead vehicle while making sure it does not enter any of the critical zones. The critical zone is modeled based on the worst case scenario that the lead vehicle can abruptly come to a full stop or brake at any point in time. In this scenario, the lead vehicle maintains a constant velocity and travels in a straight line. This situation uses just the longitudinal profile as it studies just the movement of the ego vehicle on the same lane. The goal here is for the ego vehicle to safely follow the lead vehicle without any collision. This scenario involves the ego vehicle traveling at a constant velocity of 10 m/s while the lead vehicle maintains its velocity of 20 m/s. This scenario was simulated in Matlab with the use of the Model Predictive Control Toolbox, Simulink, Sensor Fusion and Tracking Toolbox, and the Automated Driving Toolbox. In order, to create the lane following system in Simulink, the solution was broken down into different subsystems;

3D Scenario Environment

A 3D scenario has to be added to the test bench in order to create and simulate the scenario in 3D. Fig. 3.1 uses a "game like" environment which is available as one of the applications in the Matlab toolbox. This subsystem gives details on the cars' specifications, type of road, weather conditions, camera sensors and radar sensors used. It is responsible for converting the coordinates of the vehicles with respect to the center of the rear axle to a 3D coordinate system. The outputs from the cameras and radars go to the systems that detect lane boundaries, collisions and ensuring that the ego vehicle maintains a safe distance from the lead vehicle. It also ensures that the ego vehicle always stays within its intended lane.

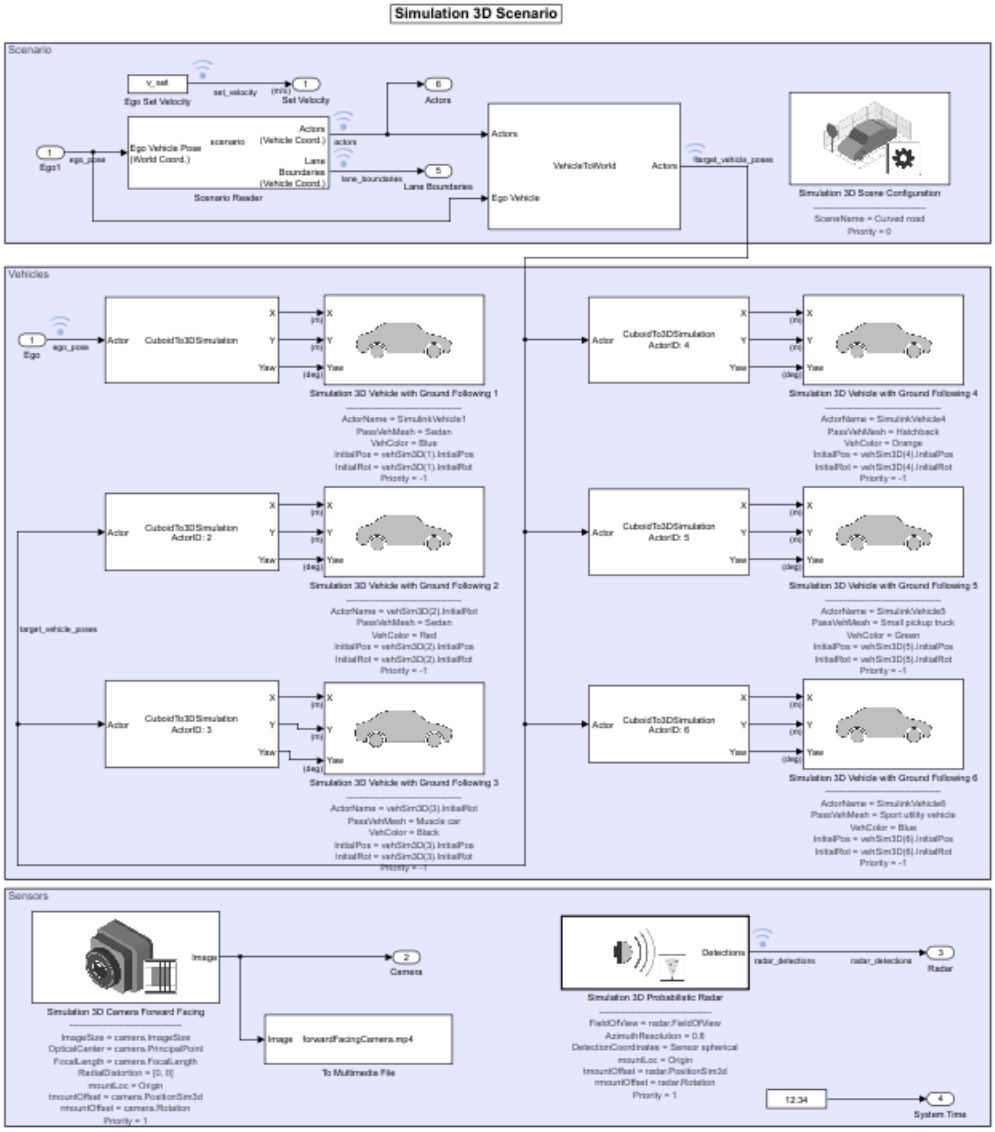

Figure 3.1. Simulation 3D scenario

Lane Marker Detector

The block in Fig. 3.2 contains the algorithm that allows the ego vehicle to know where the lane boundaries are. The outputs from the front camera is taken as an output. The front camera provides information about the lanes. It lets the vehicle know the type of marking s on the line, whether the lane has dashed or straight lines to determine where the vehicle can initiate a lane change.

Forward Vehicle Sensor Fusion

Fig. 3.3 is an algorithm that takes the detection from the radar and camera sensor. The image from the camera sensor is an input while the output is the vehicle information which are detected as boxes. These detection from the vision and radar sensors become inputs that are entered into the next block for clustering .The clustering algorithm is applied to the data to separate it into similar groups [20] . After the data is clustered, they are merged with the detection from the vision detection. Then a joint probabilistic data association tracker is used to determine the tracks.

Lane Following Decision Logic

This the model that determines how the lane following logic will work. The lane following decision logic in Fig. 3.4 takes tracks determined in the Forward Vehicle Sensor Fusion block as input. It also takes in the width of the lane and the position selector as inputs. These inputs go into the block that is responsible for selecting a lead vehicle. It outputs the relative distance, relative velocity and the track index. It also uses the lane information to find where the center of the lane will be. The lead vehicle that is selected then becomes the Most Important Object (MIO) as it is a target vehicle.

Lane Following Controller

The model in Fig. 3.5 details the values for the steering angle and the controls for acceleration. It utilizes the center of the road and the longitudinal velocity to determine the

