CHALMERS
UNIVERSITY OF TECHNOLOGY

Human Behaviour- Based Trajectory Planning For Autonomous Overtaking Maneuver

Aditya S Murthy

Pavan S Bharadwaj

Department of Mechanics and Maritime Sciences
Vehicle Safety Division (CAP Unit)
Chalmers University of Technology
Gothenburg, Sweden 2020

Master's thesis 2020:71

Human Behaviour-Based Trajectory Planning for Autonomous Overtaking Maneuver

Aditya S Murthy

M.S. in Automotive Engineering

Department of Mechanics and Maritime Sciences

Pavan S Bharadwaj

M.S. in Systems, Control and Mechatronics

Department of Electrical Engineering

Department of Mechanics and Maritime Sciences (M2)

Division of Vehicle Safety

Crash Analysis and Prevention Unit

Chalmers University of Technology

Gothenburg, Sweden 2020

Human behaviour-based trajectory planning for autonomous overtaking maneuver
Aditya S Murthy
Pavan S Bharadwaj

Supervisor: Pinar Boyraz Baykas, Department of Mechanics and Maritime Sciences
Examiner: Pinar Boyraz Baykas, Department of Mechanics and Maritime Sciences

Master's Thesis 2020:71
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Overtaking maneuver simulation on CarMaker simulation platform.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Human behaviour based trajectory planning for autonomous overtaking maneuver

Aditya S Murthy
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Pavan S Bharadwaj
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Autonomous driving is a contemporary research area that bears great potential to improve safety and environmental sustainability in the automotive domain. Trajectory planning is an important layer in the hierarchical structure of such autonomous vehicle systems, and has been a topic of extensive research in recent years. This layer is responsible for generating a sequence of continuous states that guides the vehicle through its environment safely, comfortably and following traffic rules.

A majority of the currently used trajectory planning algorithms depend on a predetermined reference trajectory generated by GPS or high-resolution vector maps, which often are either unreliable, expensive or simply unavailable. The aim of this thesis is to develop a computationally robust trajectory planning algorithm for overtaking maneuver, that uses naturalistic driving data to generate a human like trajectory to navigate the vehicle through a dynamic environment. A Euler spiral and Markov Decision Process (MDP) based framework is developed that is used to solve the trajectory generation and selection problem. The generated reference trajectory is then input to a local feedback controller to actuate the steering of vehicle. The developed algorithm is implemented on a simulation platform to assess parameters such as safety, passenger comfort and maneuver execution time. The robustness of the algorithm is evaluated by simulating the overtaking maneuver for different velocity profiles of the target vehicle. An unsupervised learning based intent recognition model is proposed to identify and predict lane change intentions of surrounding vehicles, with the aim of further improving the safety of the autonomous vehicle in dynamic environments. The model is evaluated for accuracy against a simple Pseudo-Ground Truth (PGT) model, to validate the promptness of lane change prediction.

The overtaking maneuver simulation results demonstrated that the developed algorithm is able to successfully execute the maneuver in all of the test scenarios, maintaining sufficient clearances to the target vehicle and road boundaries while respecting passenger comfort limits. The trained intent recognition model is found to be prompt in issuing lane change warnings as a rear end collision is avoided in ten out of eleven test cases.

Keywords: Autonomous driving algorithm, Markov decision process, Field/naturalistic data, Hybrid dynamic system, Linear controller, Traffic micro simulations, Driver behaviour, Data processing

Acknowledgements

We would like to thank of examiner and supervisor, Pinar Boyraz Baykas for extending her constant support throughout the duration of this thesis. Her fresh outlook and sustained encouragement helped us gain clarity and kept us motivated during tough times. We would also like to extend our regards to the Department of Electrical Engineering and the Department of Mechanics and Maritime Sciences , the division of Vehicle and Traffic Safety at Chalmers for providing us with the excellent opportunity of working on this thesis. Finally, we would like to extend our gratitude to the team at RWTH Aachen for providing us with the High-D naturalistic data set, which has been an essential foundation for this thesis.

Göteborg 2020

Aditya Murthy, Pavan Bharadwaj

Contents

List of Figures

List of Tables

1
Introduction

For decades, autonomous navigation has been a subject of interest for academia and industries alike. Although still in their testing stages, many automobile companies have deployed prototypes of self-driving cars on the roads to verify their reliability and performance. These companies adopt similar underlying/supporting Advanced Driver Assistance System (ADAS) technologies such as Lane Departure Warning System (LDWS), Adaptive Cruise Control (ACC), Collision Avoidance, Pedestrian Crash Avoidance Mitigation (PCAM),incorporated traffic warnings, alert driver to approaching traffic or dangers and automatic lane centring integrated with several sensors and controllers [2] to ensure a safe operation of these autonomously navigated vehicles.

The problem area in the domain of autonomous mobility can be bifurcated into two major sub-domains. The first, perception and localization and second, path planning and control. In order to tackle the problem of perception and localization, Global Positioning System (GPS) is widely used in tandem with a variety of sensor modalities such as RADAR, LIDAR and Vision cameras. The data from a positioning system such as GPS/Differential GPS (DGPS) fused with perception data from other sensor modalities is generally used to estimate the position of the vehicle in a known/unknown environment. However, the accuracy of perception/location depends on environmental features such as dynamic or static objects, weather, etc. Hence, increasing the number of sensors and sensor modalities, employing better data processing and data fusion techniques affects the safety of these autonomous system positively [2].

The problem of perception and localization is a precursor to the second sub-domain of Path planning and Control, as the accuracy of the latter completely depends on the former [3]. Path planning can be further divided into global and local path planning based on the scale of planning. In global path planning, digital macro scale maps and simple geometric segments are used to generate a path that allows a vehicle to pass through a given set of way points. While global path planning deals only with finding the shortest/fastest path for the vehicle from the start to the destination, local path planning is concerned with finding a feasible trajectory for the vehicle navigating a dynamic environment while following that path.

These scales of path planning are analogous to a driver using a navigational map to get to the destination. The map represents a macro level/global path planning while the driver navigating the car and avoiding obstacles represents the local path planning. However, there are several technological issues that are commonly encountered during such path planning like,

- GPS based path planning suffers from signal interference, data reception failure, large errors from the ground truth. Common examples are GPS interruption or poor accuracy in a small part of high-rise buildings, tunnels and underground parking
- Another consideration is the human factors in path planning. Even if the path planning is completed at the point where the goal is reached with shortest travel length (fuel economy) or the minimum travel time, the path may deteriorate the comfort of human passengers, and not be optimal for the 'human' aspect of transportation [3].

1.1 Background

In recent years, significant progress has been made in the field of autonomous navigation. Regarding trajectory planning, a few of the most common methods are as described below.

The RRT algorithm (Rapidly exploring Random Trees) was developed by LaValle et. al. [4] which can handle problems with obstacles and differential constraints (nonholonomic and kinodynamic) and has been used widely in autonomous robotic motion planning. RRT generates open loop trajectories for non-linear systems with state constraints that are suitable for route planning problems that include obstacles and non-autonomous constraints. This method ensures kinematic feasibility and can be easily implemented in real time. However, in the presence of many obstacles or heavy traffic, this method will check for all possible collision instances for each extended node which leads to an explosion in computational complexity [5].

Another local trajectory planning approach is Lattice planners which is an extension of grid-based planners. It defines a state lattice that is discretized in all state parameters of interest, such as position, heading, curvature, and velocity. A trajectory generator is used to precompute feasible motions between states in the lattice in the absence of obstacles. The states define nodes in a graph, where edges describe motions. They are generally well-suited for non-holonomic and highly constrained environments, such as the road environment. Lattice planners are resolution complete. This means that the control space can automatically be adjusted for every resolution change and the space is explored consistently. Lattice planners also guarantee optimality and smoothness. Decision making for autonomous driving is a challenging task due to the uncertainty of the complex environment surrounding the vehicle. Partially Observable Markov Decision Process (POMDP) offers a framework for autonomous robot navigation in dynamic environments. With this approach, the state of a car's environment can be estimated and the development of traffic situations can be predicted. Unfortunately, the implementation of these methods requires expensive collection of training data. In this thesis, we will be developing an autonomous driving algorithm for trajectory planning with clothoid tentacles, to avoid obstacles and follow the reference trajectory, coupled with a decision process inspired from well known MDP (Markov Decision Process) [6].

1.2 Problem description

This master thesis investigates the possibility of developing a path planning algorithm for executing 'human-like' overtaking maneuvers in highly automated driving systems by integrating naturalistic driving data with the heuristics-based path planning algorithms. This work aims to implement such a path planning algorithm without using a reference trajectory, such as GPS or Vector Maps. The resulting trajectories from such a hybrid algorithm when validated against human driven trajectories from naturalistic data sets like High-D, would yield an interesting insight on how closely such algorithms can mimic human driving behaviour. To facilitate such an integration, a probabilistic framework will be formulated to represent naturalistic driving data which will then be incorporated into the Markov Decision Process based path planning framework.

The scope of this thesis also extends to exploring the feasibility of developing an intent recognition model using naturalistic driving data in the High D data set using the methods of unsupervised learning models like Gaussian Mixture Models. Such a model can then be used to predict the behaviour of surrounding traffic, which can prove to be useful in avoiding potential rear-end collision scenarios.

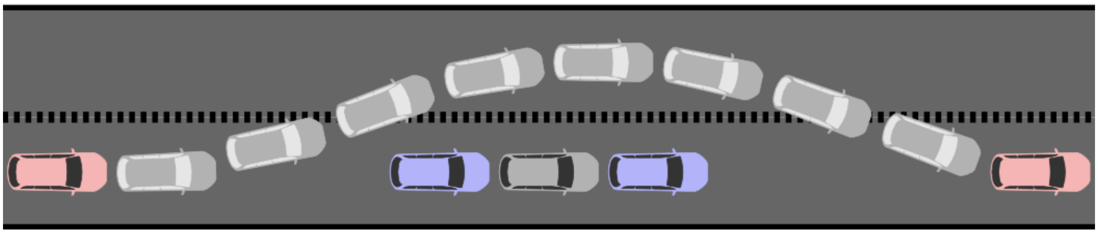


Figure 1.1: Overtaking Scenario

1.3 Purpose

The purpose of this thesis work is to study and develop algorithms for automated driving systems to plan and execute safe maneuvers and to predict the future trajectories for surrounding road users. Explicitly, such maneuvers can be modelled using a 'Hybrid Dynamic' system, which explores the possibility of combining continuous time regulatory control processes for modelling motion phases in the maneuver, with probabilistic decision-making processes featuring often discrete nature such as Finite State Machines and Markov Decision Processes. The model would then be validated against field/naturalistic data set, which may yield explanatory insights in distributions of model parameters across the population of drivers, while also representing the mean/median behaviour that might be used in system design.

1.4 Thesis Outline

The rest of the thesis is structured as follows:

- Chapter 2 Introduces all relevant theories used in the thesis
- Chapter 3 Presents the problem formulation approach and implementation
- Chapter 4 Establishes the results of the thesis work
- Chapter 5 Describes potential future developments for the thesis

2

Relevant theory

2.1 Clothoids

A clothoid, also known as Euler spiral is a geometric curve whose curvature changes linearly with its curve length. The mathematical representation of Euler spiral is expressed as follows [7].

$$\rho = \frac{2}{\kappa^2} s \tag{2.1}$$

where ρ represents the curvature of clothoid, κ is a constant and s is the curvilinear abscissa (length along the curve).

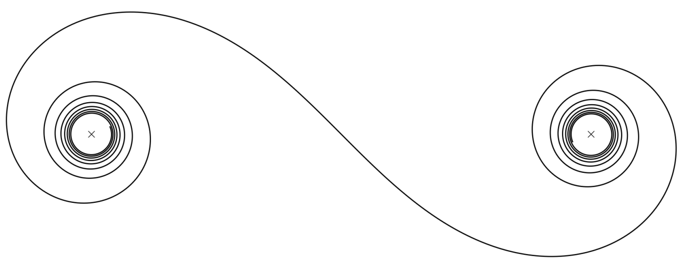


Figure 2.1: Euler spiral [1]

A set of such spirals generate a set of virtual 'antennas' known as tentacles in the ego-centered reference frame of the vehicle which model the dynamically feasible trajectories. The curvature of tentacles is the derivative of the tangent angles and are represented in the Cartesian co-ordinate using the following expressions known as Fresnel integrals [7].

$$\begin{aligned} x(s) &= x_0 + \int_0^s cos(\theta(t))dt \\ y(s) &= y_0 + \int_0^s sin(\theta(t))dt \\ \theta(s) &= \theta_0 + \int_0^s cos(\kappa(t))dt \end{aligned} \tag{2.2}$$

The initial clothoidal curve was discovered by Jacob Bernoulli in 1694, during his work in the field of elasticity. Later, it was reformulated by Augustin Fresnel in 1818 when he studied the problem of light diffraction and deduced the well-known Fresnel integral. Fresnel integrals were again derived by Arthur Talbot in 1890, this time in the context of transition bends for the designed railway tracks for a smooth ride [1].

A clothoid curve can simultaneously consist of segments of straight line $\rho = 0$ and circular arcs and are also commonly known as clothoid splines. Clothoid are used in railroad and highway engineering as transitional curves between straight line segment and circular arc. Spiral-based transitions are appealing, since the smooth transition in curvature produces a gradual change in the centrifugal force experienced by the vehicle and passengers which reduces the discomfort for the passenger and reduces the risk of accidents. However, a drawback of a clothoid based path planning is that feasible trajectories of the vehicle are viable without losing stability only up to a certain defined distance, as the increase in curvature of the curve is unbounded with length [1].

2.2 Single Track Vehicle Model

The single track model, also referred to often as Bicycle Model, is a simplified representation of a detailed vehicle dynamics model of a car, in which the left and right wheels of the vehicle are merged into a single wheels, connected by a rigid link as shown in Figure 2.2.