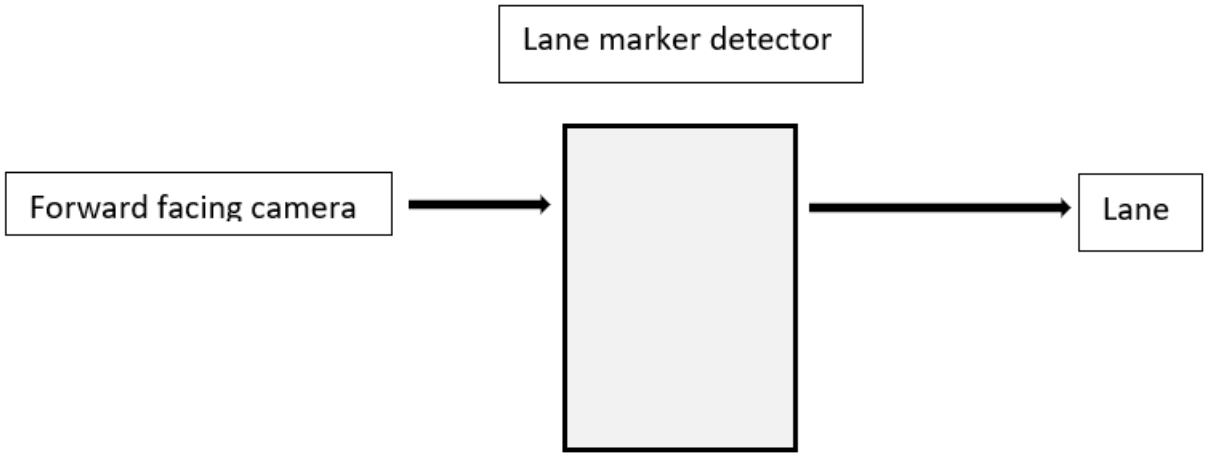

Figure 3.2. Lane marker detector

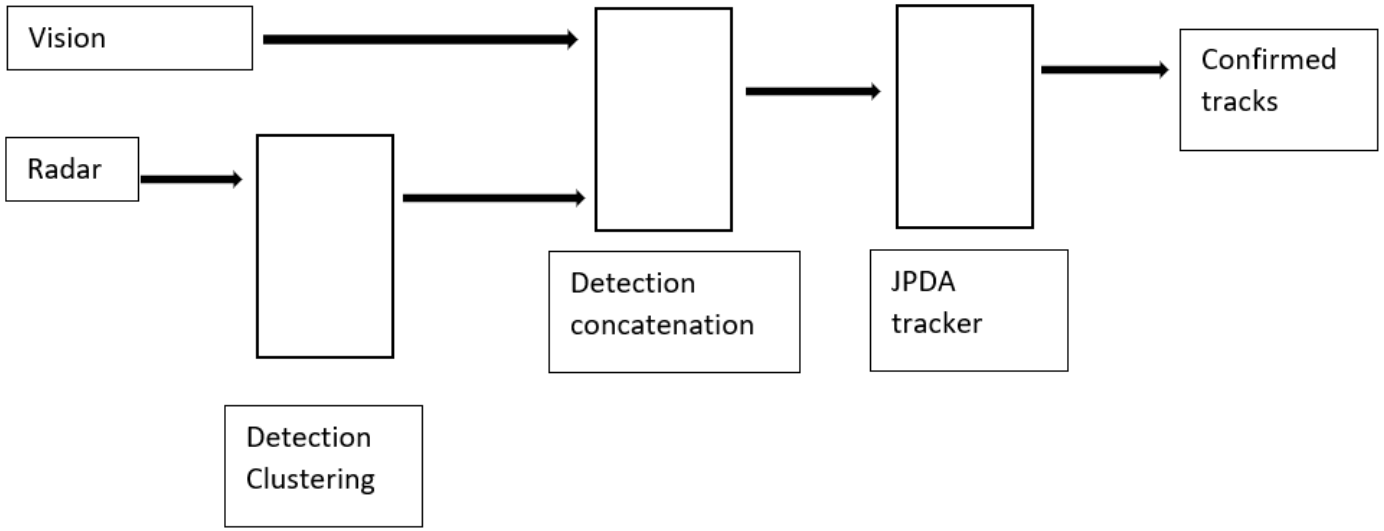

Figure 3.3. Forward vehicle sensor fusion

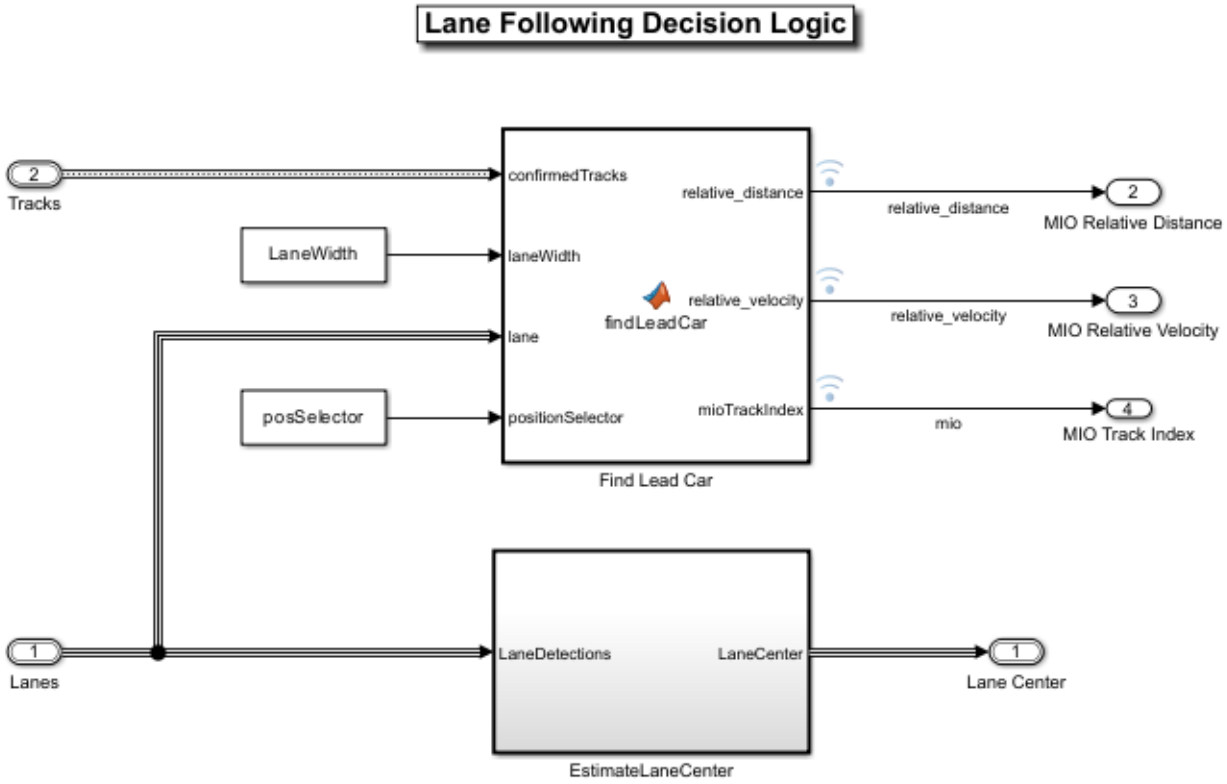

Figure 3.4. Lane following decision logic

curvature of the road, the lateral deviation and the relative yaw angle. The Model Predictive Controller takes the set velocity, longitudinal velocity, curvature, lateral deviation, relative yaw angle of the lead car, relative distance and relative velocity as inputs. The controller then outputs the steering angle and the acceleration mode of the vehicle. A watchdog controller is used as a safety mechanism while the under the assumption that the lead vehicle could brake to a stop at any time. A watchdog braking controller is used as a fail safe mode [21] to ensure that the ego vehicle brakes if it ever functioning outside of the determined ranges. The watchdog controller constantly monitors the MPC and calculates the range it is meant to work within, if the controller enters a state which is considered unsafe, the mechanism calls for a corrective action. It could make the decision to reset the controller[22]. The output from the MPC controller and braking controller go into a different block which now determines if it is safe for the vehicle to proceed as intended.

Bicycle Dynamic Model

The ego vehicle is modeled using the bicycle dynamic model in Fig. 3.6. This models a rigid two-axle single track vehicle body model to compute the longitudinal, lateral and yaw motion. The block takes into account the mass of the vehicle, the aerodynamic drag and the weight distribution between the axles.

Metrics Assessment Model

A metrics assessment model is basically the foundation of the entire lane following system. This ensures that the lateral motion from the center of the lane is within the set range. It makes sure that the vehicle is on the appropriate lane and that it stays in its lane throughout the simulation. The metrics model in Fig. 3.7 also ensures that vehicles maintain the time gap which is calculated from the critical zones. It ensures that there is no collision by preventing the vehicle from entering the critical zones.