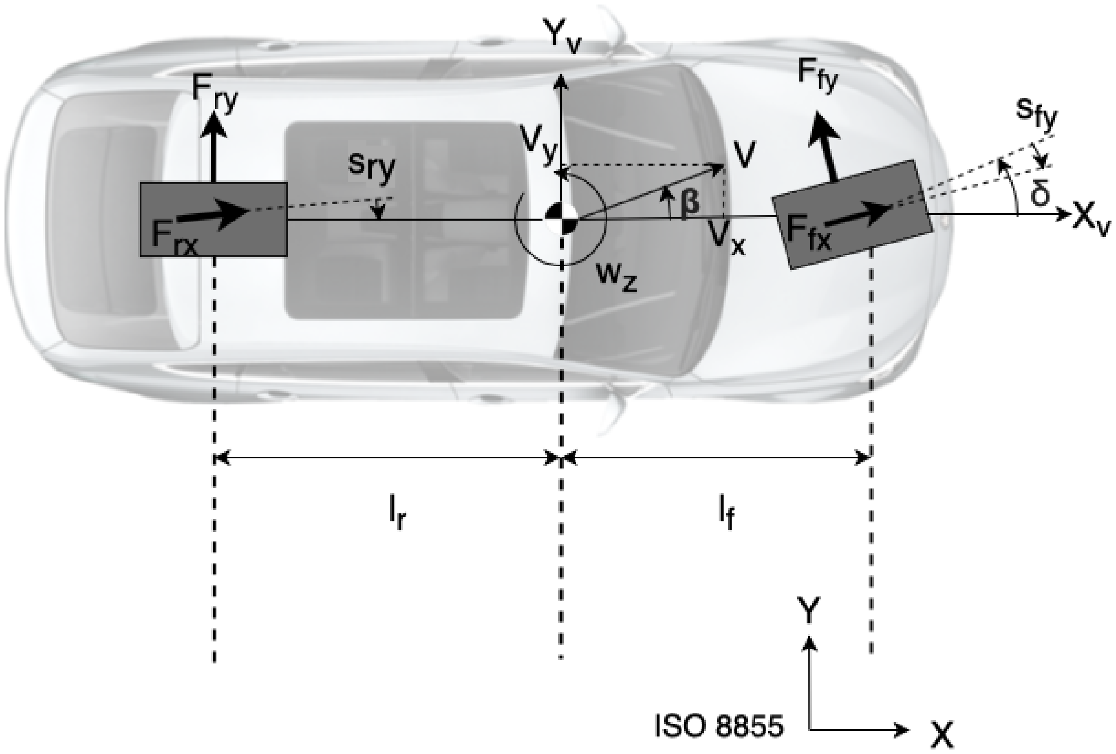


Figure 2.2: Single track vehicle model [2]

2.2.1 Notation

X_v: Egocentric X coordinate

Y_v: Egocentric Y coordinate

V: Resultant velocity vector of the vehicle

V_x: Velocity in egocentric X coordinate

V_y: Velocity in egocentric Y coordinate

β: Side slip angle

ω_z: Vehicle yaw rate

δ: Steering angle at wheel

s_{fy}: Front axle slip angle

s_{ry}: Rear axle slip angle
F_{fx}: Front axle longitudinal force
F_{rx}: Rear axle longitudinal force
F_{fy}: Front axle lateral force
F_{ry}: Rear axle lateral force
F_x: Net longitudinal force in vehicle co-ordinate F_y: Net lateral force in vehicle co-ordinate F_{fz}: Front axle longitudinal
l_f: CoG distance to front axle
l_r: CoG distance to rear axle
m: mass of the vehicle
J: Moment of inertia of the vehicle about the Z-axis
L:vehicle wheelbase
C_f: Front axle cornering stiffness
C_r: Rear axle cornering stiffness

2.2.2 System dynamics

The single track vehicle model addresses the problem of vehicle motion estimation using three degrees of freedom, namely longitudinal movement, lateral movement and yaw around the z-axis (ISO 8855). The vehicle in this case represents a non-holonomic system [8] as the vehicle's motion can not be represented using closed-form solution of the form,

$$f(q_1, q_2, ..., q_n, t) = 0 \tag{2.3}$$

where, $\{q_1, q_2, ..., q_n\}$ represent the set of co-ordinates needed for completely describing the vehicle's motion. Hence, the dynamic system is defined using velocity constraints using ordinary differential equations as in equation 2.4. In other words, the history of the vehicle in the configuration space is essential to calculate its final configuration.

$$f(q_1, q_2, ..., q_n, \dot{q_1}, \dot{q_2}, ..., \dot{q_n}, t) = 0 \tag{2.4}$$

2.2.3 Forces

$$\begin{aligned}
F_{fz} &= \frac{mgl_r}{L} \\
F_{rz} &= \frac{mgl_f}{L} \\
F_x &= F_{fx}cos(\delta + s_{fy}) - F_{fy}sin(\delta + s_{fy}) + F_{rx}cos(s_{ry}) \\
F_y &= F_{fy}cos(\delta + s_{fy}) + F_{fx}sin(\delta + s_{fy}) + F_{ry} + F_{rx}sin(s_{ry})
\end{aligned} \tag{2.5}$$

The lateral forces on each of the axles can be represented using equation 2.6.

$$\begin{aligned}
F_{fy} &= C_f * s_{fy} \\
F_{ry} &= C_r * s_{ry}
\end{aligned} \tag{2.6}$$

For low values of slip angles, the linear tyre model is a good approximation. However, for high speed maneuvers, the relation between the cornering stiffness and normal load is estimated using the equation 2.7.

$$\begin{aligned}
C_f &= C_0 * F_{fz} + C_1 * F_{fz}^2 \\
C_r &= C_0 * F_{rz} + C_1 * F_{rz}^2
\end{aligned} \tag{2.7}$$

where, C_0 and C_1 are estimated tyre parameter constants. This non-linear model is a better approximation to the actual tyre model as represented in [9].

2.2.4 Tyre slip

The front and rear slip angles for small steering angles are estimated using equation 2.8

$$\begin{aligned}
s_{fy} &= \frac{-(v_y - \delta v_x + l_f\omega_z)}{v_x} \\
s_{ry} &= \frac{v_y - l_r\omega_z}{v_x}
\end{aligned} \tag{2.8}$$

2.2.5 State variables

The change in the state of the system is given by equation 2.9

$$
\begin{aligned}
\dot{v_x} &= \frac{F_{fz}cos\delta - F_{fy}sin\delta + F_{ry}}{m} + w_z v_y \\
\dot{v_y} &= \frac{F_{fx}sin\delta + F_{fy}cos\delta + F_{ry}}{m} - w_z v_x \\
\dot{w_z} &= \frac{(F_{fx}sin\delta + F_{fy}cos\delta)l_f - F_{ry}l_r}{J}
\end{aligned}
\tag{2.9}
$$

2.3 Occupancy Grid

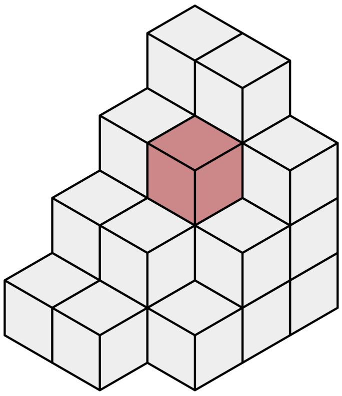
Occupancy grid basically represents the environment as uniformly spaced field of binary random variables where each binary variable represents the presence of obstacles at that location in the environment.

Occupancy grid mapping belongs to family of computer algorithms in probabilistic mobile robots to solve the problem of generating maps of an environment from a noisy and uncertain sensor measurement data of that environment. The algorithm calculates the posterior estimation for the random variables describing the presence of an obstacle in the environment at discrete points in the environment[10].

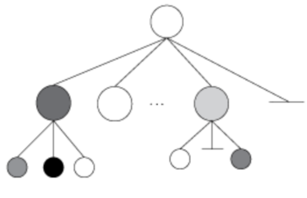
An occupancy grid can be created with different sizes and resolutions to fit the application. The information of the environment is a loaded information from prior knowledge(i.e pre-recorded sensor information) or extracted from the sensor in real time. Most commonly used sensors to locate the obstacles in the environment are camera, LIDAR, RADAR, depth sensors and LASER Range Finder(LRF). In general occupancy grid are of two kinds 3D and 2D occupancy grids.

2.3.1 3D Occupancy Grid

In a 3D occupancy map, the information is stored in a 3D map as probabilistic values in an octree data structure. Basically, an Octree is a hierarchical data structure which divides the environment into cubic volume called voxels. For a given environment, the space is recursively divided until the desired map resolution is achieved. The divisions are expressed as a tree which stores the probability value of the location in the map as shown in figure 2.3a.



(a) 3D Voxel



(b) Voxel Tree

Figure 2.3: 3D Occupancy Map

2.3.2 2D Occupancy grid

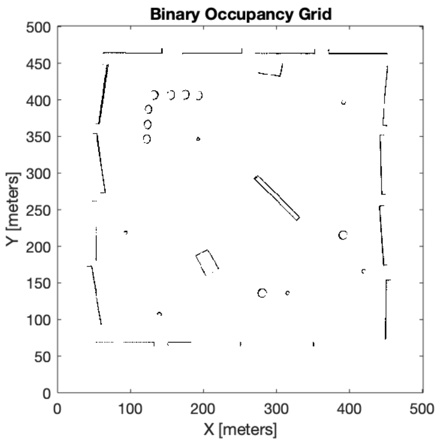
2D occupancy grid are the most common type of occupancy grid that describes a slice of the 3D world. 2D occupancy grids are of two kinds namely binary occupancy grids and probability occupancy grids.

2.3.2.1 Binary occupancy grid

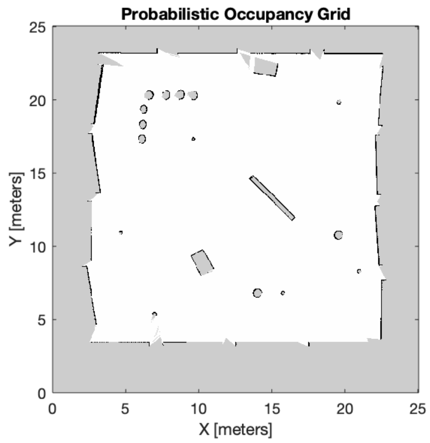
In a binary occupancy, one or true(boolean) value represents an obstacle in the environment, and a zero or false value represents an unoccupied space in the environment as shown in 2.4a.This method is the most suitable when the application has strict constraints on computational memory.

2.3.2.2 Probability occupancy grid

Probability occupancy graphs use probability values ranging from [0,1] to create environments [10]. In this case, each cell in the occupancy grid has a value, which represents the probability of that cell being occupied. The probability values which are close to unity are more probable to be occupied, those that are close to zero are less likely to be occupied and those cells with a probability of 0.5 are cells with uncertain measurements/occlusions as shown in 2.4b. Probability values can provide better object fidelity which might improve the performance of algorithms in certain applications.



(a) Binary Occupancy



(b) Probabilistic Occupancy

Figure 2.4: 2D Occupancy Map

2.4 Sensors

Sensors are devices, machines or subsystems used to detect events or environmental changes and often transmit information to be processed. Over the past three decades, the use of sensors in driverless vehicles has increased and their reliability has made significant progress. The different types of sensor modalities used in driverless vehicles are briefly covered here:

2.4.1 LIDAR

LIDARs (Light Detection and Ranging) are light based range detection sensors that have found increasing applications in self driving vehicles [11]. LIDARs can be classified based on a number of features, an important one of which is the sweep angle. The most commonly used LIDARs have a sweep angle of 180 or 360 degrees. LIDAR is usually a continuously rotating/oscillating system that can send out thousands of laser pulses per second. These pulses collide with the surrounding objects and reflect back as point clouds, which are then used to construct a 3-D environment using image analysis techniques running on on-board computers. LIDAR makes use of the speed of light and the time of flight, to determine the distance to all the surrounding objects and thus can monitor constant developments in highly dynamic environments.

2.4.2 RADAR

RADAR or Radio Detection and Ranging use radio frquencies in order to scan the surrounding environment, detect objects from a long distance and determine their speed and disposition. In RADAR, the radio waves from the transmitter reflect off the object and return to the receiver, giving information about the object's location and speed. The advantage of RADAR is that it has a long working distance and is more reliable in adverse weather conditions such as fog and rain. However, compared to other sensors, RADARs have low object resolution and can lead to incorrect data representation of the environment.

2.4.3 Cameras

2.4.3.1 Stero Camera

A stereo camera is a pair of cameras that can use triangulation to reconstruct depth from two simultaneously captured images. Usually, this configuration requires additional synchronization and calibration [11].

2.4.3.2 Monocular Camera

The camera is a passive sensor system that projects the visible segment of the electromagnetic spectrum onto the electronic sensor plane to record environmental images [11].

2.4.3.3 Infrared Camera

An infrared or thermal imaging camera is very similar to a conventional camera, but it replaces visible light and detects infrared radiation, so it can visualize the surface temperature of an object [11].

Most of the self driving cars use cameras as it is the most accurate way to represent the environment. Self-driving cars rely on cameras installed on the front, back, left, and right sides to view 360 degree of the surrounding environment. Some have a wide field of view-up to 120 degrees-and a shorter focal length while others focus on a narrower field of view to provide long-range visuals. A camera can provide accurate visualization, but has limited accuracy in low visibility conditions such as fog, rain or at night in which cases they might fail to distinguish objects from the environment.

2.5 Sensor Fusion

The high resolution on-board sensors on an autonomous vehicle provide a good representation of the environment. However, each sensing modality inherently reports uncertain measurements. Sensor fusion is a mathematical framework that aims to combine the measurements from different sensors in order to lower the uncertainty of state estimation, as compared to a single sensor. The sensor fusion algorithms can be broadly classified into linear and non-linear filters.

2.5.1 Linear Filters

The linear filters such as Kalman Filter [12] also known as linear quadratic estimators, work on the assumption that the system under consideration is linear with Gaussian noise. These filters are very robust and consume less memory, however, they are less flexible and give away in accuracy as the systems increasingly deviate from the linearity assumption.

2.5.2 Non-linear Filters

Unlike linear filters, non-linear filters such as Particle filters [13], Unscented Kalman Filters (UKF) and Cubature Kalman Filter(CKF) [14] try to estimate the state distribution using a large number of samples (also called particles) instead of a Gaussian distribution. The result is increased flexibility and state estimation accuracy but at the expense of memory and computational complexity.

2.5.3 Kalman Filter

Kalman Filter is a recursive Maximum a posteriori(MAP) estimators that try to maximize the posterior distribution for the unknown state parameters. These estimators are very similar to Maximum Likelihood Estimators(MLE), with the only difference that MAP estimators are weighted with prior distribution of the parameters as shown in equation 2.10

$$
\begin{aligned}
\hat{\theta}_{MAP}(x) &= argmax_{\theta} f(x|\theta) \\
\hat{\theta}_{MLE}(x) &= argmax_{\theta} f(x|\theta) g(\theta)
\end{aligned}
\tag{2.10}
$$

where $f(x|\theta)$ are the $g(\theta)$ are the likelihood and prior distributions respectively. The Kalman Filter estimates the state parameters in two steps namely, prediction and update step. In the prediction step, the prior information of the state and the process model of the system is used to estimate the state at current time step as shown in equation 2.11

$$
\begin{aligned}
\hat{x}_{k|k-1} &= A_k \hat{x}_{k-1|k-1} + u_k \\
\hat{P}_{k|k-1} &= A_k \hat{P}_{k-1|k-1} A'_k + Q_k
\end{aligned}
\tag{2.11}
$$

where $\hat{x}_{k|k-1}$ is the predicted state vector calculated from the posterior distribution $\hat{x}_{k-1|k-1}$ at time $k-1$ (prior at time k) using the process/motion model A_k and gaussian noise u_k. The corresponding covariance is estimated in a similar fashion as shown in equation 2.11. The update step of the Kalman filter uses the predicted state and the available measurement to update the state estimate. This step is represented mathematically in equation 2.12.

$$
\begin{aligned}
y_k &= H_k x_k + v_k \\
K_k &= P_{k|k-1} H' (H_k P_{k|k-1} H'_k + R_k)^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k (y_k - H_k x_{k|k-1}) \\
\hat{P}_{k|k} &= (I - K_k H_k) \hat{P}_{k|k-1}
\end{aligned}
\tag{2.12}
$$

y_k represents the noisy measurement calculated from the observation/measurement matrix H_k, which maps the true states to the observations and the Gaussian measurement noise v_k. The Kalman gain K_k is a parameter that in essence dictates the level of 'confidence' that is distributed between the prediction estimate and measurement estimate of the state, based on the covariance values of the corresponding estimates. The updated state and covariance estimates are then calculated as shown in in equation 2.12. Figure 2.5 represents the Kalman iterative linear filtering algorithm.