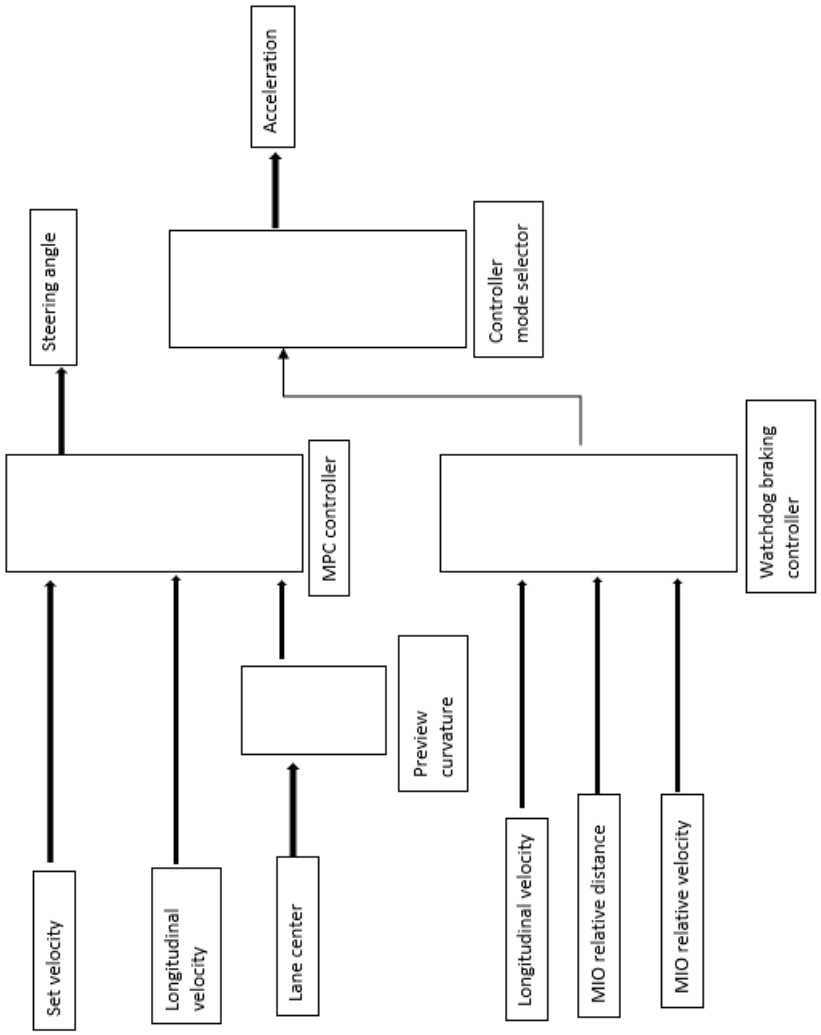

Figure 3.5. Lane following controller

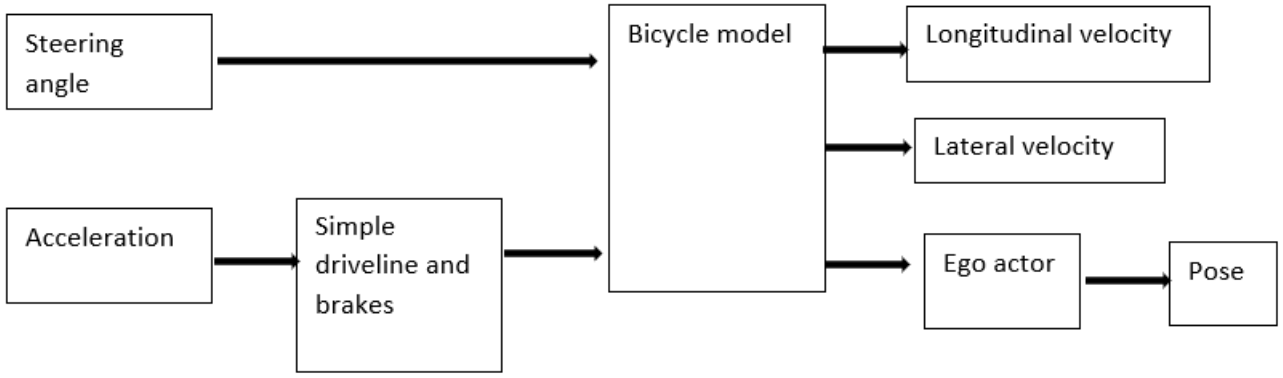

Figure 3.6. Vehicle dynamics

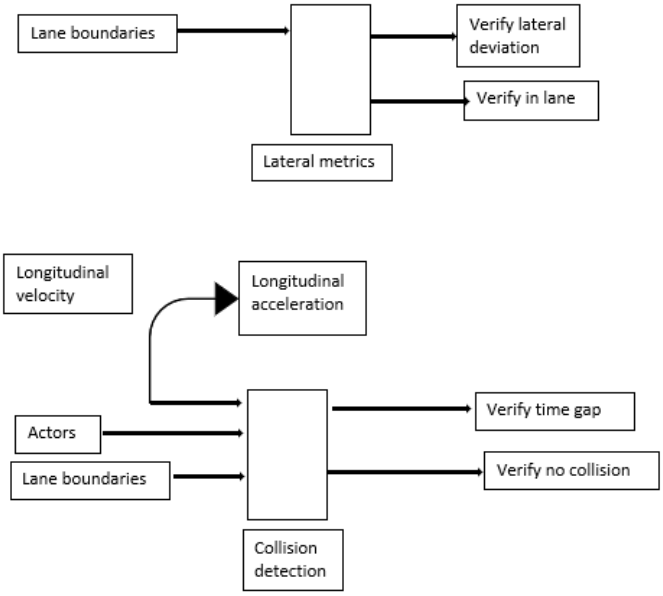

Figure 3.7. Metrics Assessment model

3.2 Second Scenario: Lane Changing

This scenario involves a complete lane change. The EV goes from its initial lane to the center of a target lane where it has to place itself outside the parameters of the critical zones for both the leading and trailing vehicles. In this scenario, the longitudinal profile remains constant while the lateral acceleration profile changes. The ego vehicle has the ability to select a lead vehicle and determine when it's safe to plan a trajectory. The EV operates under the assumption that the lead vehicle can brake at any time. and trailing vehicle can ignore the intention of the ego vehicle to initiate a lane change and accelerate. With these considerations, the ego vehicle should be able to initiate a lane change and complete it safely without a collision. If either of the worst case scenarios happen, the ego vehicle initiates a response by either to decelerating, braking, steering or choosing to abort the lane change and remain in its current lane in order to avoid a collision. The test bench model in this case, much similar to the model from scenario 1 above has multiple subsystems with the difference of a lane change planner and controller. The test bench has the following subsystems;

Scenario and Environment Model

The scenario and environment block in Fig. 3.8 determines what the scene during the simulation will be. It determines the number and type of vehicles, sensors and map data. It controls the positions of the vehicles, uses a camera model to detect objects and outputs clustered data from the detections using a statistical model. The system also uses an INS sensor to output the position, velocity and orientation of the ego vehicle.

Vehicle Dynamics Model

A vehicle dynamics block in Fig. 3.9 uses a bicycle model for the ego vehicle.

Metrics Assessment Model

A metrics assessment subsystem is responsible for ensuring that the criteria for the lane change is met. Fig. 3.10 includes a collision detection and jerk metrics block. For the collision

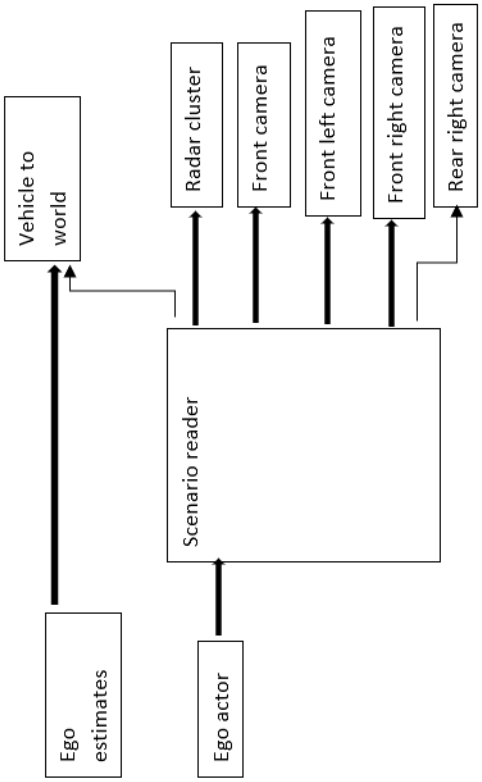

Figure 3.8. Scenario and environment model

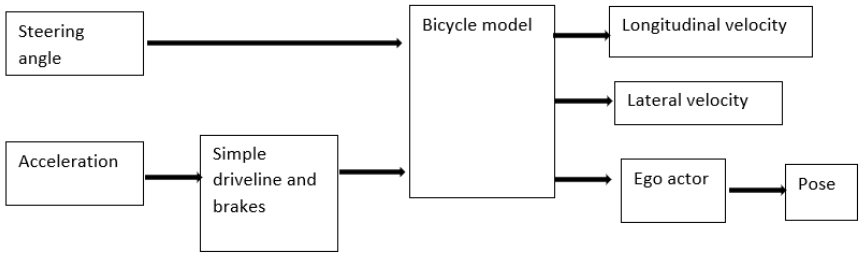

Figure 3.9. Vehicle dynamics model

detection, lane boundaries, data of the target vehicles, trajectory information, parameters from the planner and longitudinal velocity are used as inputs. These values are used to calculate the critical zone. The collision detection block verifies that there will be no collision, it verifies the time gap using the critical zone equations for braking and steering. It also checks to ensure that the vehicle is safe. The jerk metrics blocks is used to calculate the longitudinal and lateral jerk.

Lane Change Planner

The lane change planner in Fig. 3.11 is one of the most important subsystems in this test bench. This planner is also applied in scenario three, it is capable of handling different traffic scenarios. It has a state conversion block which uses a frenet converter to translate the coordinates of the ego vehicle to states. This output is then used to find all the surrounding vehicles. The next block is the terminal state sampler. The terminal state sampler considers the current state of the vehicles in the scenario, the parameters of the planner and uses these to sample different terminal states for different behaviors. This is used to determine the decision that the ego vehicle will make. This is whether the EV will proceed with a lane change or what lane it will be moving to. The motion prediction box predicts the trajectory of the other vehicles and uses the guess to generate different ways the ego vehicle could perform the maneuver. The motion planner calculates the cost of performing each of the generated trajectories then determines which would be the most effective.

Lane Change Controller

The lane change controller in Fig. 3.12 is responsible for ensuring that the ego vehicle travels along the trajectory that was generated. It uses MPC to adjust the longitudinal acceleration and front steering angle of the EV within the set constraints for velocity, acceleration and steering angle.