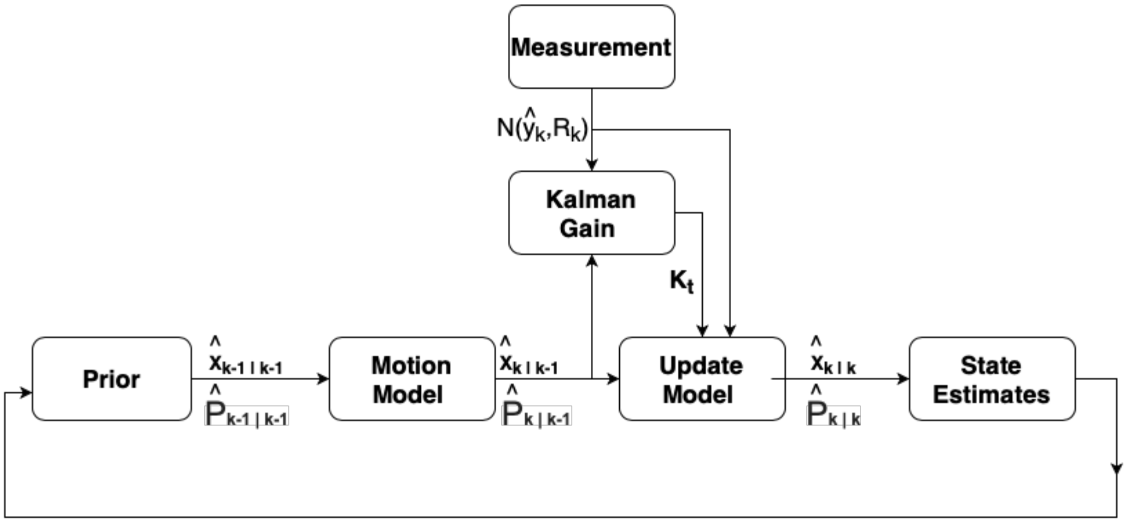


Figure 2.5: Kalman filter

2.6 Markov Decision Process

Markov Decision Process (MDP) is a mathematical framework for modelling decision making process in uncertain environments. MDPs are finding applications in an increasingly large variety of domains, ranging from automation and robotics to epidemiology [6].

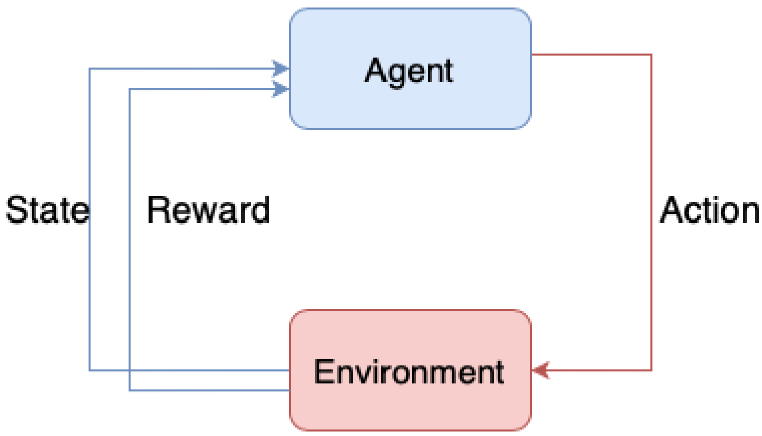


Figure 2.6: MDP Framework

The MDP can be defined using S, A, P, R, γ, where

- S: Finite set of states
- A: Finite set of actions

- T: State transition probability matrix $T(s, s') = P(s'|s, a)$, which represents the probability of the agent being in state $s' \in S$ after taking action $a \in A$
- R: $R_a(s, s')$ is the immediate reward received by the agent as a result of transitioning from state $s \in S$ to $s' \in S$ due to action $a \in A$
- γ: Discount factor that is used to assign different weights to current and future rewards.

The main objective of solving a MDP is to find a policy π that maps the set of states S to actions A, such that following this policy $\pi(a|s)\ \forall\ a \in A, s \in S$, would maximize the expected discounted sum of rewards, or consequently minimize the expected discounted sum of costs over a potentially infinite horizon given by equation 2.13.

$$E[\sum_{t=0}^{H} \gamma^t R_{a_t}(s_t, s_{t+1})] \tag{2.13}$$

where, H is the planning horizon. Solving equation 2.13 would yield an optimal policy that would guide the agent to take a specific action given a specific state. The solution to the above equation can be found using well known methods such as value iteration and policy iteration algorithm. Although computationally the policy iteration algorithm converges in fewer number of iterations, individual iterations are computationally expensive. Moreover, if the action set is small, the difference in iterations between the two algorithms becomes less significant. For this reason, value iteration algorithm was chosen for this work and is explained further.

2.6.1 Value Iteration

The value function $V^\pi(s)$, represents how good it is for the agent to be in state **s**, following a policy π. The value function is evaluated using the immediate reward that the agent receives starting from state **s**, following the policy π as shown in equation 2.14.

$$V^\pi(s) = E[\sum_{t=0}^{H} \gamma^i R_i]\ \forall s \in S \tag{2.14}$$

The optimal value function, V^*, is then obtained by evaluating over all the possible policies and choosing the maximum.

$$V^*(s) = \max_\pi V^\pi(s)\ \forall s \in S \tag{2.15}$$

The optimal policy π^* that maximizes equation 2.15 is called the optimal policy. The value function for each state can be defined more effectively using the Q-function.

The Q-function, Q(s,a) is representative of how good it is for the agent to pick an action ***a***, when in state ***s***. The relationship between the value function and the Q-function can be represented as shown in equation 2.16

$$V(s) = \max_a Q(s,a)\ \forall s \in S \tag{2.16}$$

Using equations 2.14, 2.15, 2.16 the optimal Q-function can be formulated as a recursive function using the Bellman optimality equation 2.16.

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V^*(s') \tag{2.17}$$

Since the optimal value function is related to the optimal Q-function as,

$$V^*(s) = \max_a Q^*(s,a) \tag{2.18}$$

the optimal Q- function provides the best action ***a*** for the agent in state ***s***.

2.7 Gaussian Mixture Models

A Gaussian mixture model is a probabilistic model that assumes that all the data points in a given population are generated from a mixture of finite number of Gaussian distributions.The model can be viewed as a generalization of k-means clustering to integrate information about the data covariance and potential Gaussian centers.

The Gaussian mixture object implements Expectation Maximization(EM) Algorithm to fit Gaussian distributions to the data. It can draw confidence ellipsoids of multivariate models and calculate Bayesian Information Criterion (BIC) to evaluate the most suitable number of clusters in the data.

k-means Algorithm:

- For each data point, determine the nearest cluster center.
- Each cluster center is replaced by the cooridinate-wise average of all data points that are closest to it.
- Repeat steps 1 and step 2 until the algorithm converges to the local minimum of the sum of squares within the cluster.
- Usually, multiple runs are used based on random initial guesses and the solution with the lowest within the sum of squares of the cluster is selected.

EM Algorithm:

- Start with the initial parameters guesses of $\hat{\theta}^(0)$

- Expectation Step: at jth step compute

$$Q(\theta', \hat{\theta}(j) = E(l_0(\theta' : t)|z, \hat{\theta}(j)))$$

- Maximization Step: Determine the new estimate $\hat{\theta}(j+1)$ as the maximization of $Q(\theta', \hat{\theta}(j))$ over θ'
- Repeat steps 2 and 3 until convergence.

2.8 Naturalistic Dataset

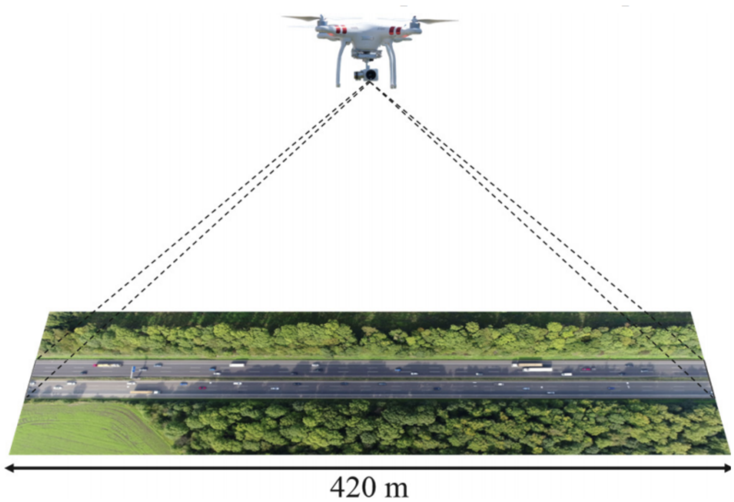


Figure 2.7: Unmanned aerial vehicle recording the highway stretch of 420m in Germany form a bird's eye view

The HighD dataset is a naturalistic vehicle trajectory dataset recorded on German highways. The data is recorded using a unmanned aerial vehicle at six different locations near Cologne Germany [15]. There are 60 recordings , with an average time of 17 minutes, covering a highway stretch of about 420m as shown in 2.7. On an average each vehicle is visible for about 13.6 s as mentioned in [15] and each location is classified based on total number of lanes and speed limits.

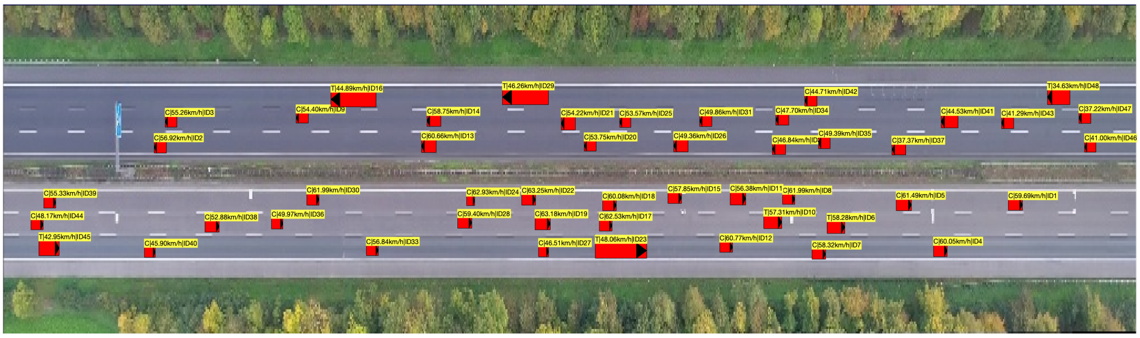


Figure 2.8: Example of HighD data for 25th recording representing class of vehicle,Vehicle ID and velocity in (km/h)

The data set includes a total of four files, including highway images and track meta data information and track csv files.The "recordingMeta.csv" file provides a general overview, such as vehicle ID, frameRate, recording time, highway section notes, and total number of vehicles recorded as shown in figure 2.8, Track meta data "trackMeta.csv" provides track information such as track ID, width, height and vehicle class. The "track.csv" contains time related values for each track including information such as current speed,position, velocity, viewing range and information about surrounding vehicles.

3

Approach and Implementation

The trajectory planning problem in dynamic environments can be formulated as a time parametrized function $\Psi(t) : [0, T] \longrightarrow \zeta$, describing the evolution of the vehicle state Ψ in its configuration space ζ. The vehicle configuration space ζ can be divided into allowed space Ψ_{free} and goal space Ψ_{goal}. Given a cost function $F(\Psi)$ and the dynamic constraint set $D(\Psi(t))$, the problem can be formulated as

$$\begin{aligned}
\underset{\Psi}{\operatorname{argmin}} \qquad & F(\Psi) \\
& \textit{subject to}, \\
\Psi(0) \in \zeta,\ & \Psi(T) \in \zeta \\
\Psi(t) \in \zeta \quad & \forall t \in [0, T] \\
D(\Psi(t)) \ & \forall t \in [0, T]
\end{aligned} \tag{3.1}$$

The trajectory planning algorithm developed in this work,for overtaking manuever, attempts to solve this optimization problem in equation 3.1 using a best first search algorithm, along with a Markov Decision Process (MDP) based heuristic cost evaluation function. The dynamic constraint problem is formulated as an unconstrained optimization by introducing penalty/cost functions. The algorithm is decomposed into modules as shown in figure 3.1. In the following sections, each module's design and implementation is discussed. Automated Driving Toolbox™ and MATLAB® was used for the design, implementation and testing of the algorithm developed in this work.

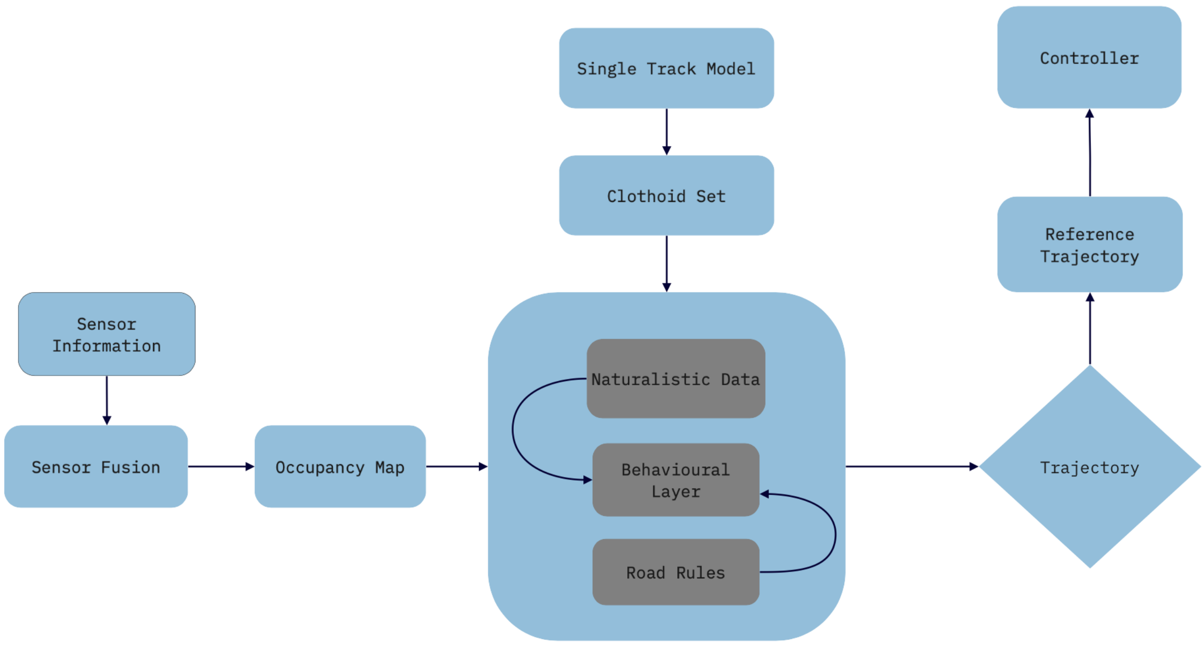
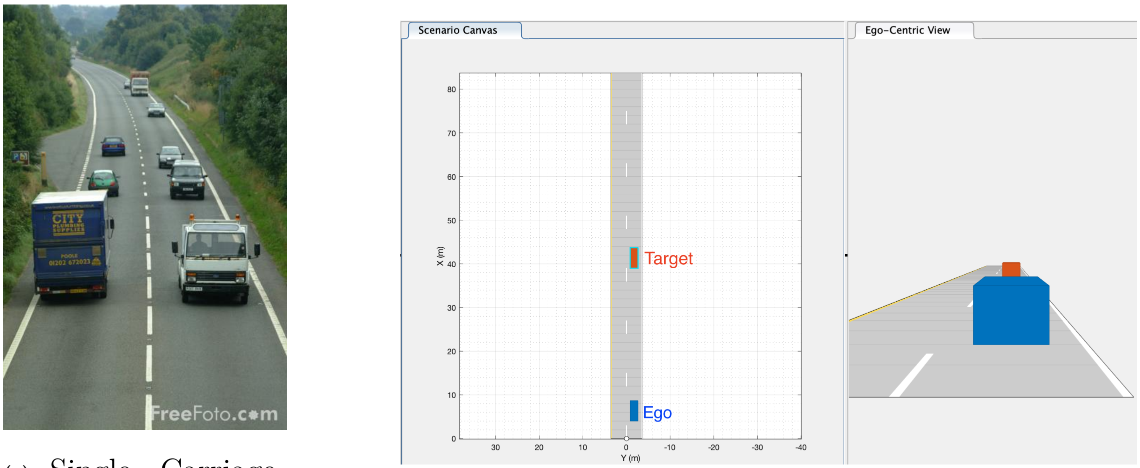


Figure 3.1: Trajectory planning modularization

3.1 Perception

The perception module's main function is to generate an accurate representation of environment around the ego vehicle, using sensors. The ground truth and the sensors were setup as a 'scenario' using the Automated Driving Toolbox™. The scenario was initialized to a single carriageway (undivided highway),with two lanes as shown in figure 3.2. A single carriageway can have multiple lanes with no median strip/central reservation between the lanes. The width of lanes is set to 3.5m, in accordance to road standards mentioned in [16]. In order to keep the complexity within limits, the following assumptions about the scenario were made:

- This work will focus on overtaking maneuvers on straight roads. However, it can easily be extended to roads with curvature with small modifications.
- There exists sufficiently large headway between the target vehicle and the vehicle preceding the target vehicle, for the ego vehicle to complete the overtaking maneuver. Hence, only the target and ego vehicle will be considered in the scene.
- The target vehicle is moving with a constant velocity in the longitudinal direction.