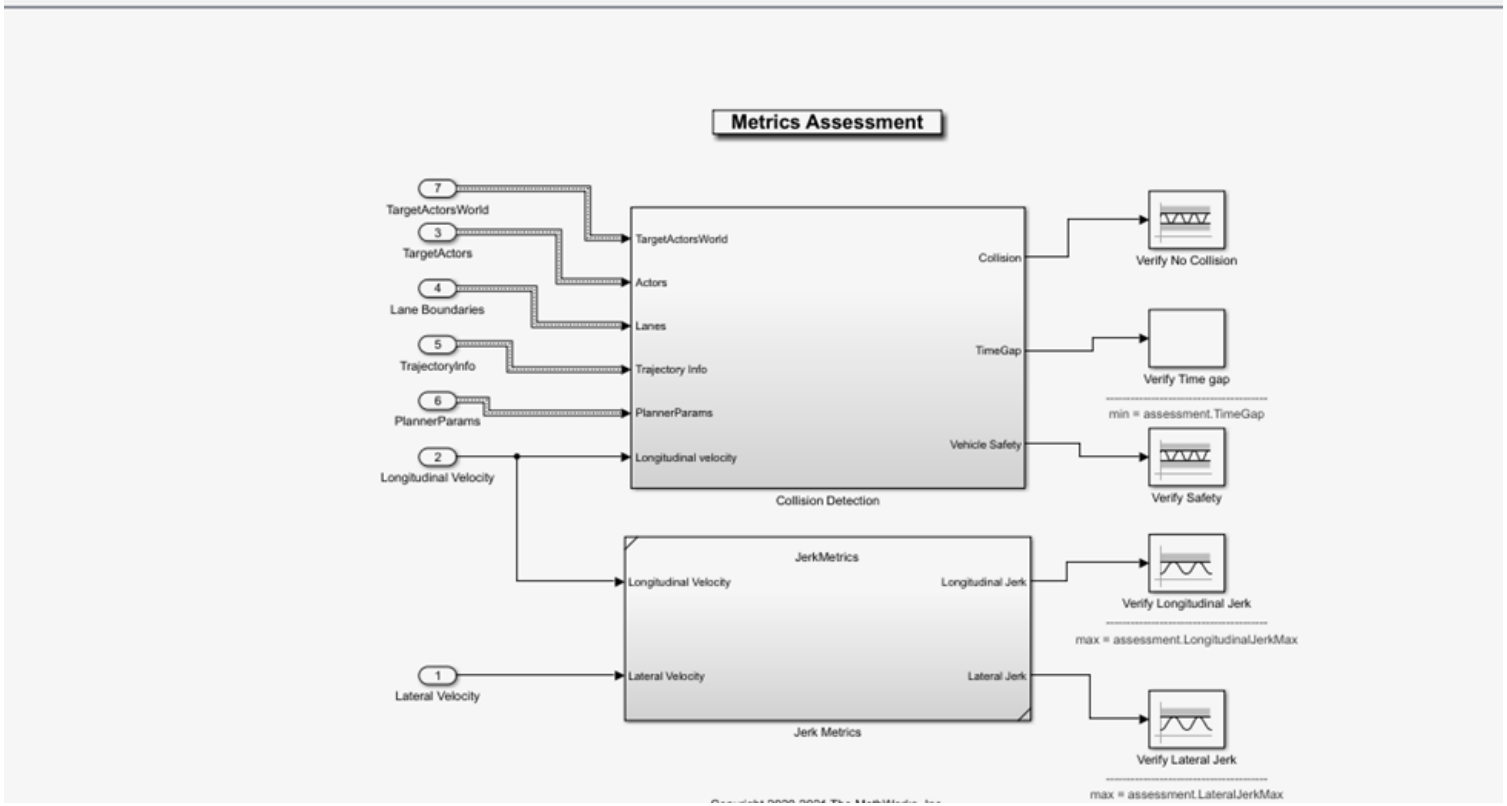

Figure 3.10. Metrics Assessment model

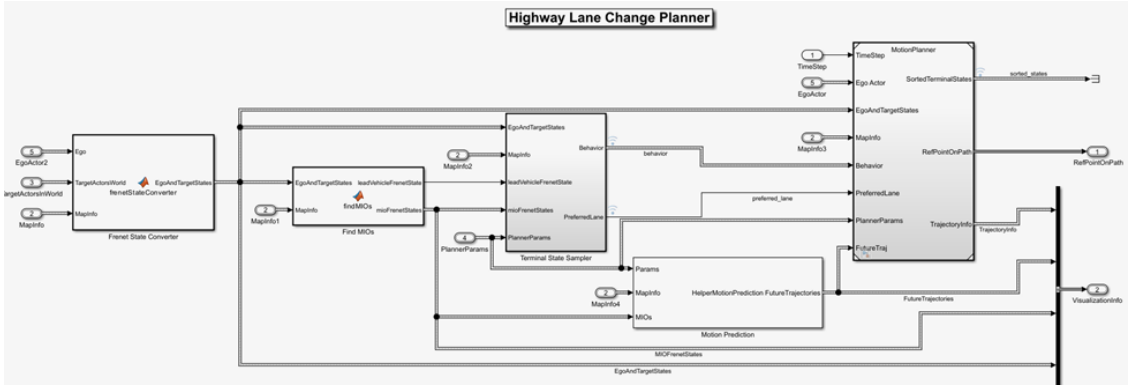

Figure 3.11. Lane change planner

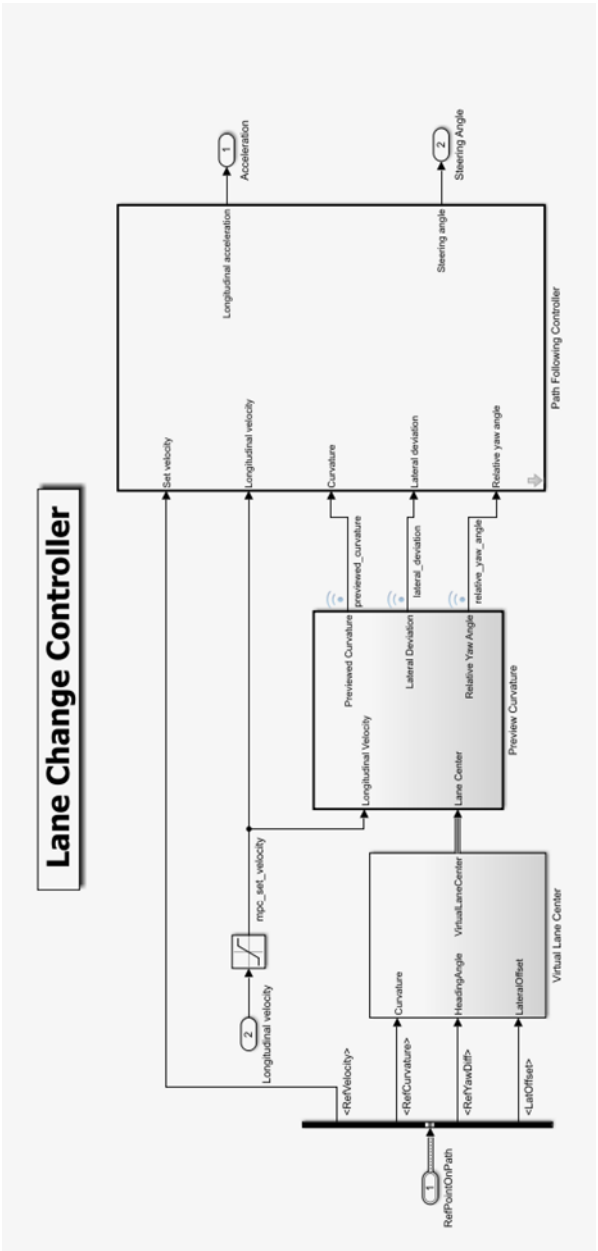

Figure 3.12. Lane change controller

Sensor Block

A sensor block is used by the ego vehicle to detect lane boundaries, objects and other vehicles on the road. This block in Fig. 3.13 receives data from the cameras and radar sensor on the EV and concatenates the data. It is responsible for tracking all the vehicles.

3.3 Third Scenario: Lane Change with Deceleration

In the third scenario, the autonomous vehicle completes a lane change but in this case, the lead vehicle decelerates. The ego vehicle has to appropriately respond to the deceleration. This scenario shows the the changes in both the lateral and longitudinal profiles both change. The test bench model used here is the same as that in scenario 2.

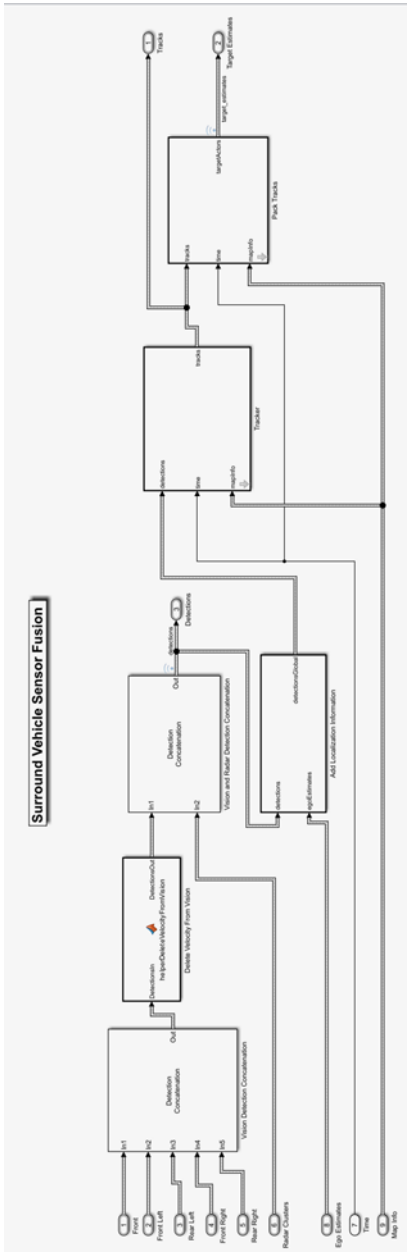

Figure 3.13. Sensor Block

4. RESULTS AND CONCLUSION

4.1 Scenario 1 results

Scenario 1 simulates a lane following traffic scenario. In this scenario, only one lead car is the target vehicle and the worst case scenario is that the lead car can abruptly brake. The critical braking time was used as the time gap instead of a constant value. This allowed for flexibility as the ego vehicle used the equations for braking time and lateral safe distance to constantly update what the critical zones were. Scenario 1 is a lane following scenario so there are no lateral changes here. In figure 4.1, the scenario is initially simulated for 10s to predetermine if there are collisions. The cars are represented as boxes and the graph on the left of the plot shows that there were no collisions in the scenario. Figure 4.2 shows the scenario in the 3D scenario where the blue car is the ego vehicle. Figure 4.3 shows the bird eye plot of scenario on the right and the scenario from the perspective of the driver in the ego vehicle. It also highlights the everything within the radar of the EV which is shown by the blue and red quadrants. Figure 4.4 is a plot showing the results of the relative longitudinal distance between the ego vehicle and the lead vehicle which slowly increased over time. The absolute velocity and acceleration stayed constant as both vehicles maintained their velocities of 10 m/s and 20 m/s the entire time. Figure 4.5 shows the lane boundary lateral offsets, lateral deviation, relative yaw angle and steering angle for the ego vehicle. The results correspond to the behavior of the vehicle as it maintained it's lane the entire time and stayed in the center of the lane. The results show that were no changes in the lateral acceleration profiles and the vehicles maintained their initial velocities through out the simulation.