(a) Single Carriage-way

(b) Scenario in Automated Driving Toolbox™

Figure 3.2: Scenario Setup

A total of four sensors are used to form a perception of the environment. The sensor setup of the ego vehicle is shown in figure 3.3.

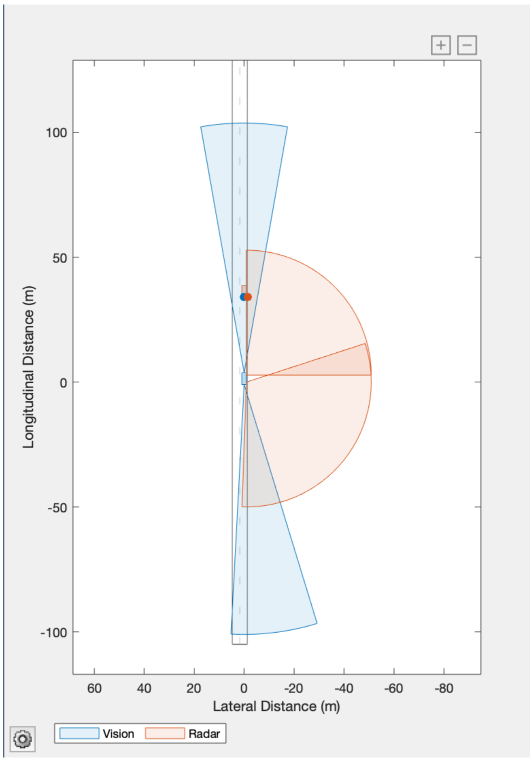


Figure 3.3: Sensor Setup

The specifications of the long range cameras and the radar are tabulated in tables 3.1 and 3.2.

Parameter	Value
Max. range	100m
Focal length	1814.81m
Max. detection speed	50m/s
Detection setting	Objects and Lane markings
Detection Probability	0.9
Update frequency	100 Hz

Table 3.1: Vision sensor parameters

Parameter	Value
Max. range	50m
Field of view[Azimuth Elevation]	$[90^o, 5^o]$
Azimuthal resolution	4^o
Max. range rate	100 m/s
Detection Probability	0.9
Update frequency	100Hz

Table 3.2: Radar sensor parameters

Owing to the constant velocity assumption for the target vehicle, a linear Kalman filter with constant velocity motion model was used to fuse the information from the cameras and radars to generate an occupancy grid. The discrete time constant velocity Kalman filter's motion model can be represented using equation 3.2.

$$\begin{bmatrix} x_{k+1} \\ v_{x,k+1} \\ y_{k+1} \\ v_{y,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_{x,k} \\ y_k \\ v_{y,k} \end{bmatrix} + a_w \tag{3.2}$$

where, $a_w = N(0, \sigma)$ models the acceleration as a Gaussian white noise. An ego-centric binary occupancy map was used to develop the algorithm, which was updated at sensor update frequency using the fused data from the Vision and RADAR sensors as shown in figure 3.4. The occupancy map was then converted into a cost

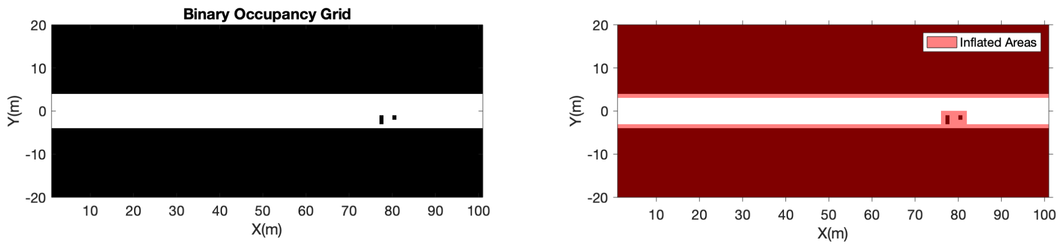


Figure 3.4: Occupancy grid and Cost map

map. Similar to the occupancy grid, a cost map represents the environment in the framework of discrete cells, with each cell bearing a cost in the range $[0, 1]$, which represents the cost of passing through that cell. Hence, an occupied grid cell on the occupancy map would correspond to a cell with a high cost value on the cost map. The cost map was inflated with a radius of 0.9m , meaning that all the unoccupied cells within a radius of 0.9 m of the occupied cell would also receive a high/inflated cost. Hence, the cost map provides a more informative view of the planning space around the vehicle as compared to binary occupancy values of the occupancy map.

3.2 Clothoid Set

The trajectory planning algorithm requires a set of potential paths to evaluate over a heuristic function. The clothoid set {C}, is a set of Euler spirals(clothoids) of equal length L, in which each clothoid is defined by its end radius R_L. There are two important properties of these curves that were used for selecting the end radii of the clothoids in the set.

- The curvature of a clothoid always increases linearly from $k_0 = 0$ ($R_0 = \infty$) to a final value of $k_L(R_L)$.
- A vehicle travelling along a clothoid experiences constant lateral acceleration. Since the ego vehicle is moving with a constant velocity, this property was used to determine the lower bound for the end radii R_{lim} of the clothoids in

the set as shown in equation 3.3

$$
\begin{aligned}
F_y &= \frac{mv^2}{R} \\
m * a_y &= \frac{mv^2}{R} \\
R_{lim} &= v^2 / a_{y,lim}
\end{aligned}
\tag{3.3}
$$

where, $a_{y,lim}$ is the maximum permissible lateral acceleration experienced by the passengers of the car. For the purpose of this work, the limit was set to $a_{y,max} = 1.0 m/s^2$ [17].

The clothoid set module uses the single track vehicle dynamics model in order to generate a set of Euler spirals (clothoids), which are then further evaluated for feasibility as potential path choices. The single track model is expressed as a function in 3.4.

$$[X, Y, \theta] = f(\delta, [X_0, Y_0, \theta_0], [v_{x,0}, v_{y,0}, w_{z,0}]) \tag{3.4}$$

where, δ, $[X_0, Y_0, \theta_0]$ and $[v_{x,0}, v_{y,0}, w_{z,0}]$ are the steering input, initial pose and initial state of the vehicle respectively. Assuming the ego vehicle has neutral steer property, i.e. the under steer coefficient ($K_u = 0$), the steering input required to follow the i^{th} clothoid in set {C}, with initial radius $R_0 = \infty$ and end final radius R_L is given by equation 3.5

$$\delta_i = tan^{-1} \frac{L}{r} \;\; \forall \, r \in [\infty, R_{L_i}] \tag{3.5}$$

where, R_{L_i} is the end radius of the i^{th} clothoid in {C}. As the radius decreases linearly along the length of the clothoid, the steering input vector δ_i in equation 3.5 consists of linearly increasing steer commands. At a given time step, the clothoid set C, generated using the above method is shown in figure 3.5

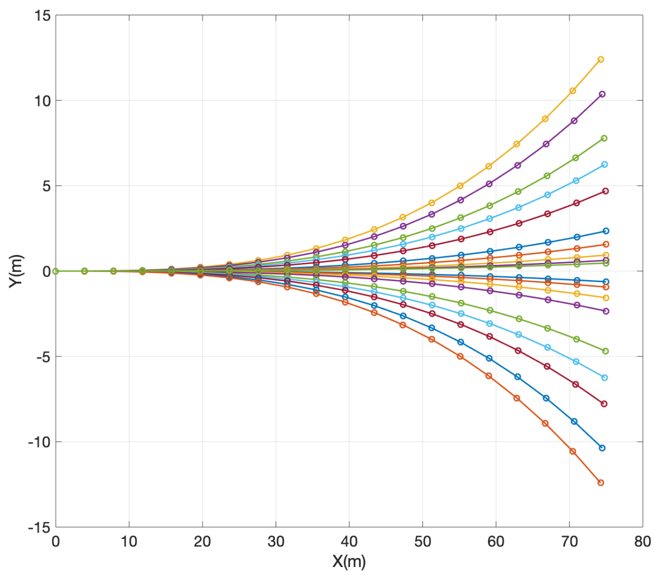


Figure 3.5: Clothoid Set {C}

3.3 Trajectory Selection

For selecting a trajectory among a set of clothoid trajectories {C}, a heuristic function was developed. A heuristic search function is an iterative search function that aims to provide a good enough solution to an optimization problem given enough time and memory. In other words, the heuristic function does not guarantee a globally optimum solution, however, it provides an acceptable solution compromising accuracy for speed and memory.

A significant amount of research has been devoted to studying the complexity of motion planning algorithms. Time and memory complexities of motion planning algorithms have been the focus of these studies [18]. Canny and Reif [19] established that the problem of finding a velocity-bounded, collision-free trajectory for a holonomic robot amidst moving polygonal obstacles falls under the NP-hard complexity group. The presence of non-holonomic constraints in vehicle motion estimation, further adds to the complexity. In an effort to find the solution to such problems in an acceptable amount of time, a number of heuristic search methods have been studied and developed [20]. Some of the well-known algorithms that use heuristic search methods are A^*, Best-First Search, *Hybrid* $A*$ and genetic algorithms. This section describes one such algorithm developed on the principle of Markov Decision

Process and Bellman optimality.
The process of selecting an execution trajectory begins with the evaluation of the trajectories in {C}. The heuristic cost function developed in this work is an aggregate of four cost functions, making it a multi objective optimization problem. While there are numerous ways to formulate and solve a multi objective optimization problem, the two most popular methods that do not require non-linear mathematical solvers are, Pareto and scalarization [21].

$$F(x) = \sum_{i=1}^{N} w_i f_i(x) \mid \quad \sum_{i=1}^{N} w_i = 1 \tag{3.6}$$

The normalized weight w_i of an objective function $f_i(x)$ in equation 3.6, depicts the importance of that objective function when compared to another. In other words, a higher weight assigns more importance to an objective function. In the scalarization technique, determining the weights of the objective functions plays a crucial role in obtaining an acceptable set of candidate solutions. For this reason, four formal methods, namely, Delphi method, Rank Order Centroid (ROC) method, Ratio method and Pairwise Comparison method were evaluated [22]. The Pairwise Comparison method was selected for further development, as it offered exhaustive comparison between objective functions when compared to the other methods.

3.3.1 Pairwise Comparison Method

The individual objective functions for the overtaking maneuver are defined as follows:

- Occupancy Cost (f_{occ}/OC): Normalized cost of occupying a particular point in the environment based on the generated occupancy cost map
- Yaw Rate Cost (f_{YR}/YRC): Normalized cost of a path based on the maximum yaw rate experienced by the vehicle while traversing the path
- Heading Cost (f_{head}/HC): Normalized cost of moving to a particular point in the environment based on the heading of the vehicle w.r.t the road's longitudinal direction.
- Lane Change Cost (f_{LC}/LCC): Normalized cost of changing lanes

Each of the cost functions mentioned above was compared to the rest of the functions in the group and a preferential score was given to each pair. Mathematically, in a pairwise evaluation matrix P, the element {i,j} indicates the relative importance of the i^{th} function evaluated against the j^{th} function as shown in table 3.4. For a consistent pairwise score reference table 3.3 was used as mentioned in [22].

Score	***Description***
10	The candidate function is of highest importance and much more important than the function being compared against
8	The candidate function is strongly important than the function being compared against
6	The candidate function is more important that the function being compared against
4	The candidate function is slightly more important than the function being compared against policies
2	The candidate function has the same importance as the function being compared against
0	The candidate function has almost no importance as the function being compared against
9,7,5,3,1	Intermediate values between two adjacent judgments

Table 3.3: Pairwise Score Reference Table

The normalized weight for the i^{th} function is calculated according to the equation 3.7

$$w(i) = \frac{score(i)}{\sum_{i=1}^{N} score(i)} \tag{3.7}$$

It is to be noted that the weights calculated above are valid only for a certain part of the trajectory, for example when the target vehicle is expected to travel straight ahead without changing lanes. In this case, lane change cost function receives a zero weight, hence not contributing to the evaluated cost function. The weights therefore need to be updated at the 'switch points' or the 'decision points', where

	LCC	YRC	HC	OC	score	weight
LCC	1	0	0	0	1	0
YRC	10	1	7	2	20	0.6
HC	1	0	0	0	1	0
OC	10	2	2	1	15	0.4
				Total	37	

Table 3.4: Pairwise scoring for pursuit state

the behaviour of the vehicle is expected to change, for instance from lane following to lane change. The concept of switch points is elaborated in section 3.6.
The behavioural layer is an important aspect of a motion planning system. However, encoding the intricate traffic and behaviour rules into a motion planning algorithm as constraints is a primary challenge in recent motion planning systems. The method of pairwise comparison provides an effective way to incorporate the intricate traffic and behavioural rules of the maneuver into the algorithm along with flexibility to add multiple objective functions into the final scalar function. Also, as all the objective functions are cost functions, the final scalarized function can be formulated as an unconstrained minimization problem shown in equation 3.8.

$$F_{scalar}(x) = w_{occ}f_{occ}(x) + w_{YR}f_{WR}(x) + w_{head}f_{head}(x) + w_{LC}f_{LC}(x) \tag{3.8}$$

$$Objective : \min_{x} F_{scalar}(x)$$

The variable x in equation 3.8, represents the discrete poses [X,Y,θ] along a clothoid path in set {C}. The evaluation of each of the paths in {C}, was performed using a Markov Decision Process model.

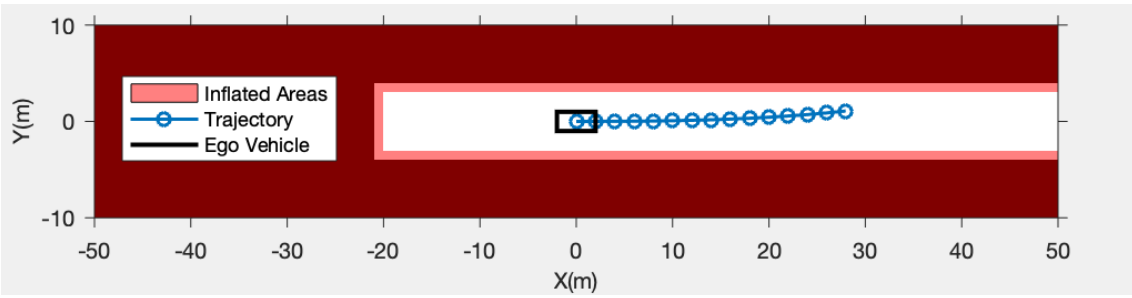


Figure 3.6: Clothoid trajectory on Costmap

To formulate the problem in the Markov Decision framework, the following terms are defined:
{S} : Finite set of all discrete vehicle poses generated by trajectory generation module : $\boldsymbol{S} \in \Psi$
{A} : Finite set of all actions
$T_a(s, s') = Pr(s_{t+1} = s'|s_t = s, a_t = a)$: Transitional probability of moving from state **s** to **s'** given action **a** , $\forall s, s' \in S$ and a $\in A$
$C_a(s, s')$:Immediate cost of moving from state $s_t = s$ to another state $s_{t+1} = s'$, due

to action ***a***

γ : Discount factor satisfying $0 \leq \gamma \ \leq 1$, for assigning higher weight to immediate states than future states.

In order to simplify the formulation of the problem at hand, the following assumptions were made:

- The set of actions {A}, represents the choice of clothoid to be followed. Hence, {A} $\equiv$ {C}
- The transitional probability from a state on one clothoid to a state on another clothoid, given an action **a**, is zero, i.e, if s_i is a state of the i^{th} clothoid and s_j is the state on the j^{th} clothoid, then $T_a(s_i, s_j) = 0$. In other words, the ego vehicle can not transition between different clothoids while trajectories are being evaluated at a given instance in time
- The transition probability from state s on a clothoid to the next state s' on the same clothoid is one, i.e., $T_a(s_i, s'_j) = 1$.
- The cost function $C_a(s, s')$ is equivalent to the scalarized function defined in 3.8

In order to find the optimal action a^* for the agent at a given state s, the optimal Q-function and value functions are derived. Considering the assumptions mentioned above, the equation 2.16 reduces to the evaluation of the states on a single clothoid, since cross transition probability between clothoids is zero. Hence, the Q-function for each clothoid $C_i \in \{C\}$ reduces to

$$\begin{gathered} Q^*(s,a) = C(s,a) + \gamma \sum_{s' \in C_i} V^*(s') \qquad \forall C_i \in \{C\} \\ Q^*(s,a) = C(s,a) + \gamma C(s',a) + \gamma^2 C(s",a) + ... \end{gathered} \tag{3.9}$$

where, $[s', s'', ...]$ are successive states on the clothoid. Once all the clothoids are assigned with Q-functions, the optimal action a^* is simply obtained by minimizing over the set of actions as shown in 3.10.

$$a^* = \min_a Q^*(s,a) \tag{3.10}$$

This process results in a method, in which each clothoid is evaluated with a fixed horizon of 30m, and the one with the least cost is selected. However, it is to be noted that the vehicle moves only to the next state on the clothoid after the evaluation and not the entire 'horizon length'. The resulting algorithm is a "populate-and-prune" algorithm, where at every time-step, a set of candidate solutions are generated and evaluated (populating) and all except the best are discarded (pruning). The clothoids generated at different stages in the maneuver are illustrated in Figure 3.7.

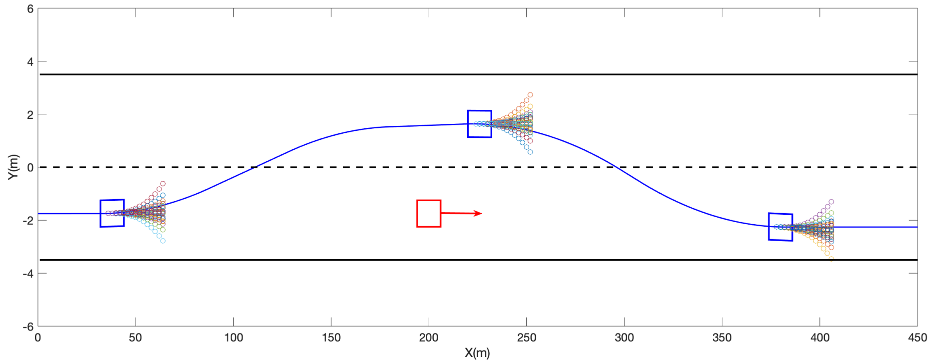


Figure 3.7: Clothoid trajectory set generated during overtaking maneuver

3.4 Feedback controller

The local feedback controller is the final layer in the trajectory planning hierarchy. The controller takes as reference the 'way point' generated by the trajectory generator module, and directly applies control inputs on the vehicle actuators such as steering and brakes. For this work, a simple kinematic controller was chosen and implemented. It was found that for the scenario under consideration, the kinematic controllers proved to be robust and computationally inexpensive. Hence, the Stanley controller designed and validated by Stanford University's Racing Team in the DARPA 2005 challenge [23] was implemented. The controller is based on the single track vehicle model as shown with the figure taken from [23]. This model does not take into account the inertial effects of the vehicle.

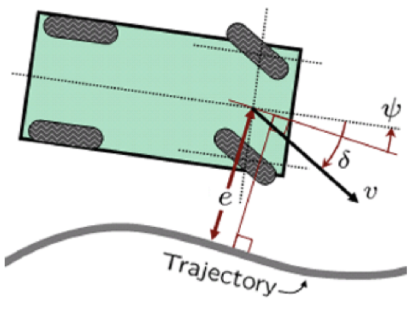


Figure 3.8: Kinematic vehicle model

The Stanley controller aims at minimizing the cross tracking error 'e', defined as the distance between the center of the front axle of the vehicle and the closest point on the reference trajectory as shown in figure 3.8. Along with the cross tracking error, the controller also tries to minimize the heading error, defined as the angle between the tangent along the closest point on the trajectory and the longitudinal axis of the vehicle. The rate of change of the cross tracking error can be expressed as

$$\dot{e} = v(t)sin(\psi(t) - \delta(t)) \tag{3.11}$$

where $\psi(t)$ and $\delta(t)$ are the heading and steering angle of the vehicle respectively, and $\delta \in \{\delta_{min}, \delta_{min}\}$. The yaw rate change $\dot{\psi}$ is given by

$$\dot{\psi}(t) = \frac{-v(t)sin(\delta(t))}{a+b} \tag{3.12}$$

where, a and b are the distance of the front and rear axles to the center of gravity of the vehicle. The steering law used to control the vehicle is as shown in equation 3.13.

$$\delta(t) = (\Psi(t) - \Psi_{ss}(t)) + tan^{-1}\frac{ke(t)}{v(t)} + k_{d,yaw}(\dot{\Psi}(t)_{meas} - \dot{\Psi}(t)_{traj}) \tag{3.13}$$

where, $\Psi_{ss}, k, k_{d,yaw}$ are the steady state yaw, cross track and yaw gains of the controller respectively.

3.5 Analysis for Naturalistic Data

In this section, a brief description about the extraction and analysis of HighD data is presented. The original data is analyzed by help of existing tools provided by HighD. Initially, the data was evaluated by comparing the HighD data parameter with the parameters used in the algorithm such as positions, velocities, heading angles and accelerations. There are many events in the data set, such as overtaking, single lane change and lane keeping. However, to evaluate existing overtaking vehicles, we considered data sets with low traffic flow, because with heavy traffic the chances of overtaking are bleak.

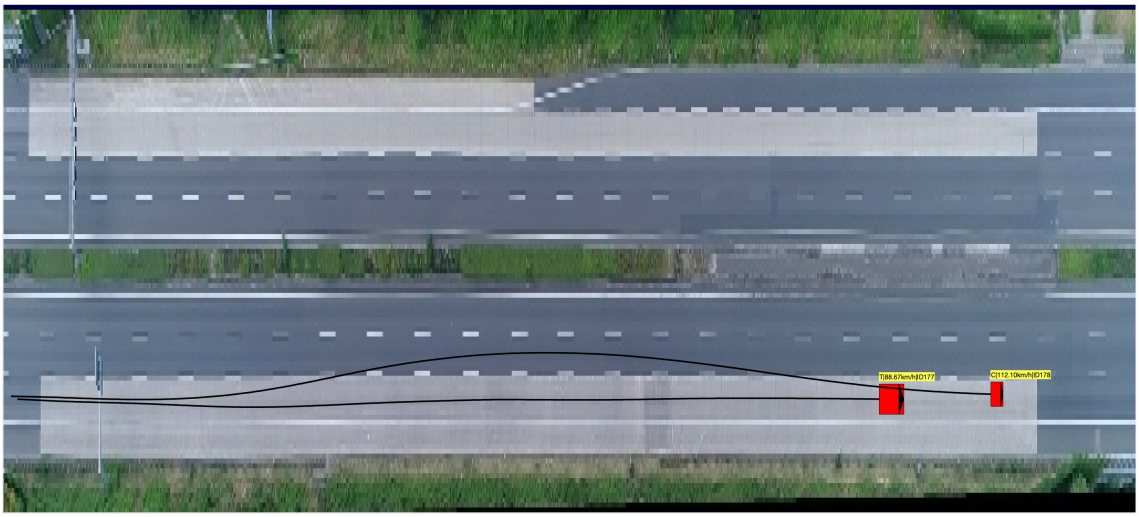


Figure 3.9: Annotated image of one of the overtaking events in the HighD dataset

The data was manually annotated with the total number of overtaking events, the vehicles' class and vehicle ID recorded. A total of 71 events were recorded, which contained information about the ego vehicle and the target vehicle. Using the recorded vehicle IDs, the parameters such as lateral clearance, vehicle velocities, longitudinal gaps before and after overtaking were extracted. An example of one of the recordings is shown in figure 3.9

The highD dataset does not allow us to know the driver's inputs, but just the resultant vehicle trajectories. It is possible to estimate the driver's inputs using inverse models of vehicle dynamics but there is no available golden truth to validate such models. Therefore, based on the few common parameters lateral velocity, acceleration, range, lateral clearance, the generated trajectory is compared with those in HighD data set.

3.6 Probabilistic modelling of Naturalistic Data

Naturalistic driving data is a vital source of information about the way humans interact with other traffic agents while driving. For the effective integration of autonomous vehicles in a majorly human-centric traffic, it becomes essential to recognize and incorporate these behavioural traits in the autonomous systems. In order to simplify the analysis of the naturalistic data, the overtaking maneuver was divided into four phases as shown in figure 3.10

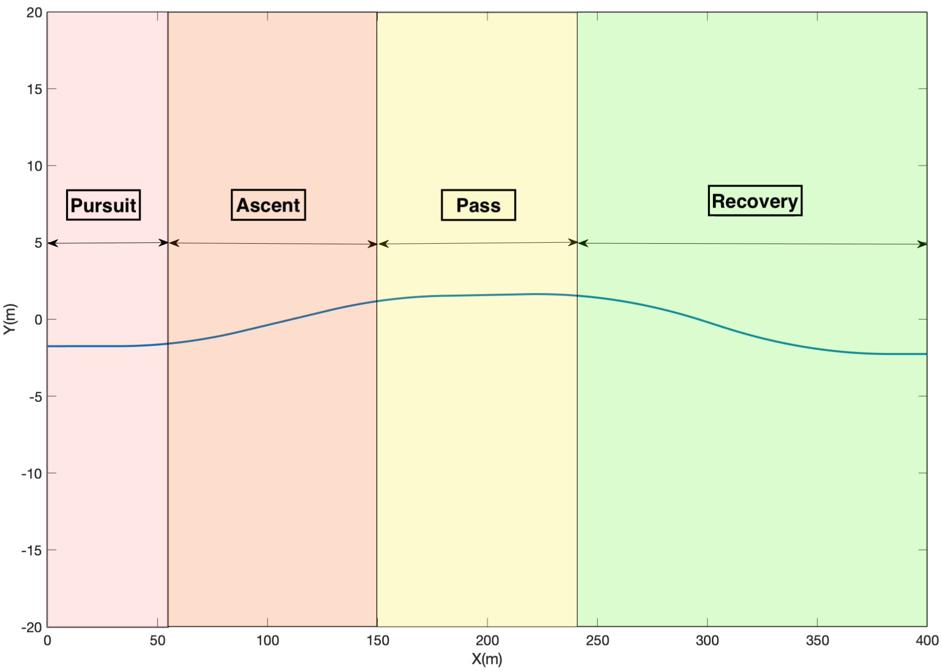


Figure 3.10: Overtaking by phases

- **Pursuit**: The ego vehicle follows the target vehicle with near zero lateral velocity (i.e. $v_y \approx 0$)
- **Ascent**: The lateral velocity of the ego vehicle increases in the positive y-direction, initiating a lane change maneuver
- **Overtake**: The ego vehicle passes the target vehicle while staying in the lane adjacent to that of the target vehicle
- **Recovery**: The lateral velocity of the ego vehicle increases in the negative y-direction, initiating a lane change maneuver until the ego vehicle has merged into the target vehicle lane

To define a probabilistic framework for analyzing the driver behaviour data, two 'switch points' were defined as shown in figure 3.10. The switch point 'A' is called 'Start of Ascent' and the switch point 'B' is called 'Start of Recovery'. The switch points, characteristic to every individual driver, represent the decision points at which the driver chooses to switch between the successive states of the maneuver, leaving enough longitudinal clearance to the target vehicle. For the parametrization of the 'switch points', the longitudinal gap between the ego and target vehicle and lateral velocity of the vehicle was considered. The longitudinal gap between the ego and target vehicle will henceforth be referred to as 'Range'. A positive range indicates that the ego vehicle trails the target vehicle, while a negative range value indicates the ego vehicle leading the target vehicle.

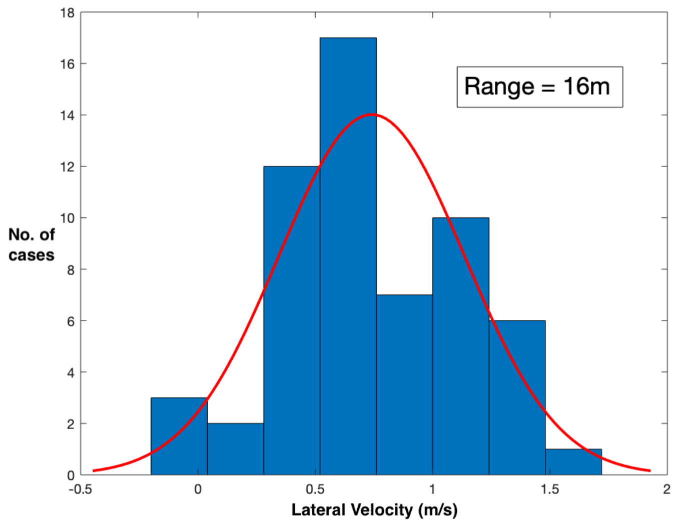
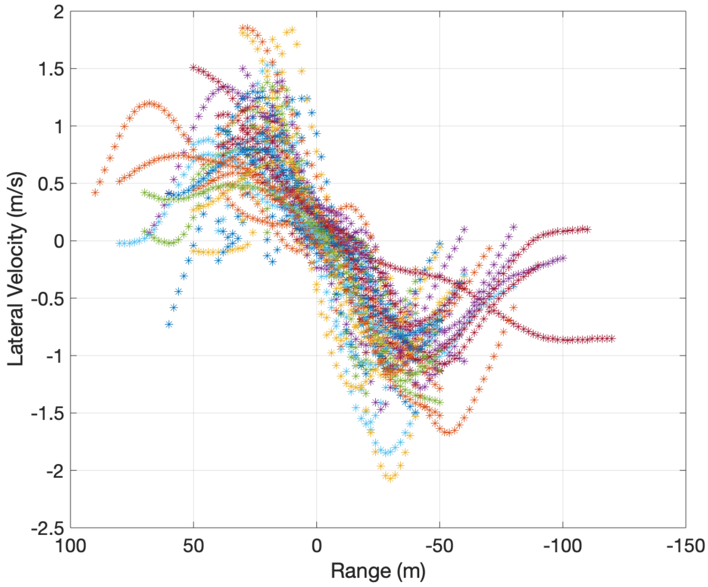
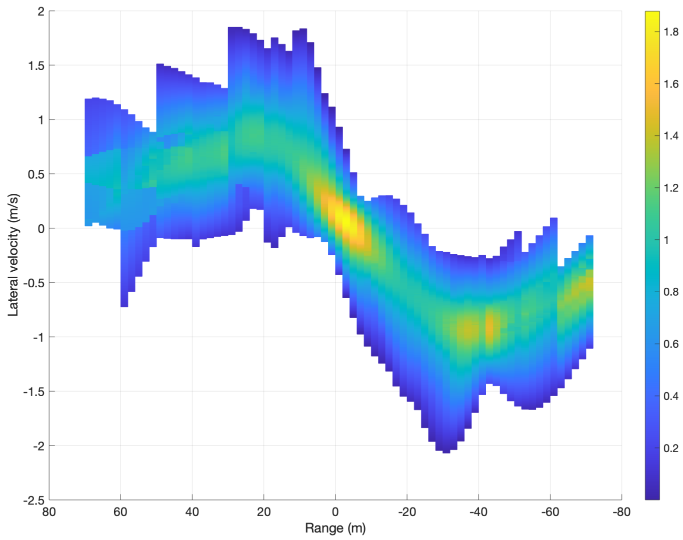


Figure 3.11: Lateral velocity histogram with normal data fitting

In total, 71 overtaking maneuvers were selected to analyze the human driving behaviour in overtaking maneuvers using the distribution of lateral velocity over different ranges. Figure 3.12a shows the distribution of range between the ego and the target vehicle and the corresponding lateral velocity of the ego vehicle for all the overtaking maneuvers recorded in the High-D data set. This distribution was used to model the naturalistic driving behaviour during an overtaking maneuver . Figure 3.11 shows a normal distribution $N(v_y; \mu = 0.7399, \sigma = 0.3962)$, that is fit to the lateral velocities recorded for 71 drivers, when the target vehicle is 16m ahead the ego vehicle. In a similar way, a normal distribution was fit to the lateral velocities for ranges from 70m to -70m. It is to be noted that although the raw data shown in figure 3.12a contains range from 90m to -120m, only ranges from 70m to -70m were used to fit normal distributions due to insufficient data points in extremities of the data. Figure 3.12b shows the lateral velocity probability distribution fit over all the ranges.