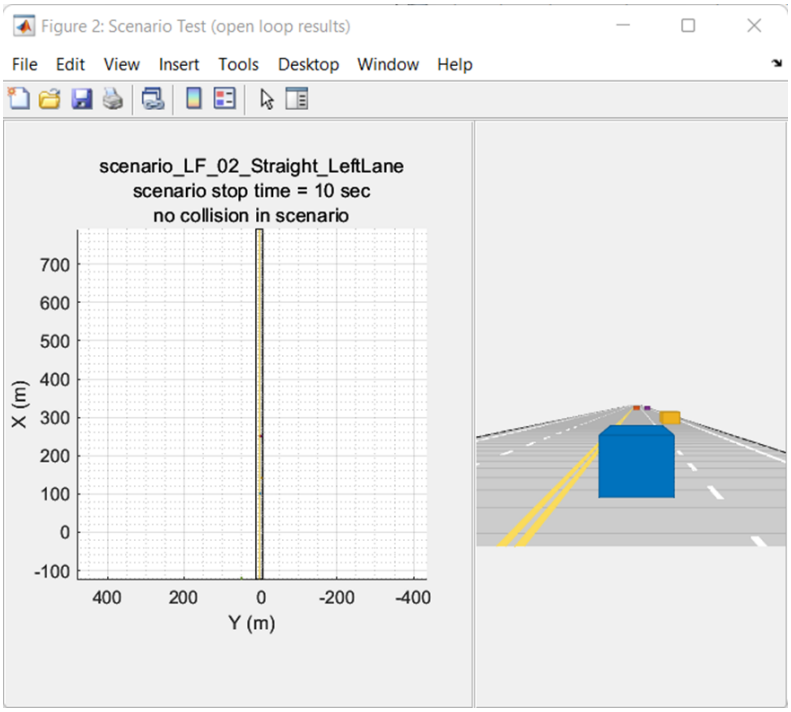

Figure 4.1. Lane following scenario

Figure 4.2. 3D scenario environment

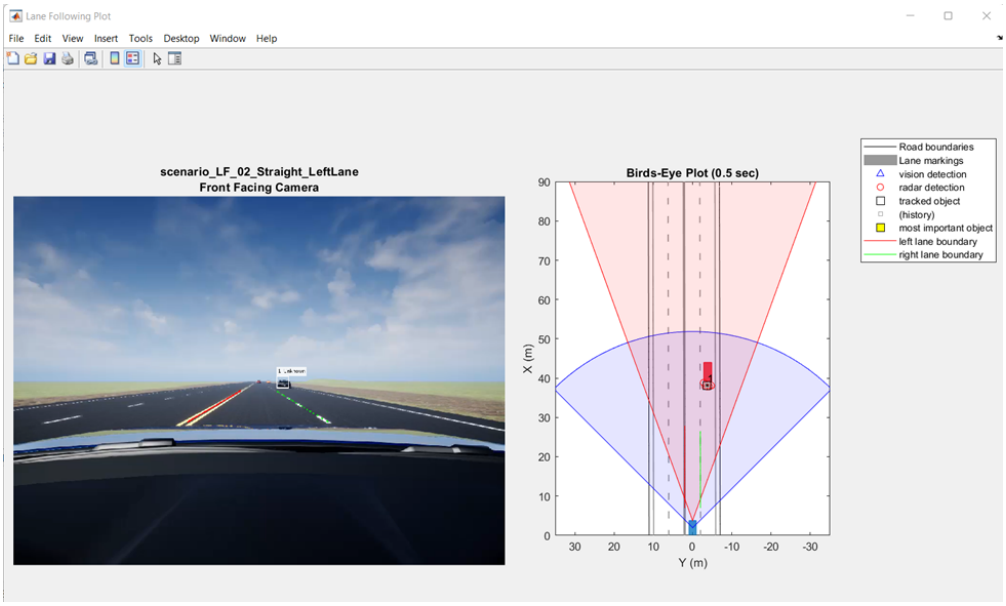

Figure 4.3. Bird eye plot

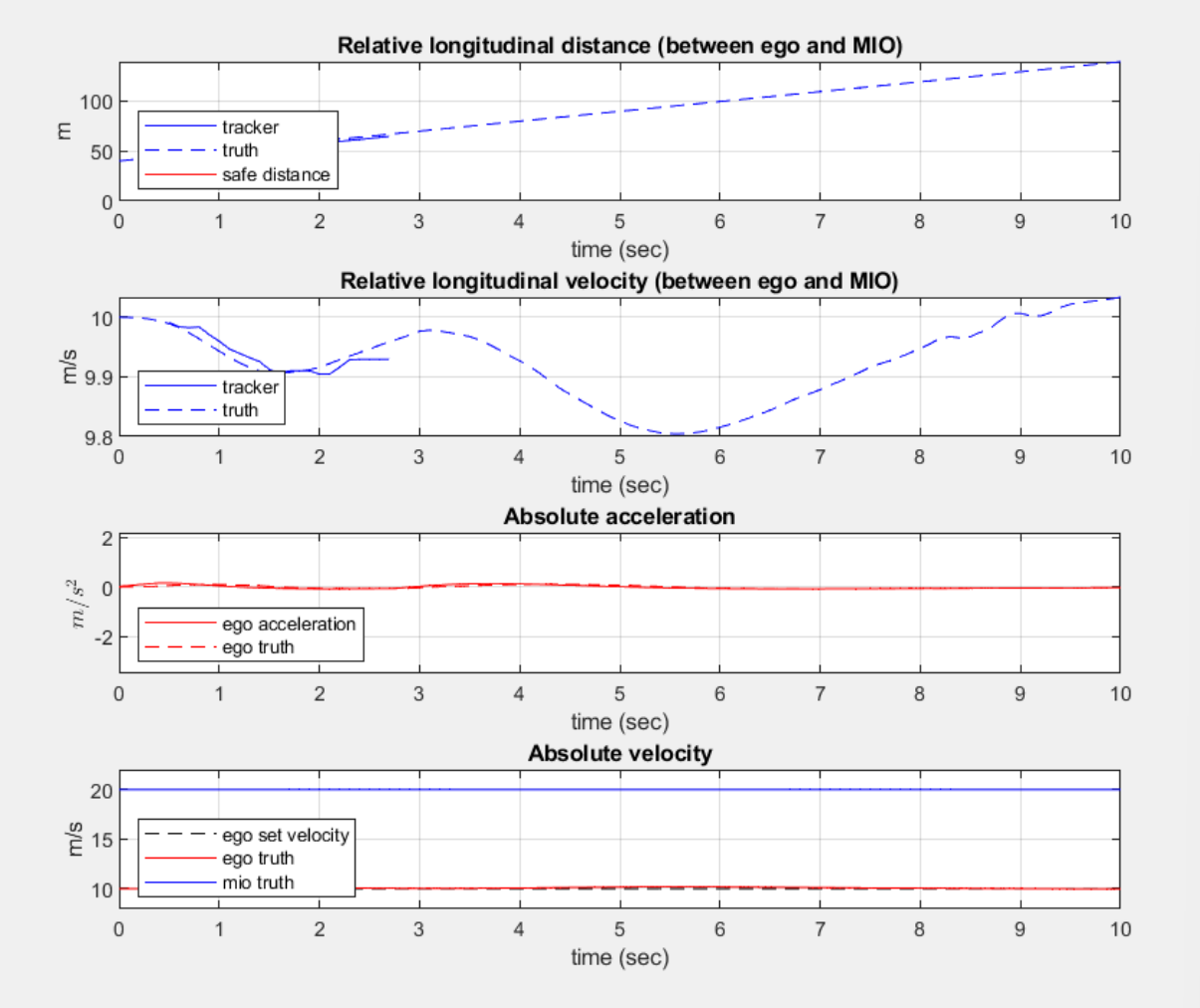

Figure 4.4. Longitudinal profiles

This shows how the EV maintained the braking distance calculated from the leading vehicle critical zone equations.

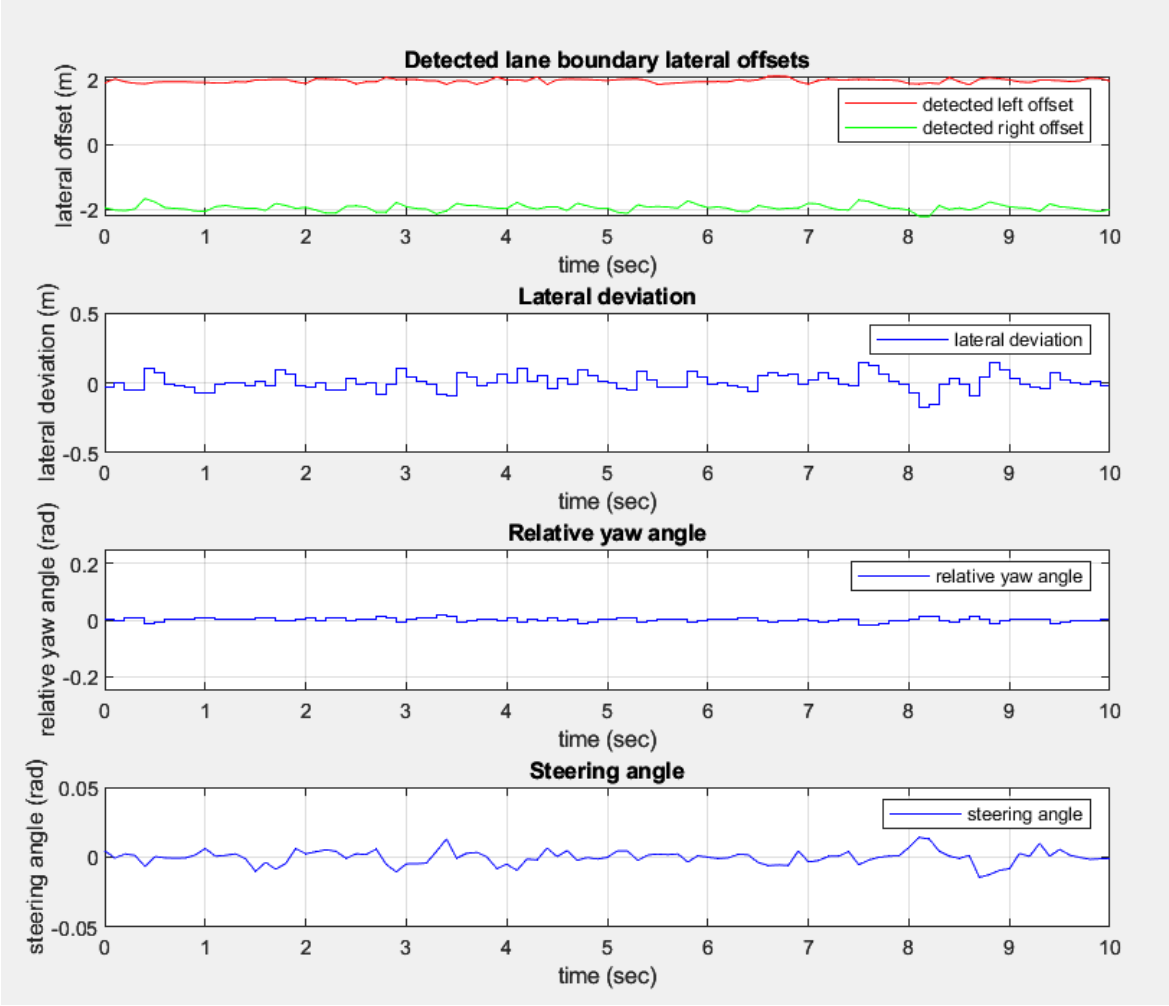

Figure 4.5. Lateral profiles

The vehicle stays in the center of its lane through out the manuever and made no lateral changes.