(a) Longitudinal gap vs Lateral velocity



(b) Probabilistic distribution

Figure 3.12: Naturalistic data analysis

Toledo et. al. [24] found that the average lane change duration for a passenger car was 4.6s and 4.4s for a left and right lane change maneuver respectively. Assuming the ego vehicle starts from the middle of the right lane and moves to the middle of the left lane, the average lateral velocities for the left and right lane changes are approximately 0.76m/s and -0.79 m/s.
In order to determine the 'Start of Ascent', the critical range for ascent R_{ca} and recovery R_{cr} was found using ,

$$\begin{aligned} R_{ca} &= P(v_y \geq 0.76|R) = 0.6 \quad \forall R \in [70, -70] \\ R_{cr} &= P(v_y \leq -0.79|R) = 0.6 \quad \forall R \in [70, -70] \end{aligned} \tag{3.14}$$

The probability values in equation 3.14, were found by converting the probability density function in figure 3.12b to probability mass function using the equation

$$P(x \geq v_{y_{crit}}) = \int_{v_{y_{crit}}}^{\infty} p(x)dx \tag{3.15}$$

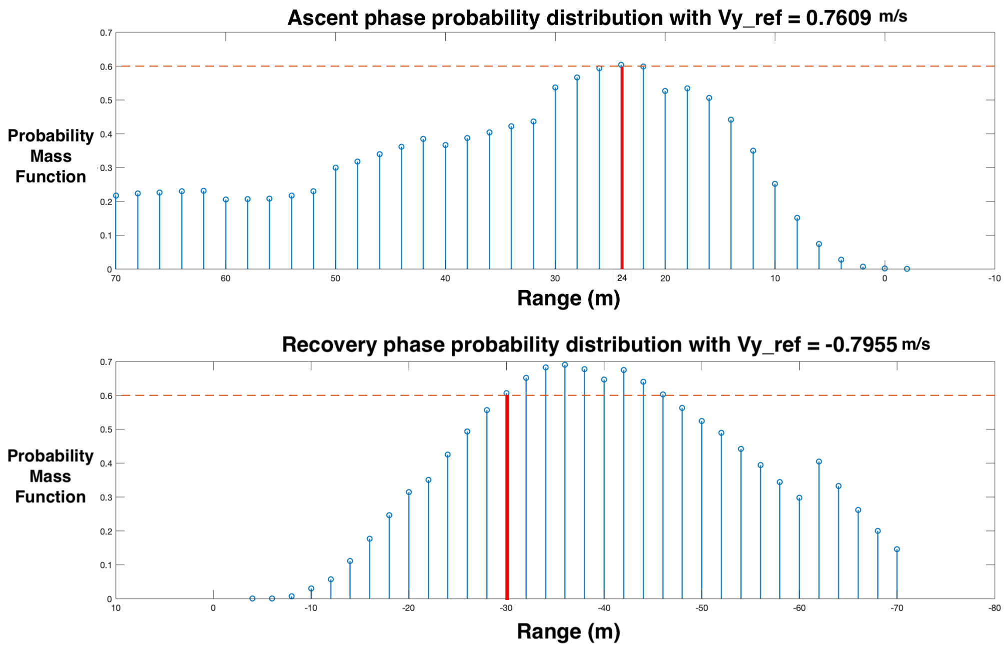


Figure 3.13: Probability mass function for critical range in Ascent and Recovery phase

The threshold value for the probability was chosen to be 0.6 based on the quality of the data set. Further addition of data to the naturalistic data set may result in better estimation and higher values of probability mass function. As seen in the figure 3.13, the critical ranges for Ascent and Recovery phases were found to be 24m and -30m respectively. It is to be noted that these ranges represent the distance between the target and ego vehicles when the ego vehicle reaches the corresponding lateral velocities, and not the 'switch point' itself. In order to derive the 'Start of ascent' and 'Start of recovery' points, the linearly increasing curvature property of the clothoid is used.

As mentioned before, one of the most notable properties of the clothoid is that a vehicle travelling along one experiences a constant lateral acceleration. This property was used to derive the 'switch points'. The time required for the ego vehicle travelling with constant lateral acceleration a_y, to reach a critical lateral velocity $v_{y,crit}$ starting from $v_y = 0$, is given by

$$\Delta t = \frac{\Delta v_y}{a_y} \tag{3.16}$$

The 'switch point', i.e the range at which the ego vehicle switches to the next state, can be derived by linearly interpolating the critical range backwards in time as shown in the following equation

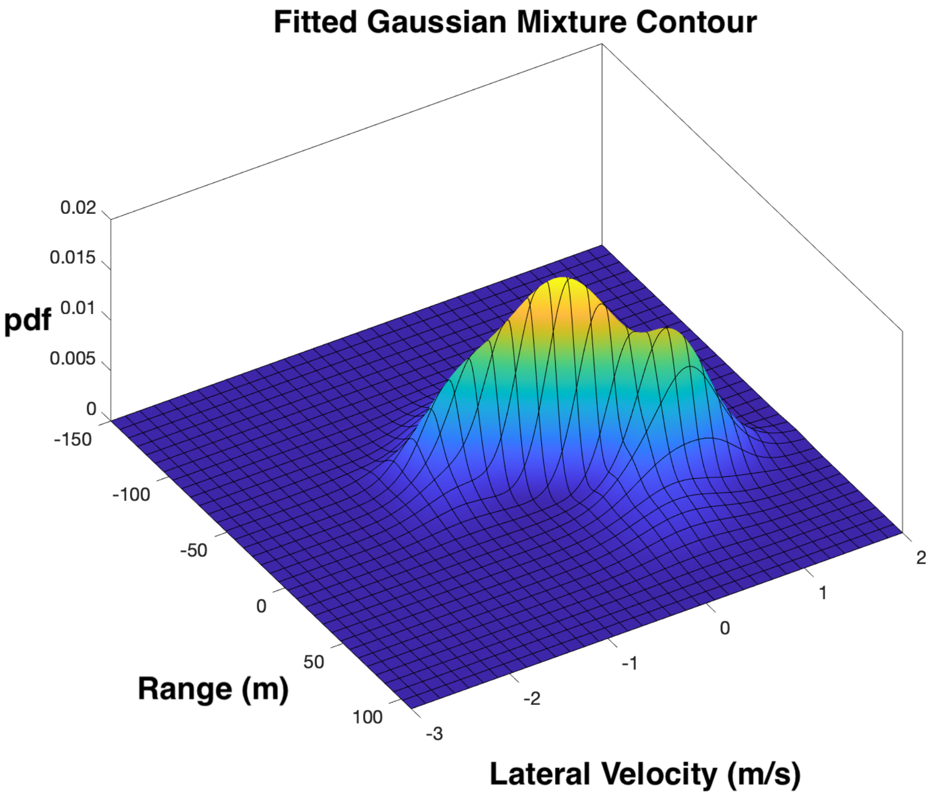
$$R_{switch} = R_{crit} - v_{ego}\Delta t \tag{3.17}$$

3.7 Intent Recognition with Unsupervised Learning

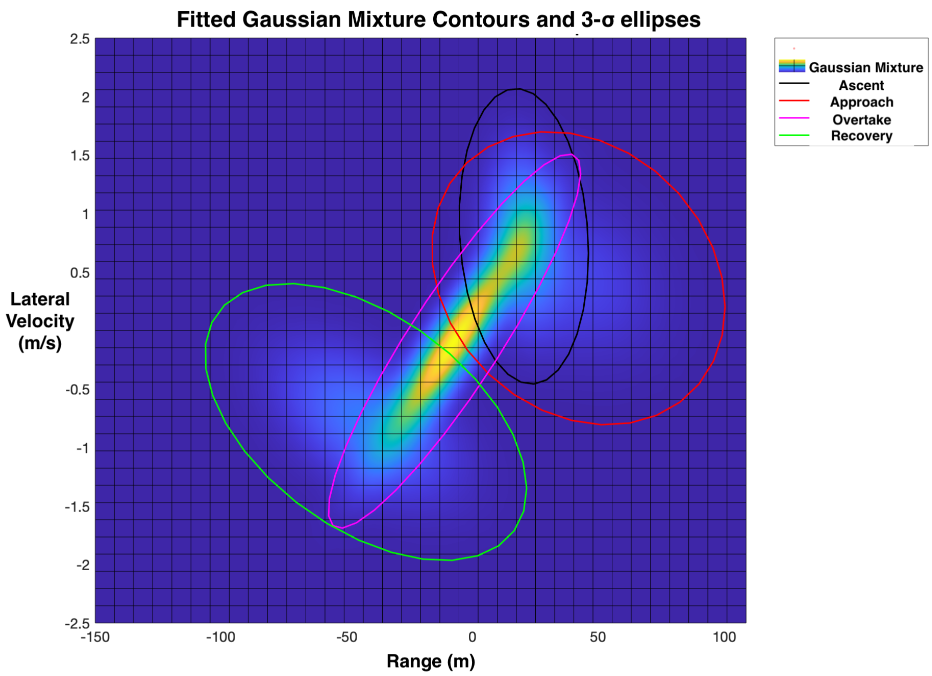
In recent years, driver intent recognition has been a topic that is intensively being researched upon. Among other literature, Haufe et. al [25] have used electroencephalography (EEG) and electromyography (EMG) to recognize and predict drivers' braking intents. Xing et. al [26] have used CAN signals along with unsupervised learning models to classify and identify drivers' braking intentions. However, in majority of the naturalistic driving data sets, such sophisticated signals are seldom available. Hence, in this section, we introduce a k-Mean and Gaussian Mixture Model (GMM) based unsupervised learning framework that recognizes and predicts the drivers' intent to overtake using just kinematic data.

Although supervised learning models are extensively used for intent recognition, one of the major challenges with such models is the labelling of data. Since it is extremely difficult to determine and label the exact moment the driver intends to overtake with only kinematic vehicle data, unsupervised learning algorithm was chosen.

The system developed in this work uses the longitudinal distance between the subject vehicle and its preceding vehicle (range) and the lateral velocity of the subject vehicle to predict the instance at which the subject vehicle begins a lane change maneuver. The k-mean clustering and the GMM algorithms are thus configured to predict using four clusters representing the four phases of the maneuver. The model is trained on 58 overtaking maneuvers from the HighD data set. Since the GMM models tend to converge on local optima and are hence sensitive to initial conditions, the initialization of the Gaussian mixture was done using one iteration of k-Means clustering. The GMM was run for 2000 iterations and the resulting converged Gaussian mixture is shown as in figure 3.14a.



(a) GMM contour fit



(b) 3 sigma ellipses for overtaking maneuver phases

Figure 3.14: Gaussian Mixture Model for intent recognition

Figure 3.14a shows the resulting Gaussian mixture model and 3.14b represents the 3-σ ellipses that corresponds to the four phases of the overtaking maneuver. Each of the four ellipses captures the driver behaviour in the form of joint distribution of range and lateral velocity for the corresponding phase. Hence, given point X in the range-lateral velocity state space, the GMM predicts the posterior probabilities for each of the clusters Y_i according to

$$P(Y = Y_i|X) = \frac{P(X|Y = Y_i)\pi_{y_i}}{\sum_{i=1}^{N} P(X|Y = Y_i)\pi_{y_i}} \tag{3.18}$$

The model was further tested on 11 highway overtaking scenarios, that had not been used for training. The accuracy and robustness of the model is assessed in further sections.

3.7.1 Pseudo Ground Truth

In order to asses the robustness of the model, the ground truth data would require the exact instance at which the driver initiates a lane change maneuver. However, due to unavailability of such ground truth data in HighD data set, pseudo ground truth (PGT) data using vehicle kinematics was generated.

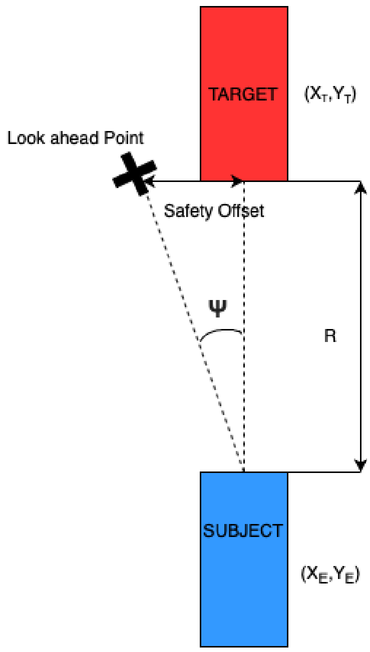


Figure 3.15: Pseudo ground truth generation

The PGT framework, for each test run, evaluates the yaw angle required by the subject vehicle in order to point it's velocity vector clear of the preceding target vehicle. This criteria was defined using a clearance yaw using a look-ahead point, that is laterally offset from the preceding vehicle's center by a safety distance. Thus, clearance yaw and the vehicle yaw was calculated using equation 3.19.

$$\begin{aligned} \Psi_R = tan^{-1}\frac{(Y_T - Y_E + SO)}{R} \\ \Psi_A = tan^{-1}\frac{v_y}{v_x} \end{aligned} \tag{3.19}$$

where, SO = safety offset from the target vehicle center line
R = longitudinal distance between the front of the subject vehicle and the rear of the target vehicle
v_y, v_x are the lateral and longitudinal velocities of the ego vehicle
Hence, the PGT is generated assuming that the lane change is initiated at the instance when the vehicle yaw (Ψ_A) exceeds the clearance yaw (Ψ_R). It is to be noted that the look ahead point, and hence the safety offset distance is characteristic to every individual driver according to their driving style. Since there does not exist a standard value for such offset in practice, the width of the target vehicle was used as the safety offset value in this model. In other words, if the subject vehicle's yaw angle steers it off the target vehicle by half the target vehicle's width, the subject vehicle can be assumed to have started a lane change maneuver.

4
Results

The section showcases the results of evaluation of the developed local trajectory planner for the scenario in 1.1. The robustness of the planner was tested for three different scenarios. In addition to using MATLAB ®, IPG CarMaker ® simulation platform was used to supplement the results of the trajectory planner. It is to be noted that while MATLAB® was used to generate a reference trajectory, the local feedback control was implemented in IPG CarMaker ®. The IPG CarMaker® complemented the MATLAB® simulation by enabling logging of dynamic vehicle parameters such as lateral acceleration and yaw rates on accurate models of real world vehicles, such as Volvo XC 90 which were then used to evaluate important factors such as ride comfort and path following capability. Subsequently, the performance of the GMM model based intent recognition system is validated. The possibility of implementing such models in emergency braking systems to avoid/mitigate a rear end collision is assessed.

4.1 Scenario 1: Ego vehicle avoiding a static target vehicle

The scenario was initialized with the ego vehicle placed at the center of the slow lane, and the target vehicle placed 80 meters ahead of the ego vehicle in the same lane. The velocity of the target vehicle was set to zero in order to simulate a situation where the target vehicle is immobilized due to a sudden breakdown. However, the ego vehicle's velocity was kept constant at 30 m/s (108 km/h) which is in accordance to the speed limits on Swedish highways [27]. The simulation was run both on MATLAB ® using the automated toolbox and on IPG ® Car Maker. Figure 4.1 shows the trajectory of the ego vehicle maneuvering around the target vehicle.

Vehicle Parameter	Value
Mass	1675 Kg
Moment of Inertia(i_{zz})	$2617kgm^2$
Wheel Base	2.675 m
Width	1.51 m
Wheel base	1.2572 m
Distance of COG from rear axle	1.4178 m
Tyre Stiffness	30.7 Nm/rad
Steering ratio	15.9
Height of CoG	0.543 m
Front Roll center height	0.045 m
Rear Roll center height	0.101 m
Total Roll stiffness	$7e^4$ Nm/rad

Table 4.1: Vehicle Specification of Saab 93

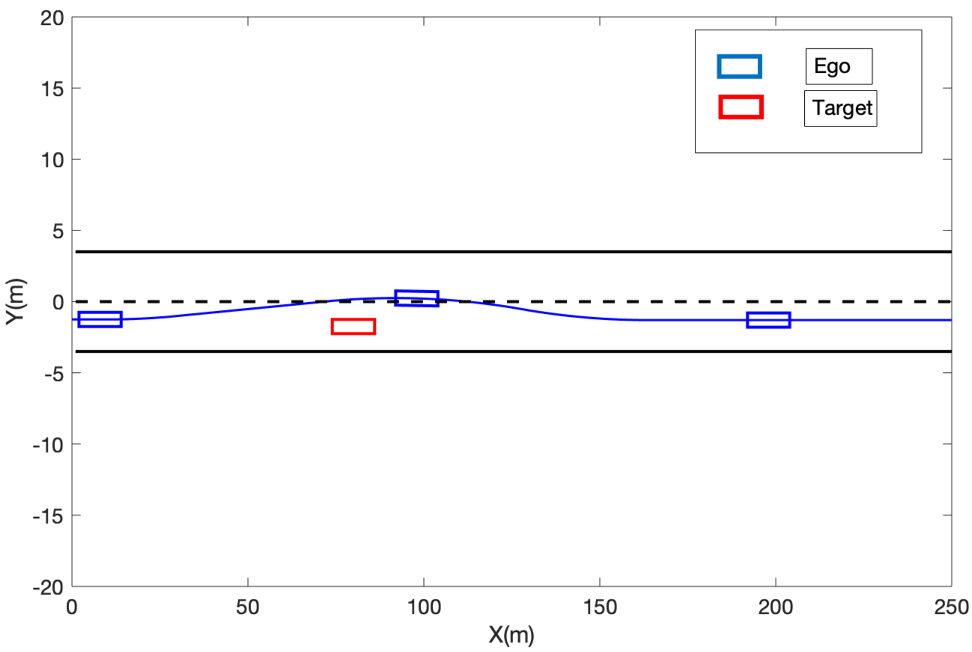


Figure 4.1: Static target collision avoidance maneuver

4.2 Scenario 2: Ego vehicle overtaking low velocity target vehicle

This situation is similar to 4.1, but in this scenario the target vehicle is kept to travel at slow speed of 10m/s (36 km/h) in the center of the slow lane (lane 1). However, the Ego vehicle velocity was kept to be constant at 30m/s. This scenario was considered as there exists mixed traffic situations on highway, for example, when the target vehicle slows down to exit the highway. Thus this scenario aims to analyse how the following vehicle will complete the maneuver with a slow moving target vehicle. The resulting maneuver is shown in figure 4.2.

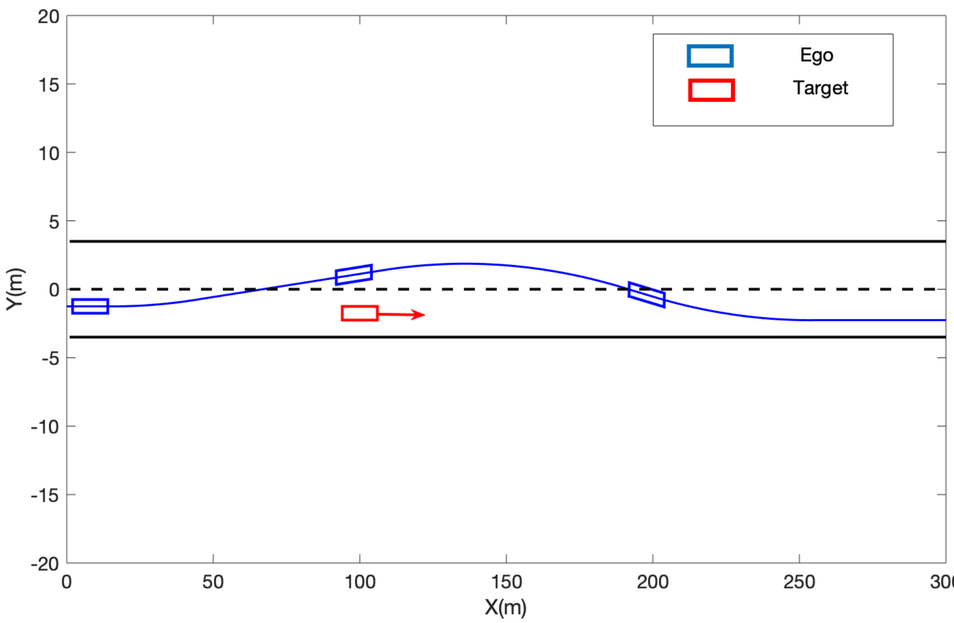


Figure 4.2: Low Speed Overtaking Trajectory

4.3 Scenario 3: Ego vehicle overtaking high velocity target vehicle

This scenario is similar to the above mentioned scenario. However, this scenario aims to simulate a situation where the ego vehicle is overtaking a target vehicle moving at a relatively high speed of 20 m/s (72 km/h). This scenario is the most likely scenario in highway traffic where the target vehicle has a cruising velocity and hence represents a more general scene.

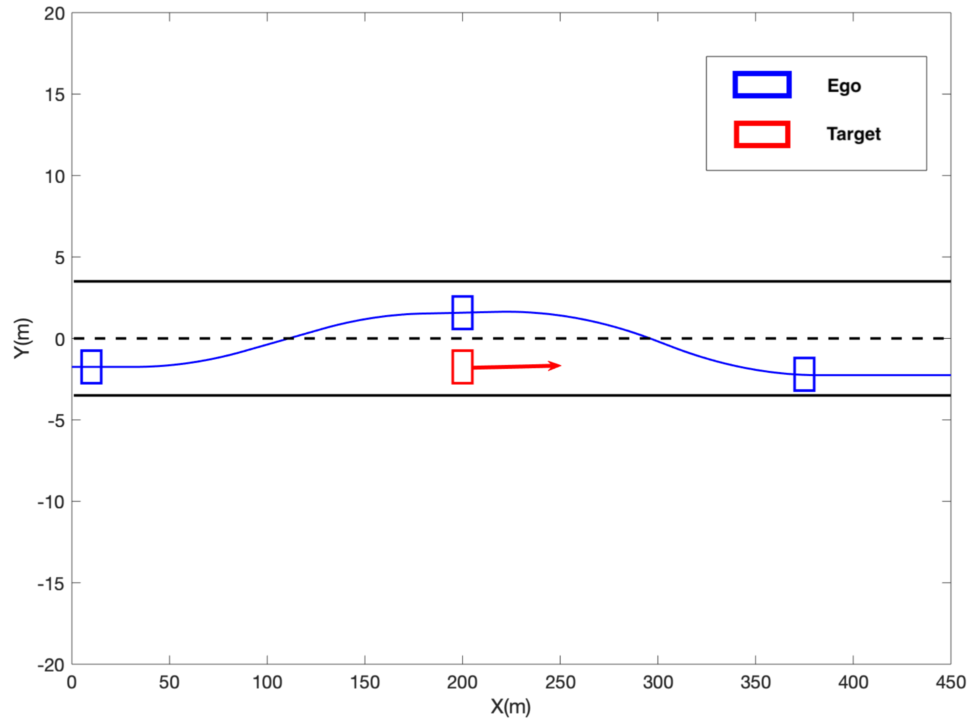


Figure 4.3: High Speed Overtaking Trajectory

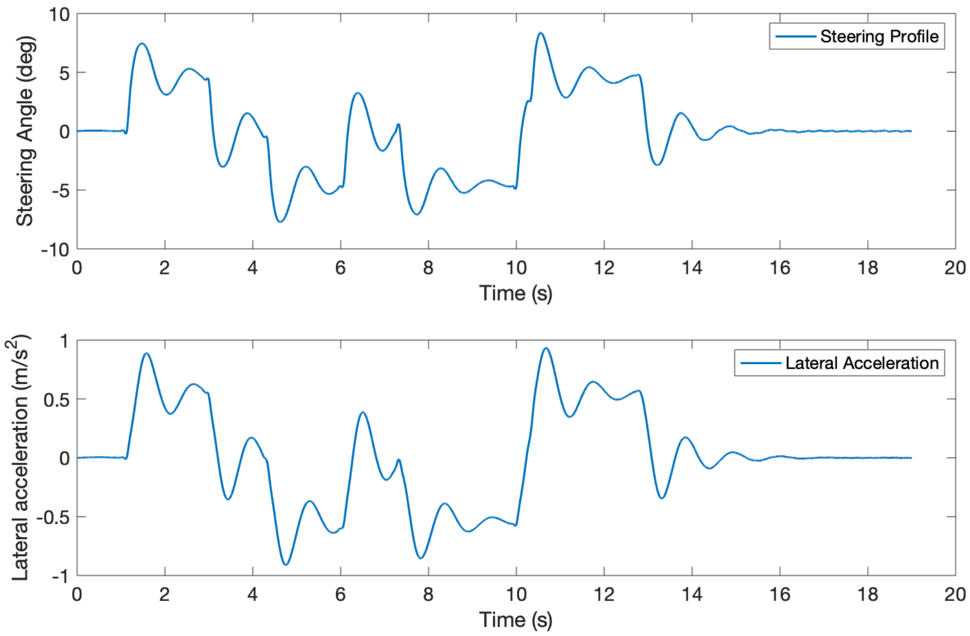


Figure 4.4: Actuation Signal

The ego vehicle's steering actuation angle and lateral velocity profile for the complete high speed maneuver is shown in figure 4.4. As seen, the lateral acceleration values are well within the comfort limits of $1m/s^2$. The steering wheel actuation ranges from about -7deg to 7deg, which maneuvers the car smoothly without excessive jerks.

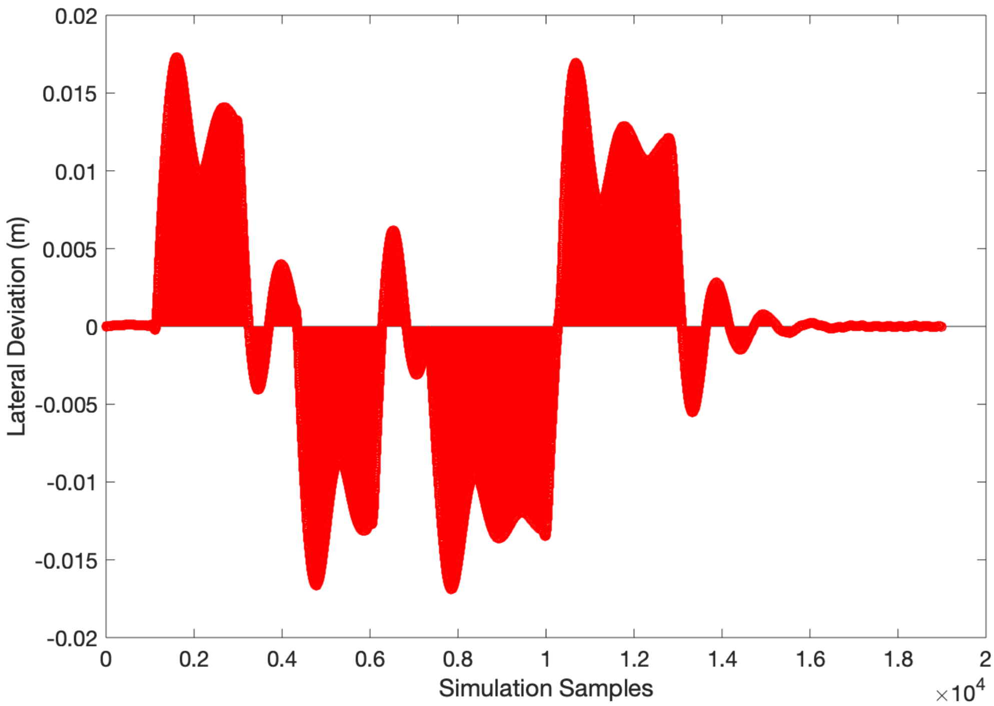


Figure 4.5: Reference Path Deviation

Figure 4.5 shows that deviation of the path generated by the Stanley controller's actuation and the reference path generated by the algorithm. The maximum absolute lateral deviation is about 0.018 m (18 mm), which states that the trajectory generated by the algorithm is smooth enough for the controller to follow. Figure 4.6 shows the simulation from the IPG CarMaker simulation platform.