4.2 Scenario 2 results

For scenario 2, the ego vehicle begins with a set velocity of 20 m/s while the lead vehicle that is selected in the opposite lane travels with 24 m/s. The ego vehicle stays in it's line while it calculates the best time to make a lane change then initiates the lane change, completes it and maintains its position in the new lane while ensuring there is no collision. The simulation time for this scenario was 25s to allow the ego vehicle enough time to initiate a lane change, complete it then proceed to follow the lead vehicle. Figure 4.7 shows the performance of the Model Predictive Controller by displaying results of the curvature of the road, lateral deviation, heading angle, ego velocity and steering angle. The graph for the curvature remained flat the whole time as it was a straight road. Around 10s, all the plots have a significant change as that is when the lane change occurred. The increase and decrease of lateral deviation between 8 to 10s when the EV initiates its maneuver. The heading angle shows the slight adjustment in degrees as the EV has to steer from one lane to another lane. The Ego Velocity graph shows the EV slow down briefly to perform the lane change safely and then adjust it's speed back to its initial velocity. The steering angle shows a minor adjustment in the angle from when the EV initiated the lane change and then when it completed the maneuver. Figure 4.8 shows the graphs for the acceleration, longitudinal and lateral jerk. These graphs just like in Figure 4.7 show changes around the 10s mark when the lane change occurred. The plots all flatten out around 12s showing that the lane change had been completed.

The lateral and longitudinal graphs show the acceleration of the vehicle and its behavior through out the lane change maneuver. The lateral profile shows how the controller adjusts itself during the lane change and then after the lane change is completed, the graph evens back out as the vehicle now goes back to a lane following model.

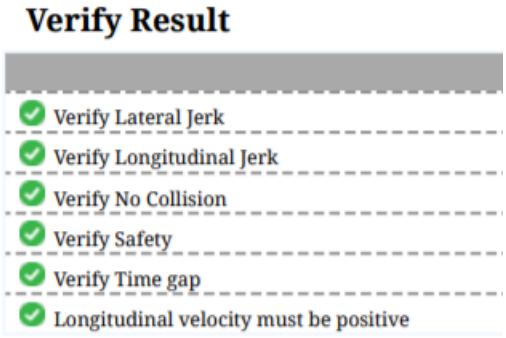

Figure 4.6. Safety checklist

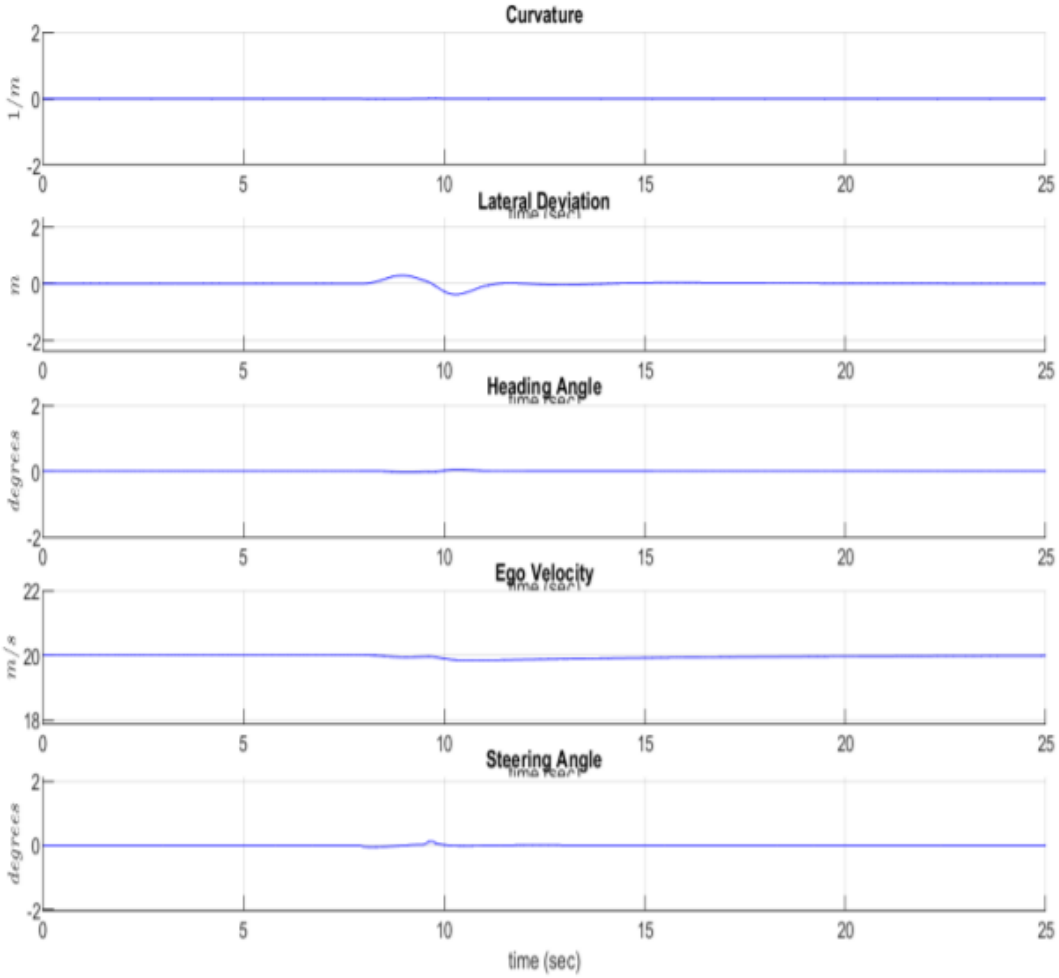

Figure 4.7. Controller performance

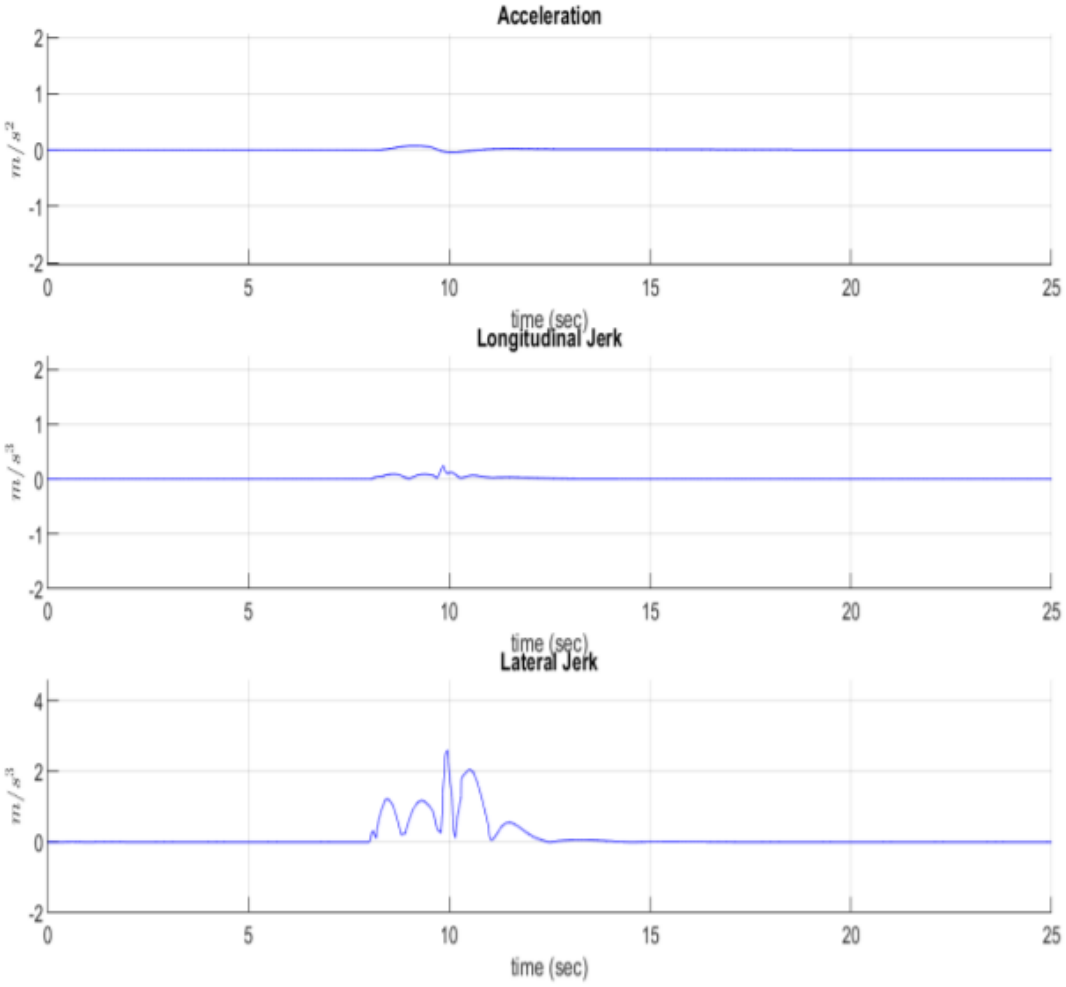

Figure 4.8. Lane change metrics

4.3 Scenario 3 results

In scenario 3, the ego vehicle starts with a velocity of 15 m/s while the lead vehicle starts with 14 m/s then slows down to 8 m/s before it increases to 13 m/s. There are also trailing vehicles in these scenarios which are considered. The results show the longitudinal and lateral profiles of the ego vehicle as well as the steering angle during the lane change. The critical zone equations are used to constantly ensure that no collision occurs and that the time gap values are relevant to the current position and time of the vehicles. Figure 4.9 shows the ego vehicle in its initial position while Figure 4.10 shows the vehicle in its final position after it has completed the lane change. Figure 4.11 shows the safety checklist that the algorithm goes through before beginning the simulation. Figure 4.12 there are three points of interest where disturbances occur in the graph. The first one is right when the ego vehicle goes from stationary then it accelerates and on its lane in less than 2s. Around 8s, the graphs show the ego vehicle begin to initiate its lane change and the graphs settles again after the maneuver has been completed. The lead vehicle abruptly slows down around 14s, the response of the EV can be seen in all three graphs. Figure 4.13 shows the performance of the controller in terms of the curvature of the road, lateral deviation, heading angle, ego velocity and steering angle. The graphs show the behavior of the ego vehicle again before the 2 second mark, around 8s and then around 14s showing it accelerated, performed a lane change then slowed down when the lead vehicle decelerated. The lateral and longitudinal results show the point in time when the vehicle begins its lane change. This lateral profile in comparison to that from scenario two, shows the two periods of interest. It shows the moment when the vehicle begins its lane change and how the lateral controllers adjusted and also again when the lead vehicle braked to a stop.

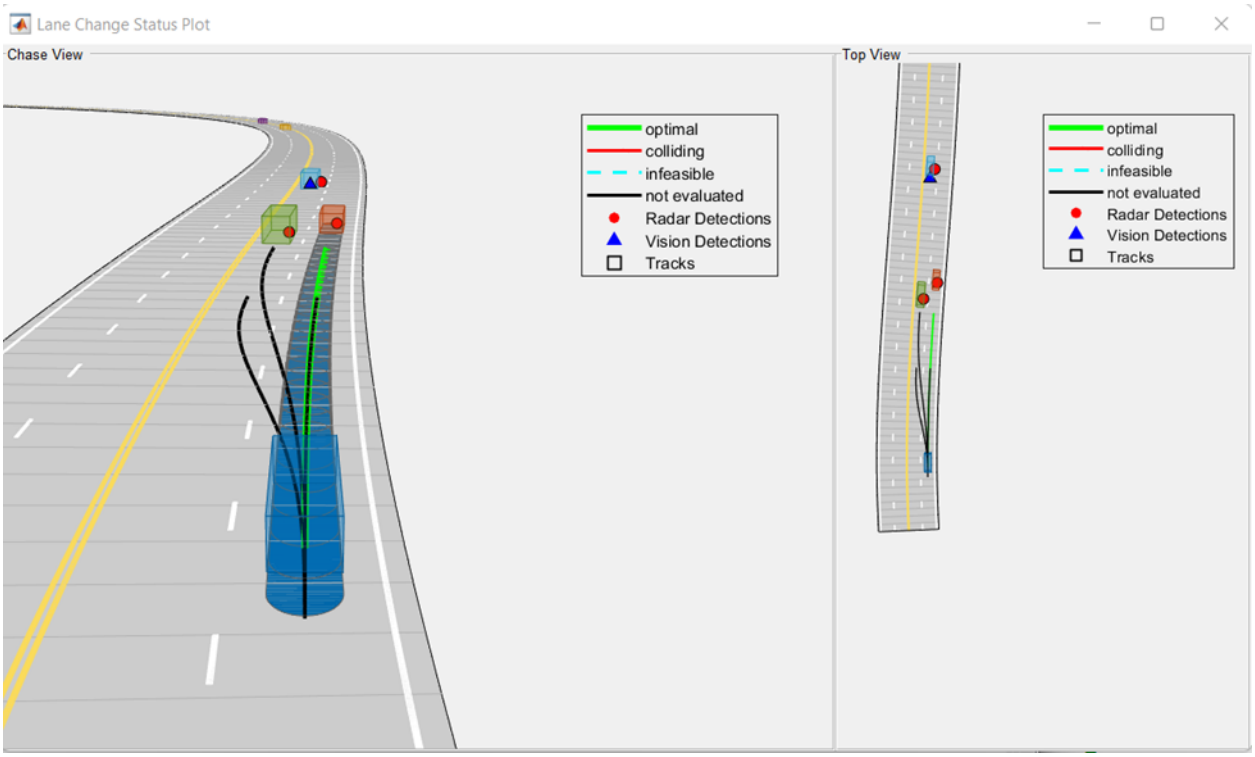

Figure 4.9. Initial position of ego vehicle

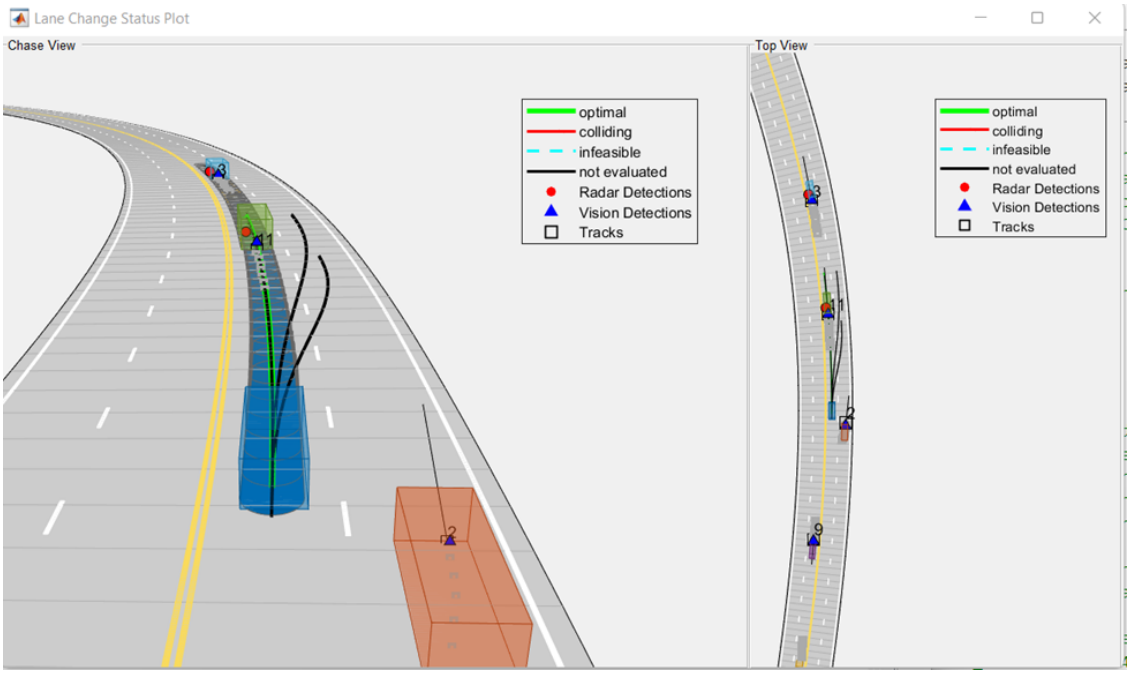

Figure 4.10. Final position of ego vehicle after lane change

Verify Result

Name
Verify Lateral Jerk
Verify Longitudinal Jerk
Verify No Collision
Verify Safety
Verify Time gap
Longitudinal velocity must be positive

Figure 4.11. Safety checklist

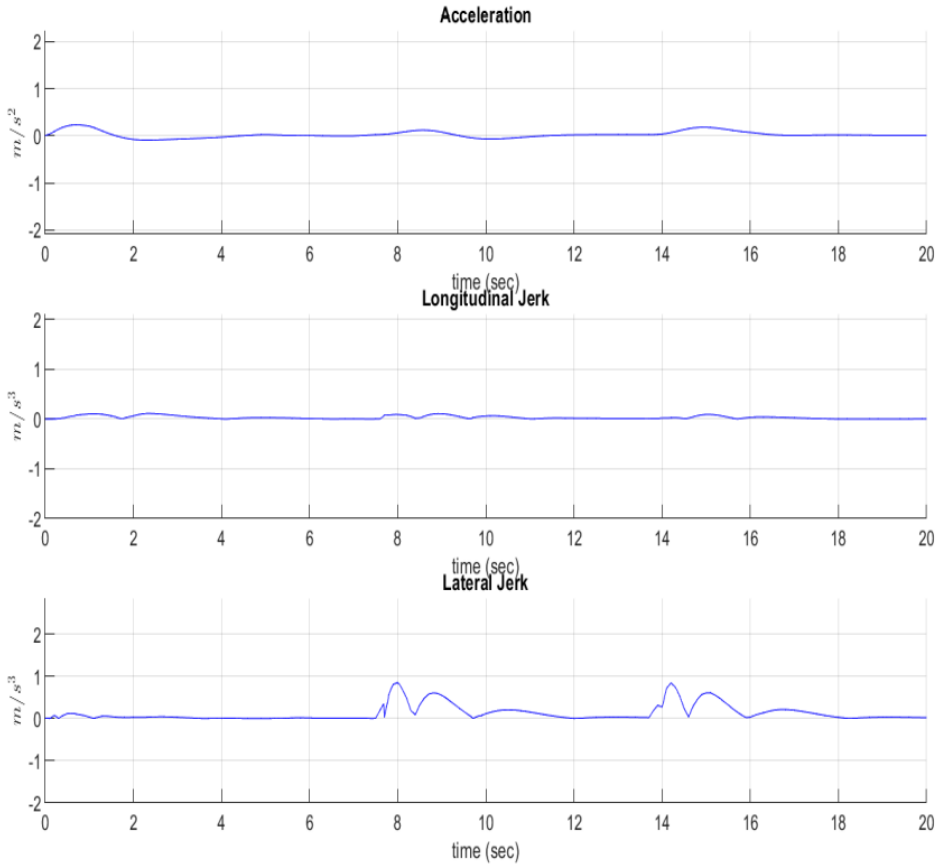

Figure 4.12. Lane change metrics

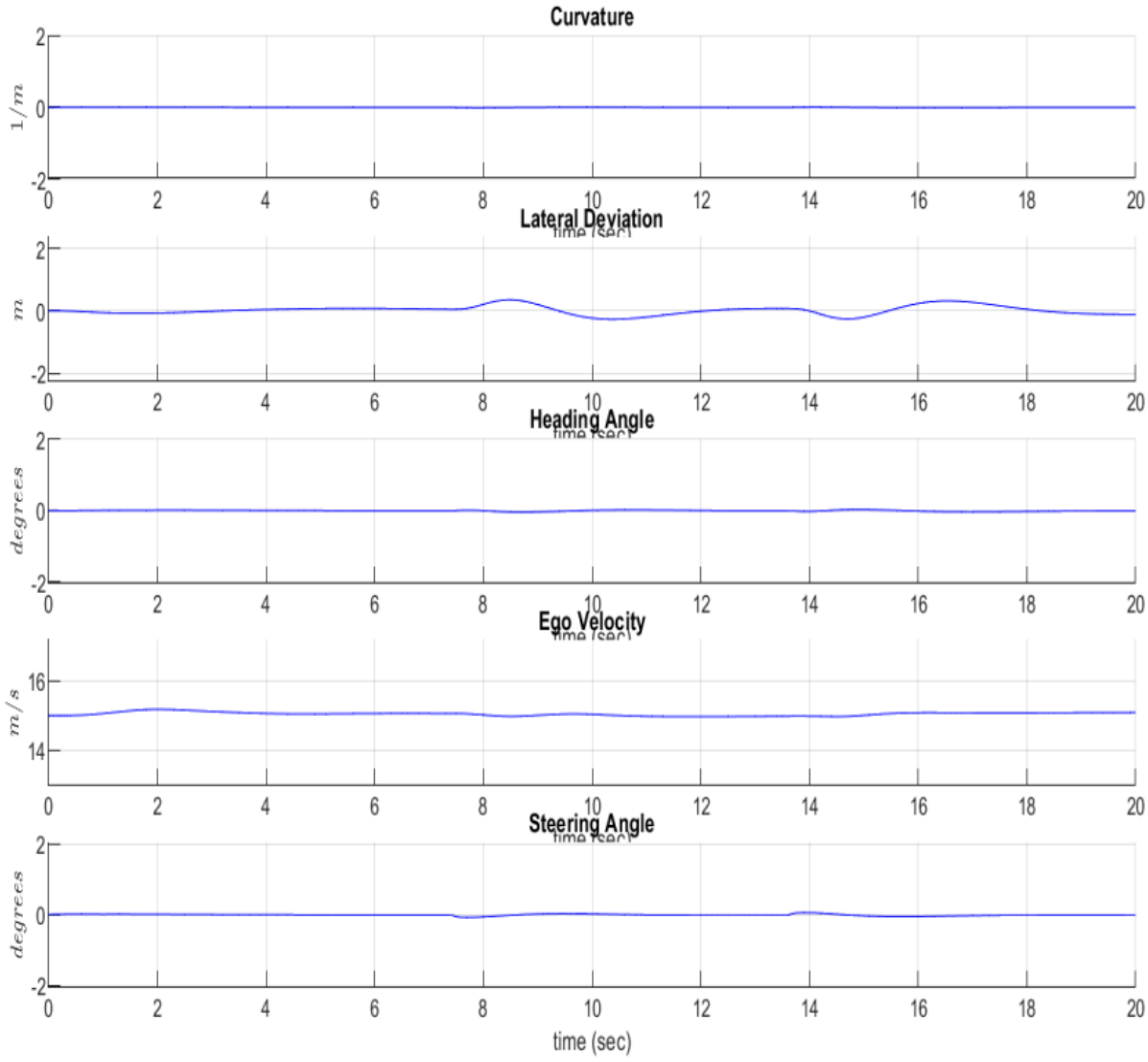

Figure 4.13. Controller performance

4.4 Summary

Ever since autonomous vehicles were invented, their features have increased from basic to standard to advanced features where vehicles can now make decisions in real time independent of a driver. This thesis expanded on the critical zone method by applying the equations from the original paper to different algorithms and traffic scenarios. The examples in [21] – [23] were used as base templates to understand how the systems worked before the equations, values and parameters and more adjustments to the test bench were made after calculations. The minimum value derived from calculating the braking time and steering time was set as the time gap for each scenario. Whenever a collision was detected, the simulation immediately stopped citing the reasons for the failure which validated the safety features implemented in the test benches. The results validated the theory from the paper as it was tested on different scenarios although numerical values were different due to different values for the length, width and simulation times used. In the future, this theory could be expanded further by adding testing different behavior models of both leading and trailing vehicles to see how the critical zones are modeled.

REFERENCES

[1] I. Micah. "Autonomous car." (2019), [Online]. Available: https://encyclopedia.pub/entry/2787.

[2] T. S. Anh Tuan Tran Noboru Sakamoto, "A combination of analytical and model predictive optimal methods for adaptive cruise control problem," Jul. 12, 2018. DOI: 10.1109/ANZCC.2018.8606602. [Online]. Available: https://ieeexplore.ieee.org/document/8606602.

[3] R. Chandra, Y. Selvaraj, Mattias, R. Kianfar, and N. Murgovski, "Safe autonomous lane changes in dense traffic," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2017.

[4] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, "Highly automated driving on highways based on legal safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 333–347, 1 2018. DOI: 10.1109/TITS.2012.2225104. [Online]. Available: https://ieeexplore.ieee.org/document/6361481.

[5] C. L. Stéphanie Lefèvre Dizan Vasquez, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal volume*, pp. 17–39, 1 2014. DOI: https://doi.org/10.1186/s40648-014-0001-z. [Online]. Available: https://robomechjournal.springeropen.com/articles/10.1186/s40648-014-0001-z.

[6] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 678–687, 2009. DOI: 10.1109/TITS.2009.2026452.

[7] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," pp. 417–422, 2009. DOI: 10.1109/ICCP.2009.5284727.

[8] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "Mpc-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, pp. 265–291, 2005.

[9] H. Okuda, K. Harada, T. Suzuki, S. Saigo, and S. Inoue, "Design of automated merging control by minimizing decision entropy of drivers on main lane," pp. 640–646, 2017. DOI: 10.1109/IVS.2017.7995790.

[10] J. Nilsson, J. Fredriksson, and E. Coelingh, "Trajectory planning with miscellaneous safety critical zones," *IFAC-PapersOnLine*, vol. 50, pp. 9083–9088, 2017.

[11] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 644–649, 2015.

[12] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "Mpc for tracking piecewise constant references for constrained linear systems," *Autom.*, vol. 44, pp. 2382–2387, 2008.

[13] B. Higgs, M. M. Abbas, and A. Medina, "Analysis of the wiedemann car–following model over different speeds using naturalistic data," 2011. [Online]. Available: https://onlinepubs.trb.org/onlinepubs/conferences/2011/RSS/3/Higgs,B.pdf.

[14] H.-J. Cho and W. YUH-TING, "A car-following model for intelligent transportation systems management," 2007. [Online]. Available: https://www.academia.edu/26895064/Modelling_Simulation_Methods_for_Intelligent_Transportation_Systems.

[15] H. Kita, K. Tanimoto, and K. Fukuyama, "A game theoretic analysis of merging-giveway interaction: A joint estimation model," 2002. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/9780585474601-025/full/html.

[16] Q. Zhang, D. Filev, H. E. Tseng, S. Szwabowski, and R. Langari, "A game theoretic four-stage model predictive controller for highway driving," *2019 American Control Conference (ACC)*, pp. 1375–1381, 2019.

[17] T. A. N. Heirung, J. A. Paulson, J. O'Leary, and A. Mesbah, "Stochastic model predictive control - how does it work?" *Comput. Chem. Eng.*, vol. 114, pp. 158–170, 2018.

[18] Q. Zhang, R. Langari, H. E. Tseng, D. Filev, S. Szwabowski, and S. Coskun, "A game theoretic model predictive controller with aggressiveness estimation for mandatory lane change," *IEEE Transactions on Intelligent Vehicles*, vol. 5, pp. 75–89, 2020.

[19] O. Pauca, C. F. Caruntu, and C. Lazar, "Predictive control for the lateral and longitudinal dynamics in automated vehicles," *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 797–802, 2019.

[20] J. Xue, Z. Huang, J. Zhou, Y. Chen, and S. Y. P. Chien, "A hierarchical clustering analysis (hca) in automatic driving regarding to vehicle-to-vehicle pedestrian position identification," 2017.

[21] Mathworks. "Highway lane following." (), [Online]. Available: https://www.mathworks.com/help/mpc/ug/highway-lane-following.html.

[22] Lumen Learning. "Watchdog controller for an electric vehicle - jaguar land rover ltd." (), [Online]. Available: https://www.sumobrain.com/patents/wipo/Watchdog-controller-electric-vehicle/WO2018036977A1.html.

[23] Mathworks. "Automated testing for highway lane change." (), [Online]. Available: https://www.mathworks.com/help/driving/ug/automate-testing-for-highway-lane-change.html.