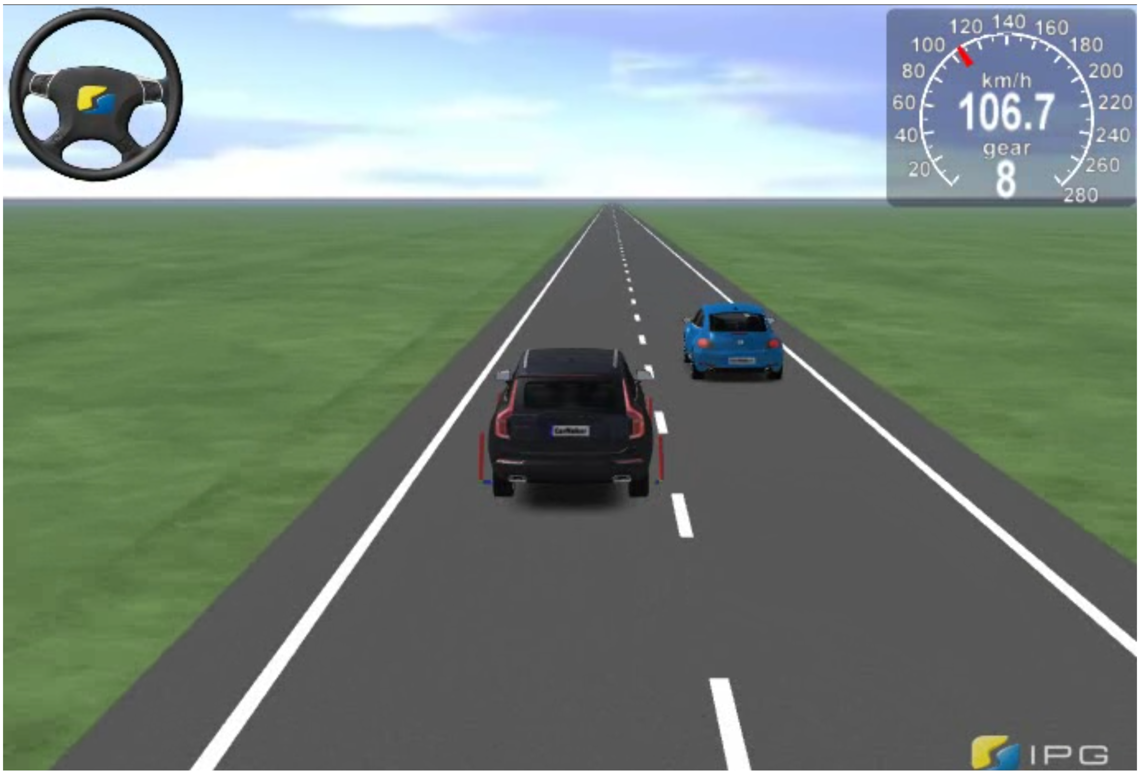


Figure 4.6: Car Maker Simulation

To compare the performance of the algorithm in the three scenarios mentioned above, the following parameters were selected.

- **Simulation run-time** : The time elapsed between the start of the scenario until the ego vehicle reaches the goal state, i.e, returns to the slow lane after successful overtaking
- **Lateral clearance** : The lateral clearance between the ego and the target vehicles measured from their center axes
- **Time spent on the fast lane**: The time elapsed between the instance when the ego vehicle enters the fast lane until it returns to the slow lane

These parameters from the three scenarios is shown in figure 4.7.

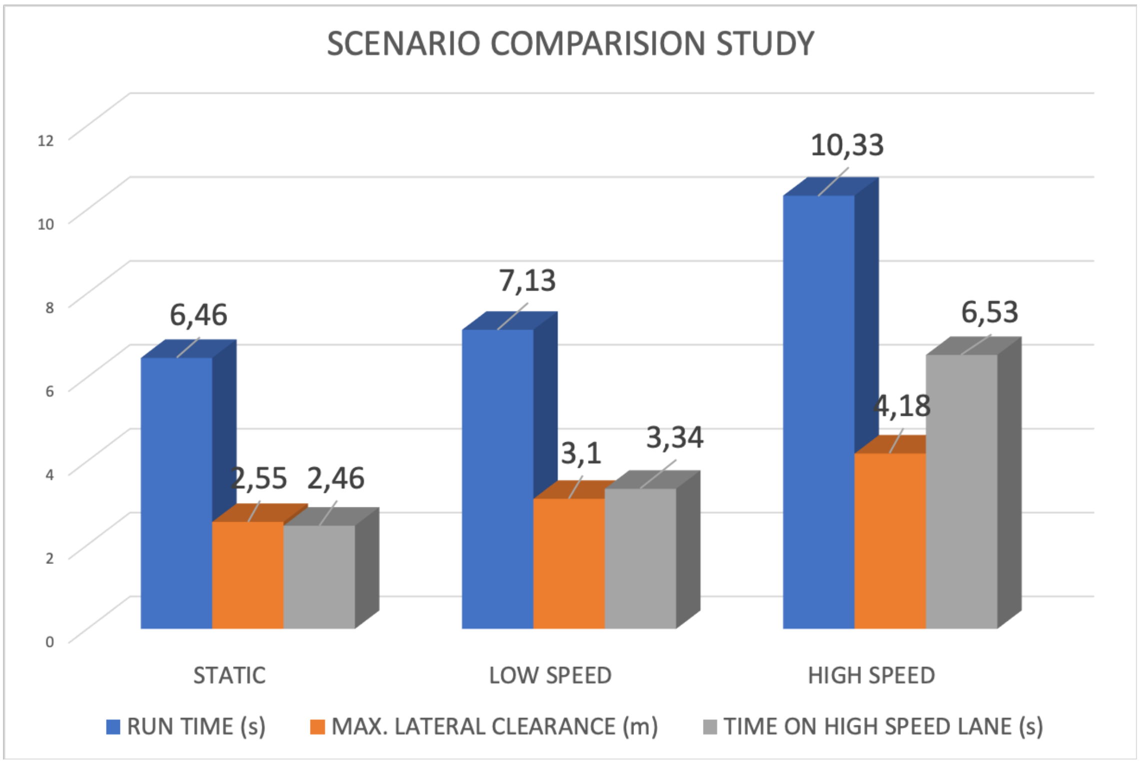


Figure 4.7: Scenario Parameter Comparision

The simulation run time increases as the velocity of the target vehicle increases, which is expected as the ego vehicle takes longer to pass the target vehicle and merge back to the slower lane leaving enough headway for the target vehicle.

The lateral clearance is an important parameter especially in overtaking scenarios, as it directly represents the comfort zone boundaries (CZB) of the drivers in both ego and target vehicles. As seen in figure 4.7, the lateral clearance between the ego and target vehicle increases with increase in target vehicle's longitudinal velocity. This result is in agreement with the findings of Budhkar et. al [28], which points towards a positive correlation in the naturalistic data, between the interacting vehicles' longitudinal velocities and the lateral clearance between them while overtaking. From common experience, it is known that minimizing the driving time on a fast lane leads to better safety. Hence, after the overtaking is completed it is advisable to merge to the slower lane, given there is no other target vehicle to overtake. The results show that the algorithm allows the ego vehicle to return to the slower lane as soon as the overtaking is complete. Hence, it aims to minimize the time spent on the fast lane while leaving a safe headway for the target vehicle, while merging.

In order to asses the similarity between a human driven overtaking maneuver and the trajectory generated by the automated system, the three parameters mentioned above were compared for a high speed highway scenario and the results are as shown below.

Parameter	**Naturalistic**	**Automated**
Run-time(s)	12	10.33
Max. Lateral Clearance (m)	3.7	4.18
Time on high speed lane (s)	6.8	6.53

Table 4.2: Naturalistic vs Automated driving trajectories

The incorporation of naturalistic data in a probabilistic framework allows the developed path planning algorithm to truly imitate the human driving behaviour.

4.4 Intent Recognition

In this section the robustness of the intent prediction model is assessed using a highway scenario as shown in figure 4.8. In the scenario, the ego vehicle is approaching the target vehicle to perform an overtaking maneuver. However, the target vehicle itself initiates an overtaking maneuver on the lead vehicle. Given the longitudinal gap, i.e., range between the target and the lead vehicle and the lateral velocity of the target vehicle, the intent recognition system on the ego vehicle should be able to predict the overtaking intentions of the target vehicle in order to avoid a potentially dangerous scenario.

The robustness of the intent recognition system was evaluated on its ability to recognize the lane change intention of the target vehicle early enough, so that there exists enough braking distance S_{warn} for an emergency braking system to be deployed to avoid a rear end collision.

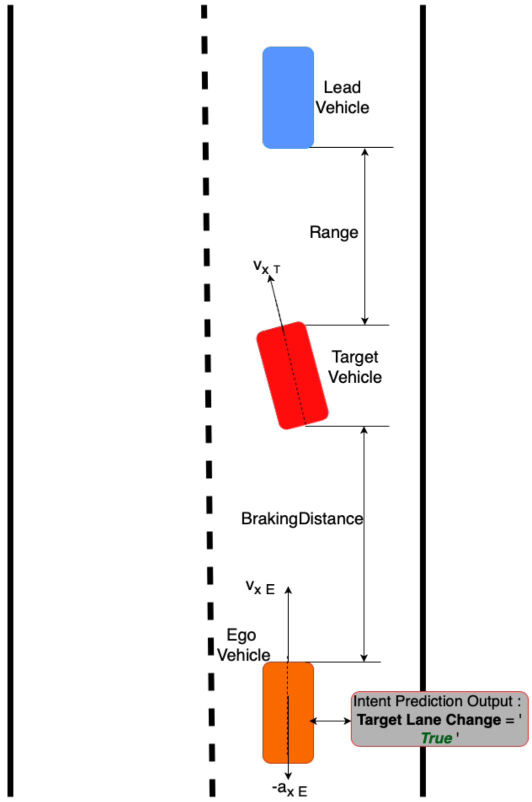


Figure 4.8: Target Lane Change Scenario

where, $v_{xE} \geq v_{xT}$ are the velocity vectors of the ego and target vehicles in m/s and a_{xE} braking deceleration of the ego vehicle in m/s^2.
Figure 4.8 shows the scenario, in which the ego vehicle proceeding to overtake the target vehicle, and predicts the target vehicle's intention to initiate a lane change maneuver. In response, the ego vehicle should execute a braking maneuver until $v_{xE} \leq v_{xT}$ in order to avoid a rear end collision. The critical range $S_{critical}$ required to perform such a maneuver was calculated using equation 4.1

$$S_{critical} = \frac{(v_{x,E}^2 - v_{x,T}^2)}{2a_{max}} \tag{4.1}$$

where, a_{max} is the maximum deceleration capability of the ego vehicle which is assumed to be $8m/s^2$ [29]. Thus, the condition for avoiding a rear-end collision was defined by, $S_{warn} \geq S_{critical}$. The performance results of the module are tabulated in table 4.3

Run	PGT-GMM ΔT (s)	Collision Avoidance
1	**-0.6**	✓
2	**1.32**	×
3	**-1.6**	✓
4	**1.64**	✓
5	**-0.32**	✓
6	**0.44**	✓
7	**-0.56**	✓
8	**-0.8**	✓
9	**-0.64**	✓
10	**2.72**	✓
11	**-0.48**	✓

Table 4.3: GMM Prediction Results

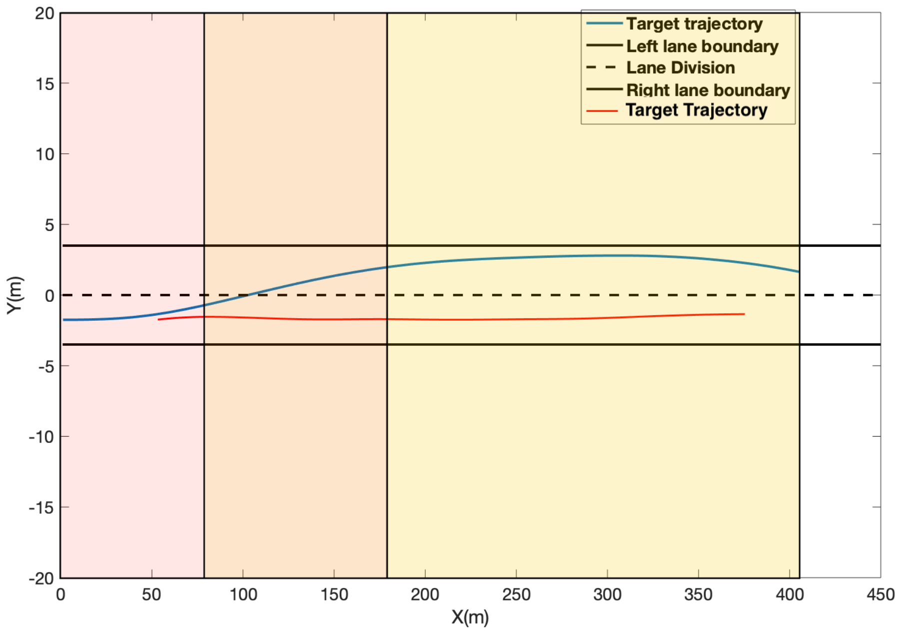


Figure 4.9: Overtaking maneuver phase prediction using GMM

In order to test the intent recognition system, 11 test cases were considered from the naturalistic data set. The results in 4.3 show the performance of the trained GMM model, as the intent recognition system manages to issue a warning early enough to avoid a rear end collision in 10 out of 11 cases. Also, the time delay between the PGT and the GMM model warnings is negative in 63% of the cases, which indicates that the GMM model recognizes the intent well in advance compared to the PGT model.

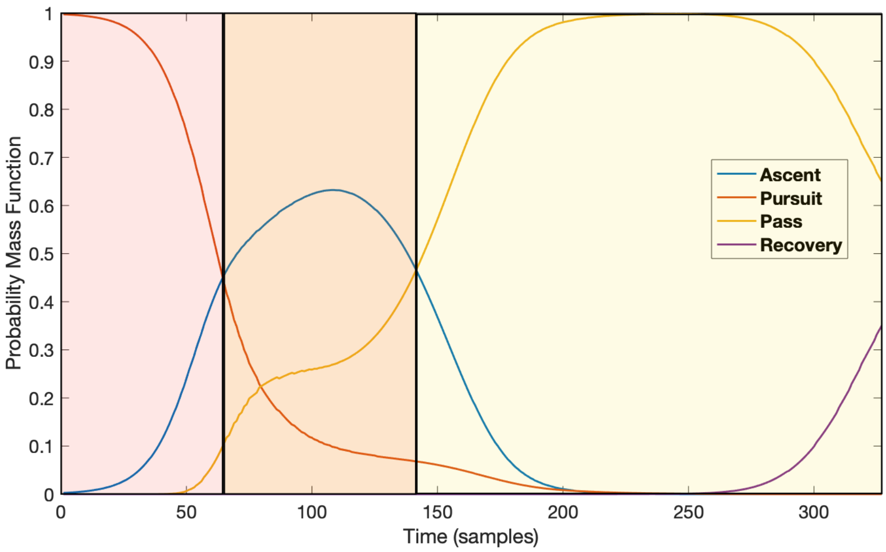


Figure 4.10: Posterior densities for overtaking maneuver phase predictions

Figure 4.9 shows the phases of the overtaking maneuver as classified by the GMM model for a maneuver selected from the High-D data set. The posterior densities corresponding to each of the four phases is shown in figure 4.10. The predicted phase is obtained by simply considering the maximum of the posterior distributions at any given point in the trajectory. The shaded regions represent the transition of the vehicle through different phases of the maneuver.
It is important to note that due to the limited amount of training data and the smoothing filters used during the post processing of the naturalistic data, there are some inherent inaccuracies in the GMM intent prediction model. However, the results still look promising and can be expected to improve with further training of the model on high quality naturalistic data.

5

Conclusion

In this thesis project, a path planning algorithm has been developed, obtaining trajectory based on human driving patterns. Simplified vehicle dynamic models and naturalistic driving data are used to obtain a human-like, curvature bounded, smooth trajectory. The results from different simulation scenarios reflect the ability of the algorithm to perform in different environments while maintaining the ride comfort for the passengers in the vehicle. In this regard, the path planning is implemented effectively considering both human aspect and algorithm efficiency.

The use of single track vehicle dynamic model along with Euler spiral geometry for trajectory planning has shown some good potential with promising results. This method is feasible when the GPS based path planning suffers from signal interference in tunnels, rural areas with poor signal strength or on roads between high rise buildings. The vehicle dynamics model incorporating actual vehicle data allows for a more realistic trajectory generation without the need for additional path smoothing techniques.

The comparison between the human and automated overtaking trajectories truly reflects the ability of the developed path planning method to imitate the human driving behaviour. In the current traffic scenario, where human driven traffic dominates, such imitations define the ease of acceptance of autonomous vehicles in the traffic and the society in general.

The intention recognition model that predicts the lane change intention of the target vehicle based on GMM modeling has shown good results in predicting the behavior of the target vehicle. The prediction promptness of such systems is very essential to avoiding a high-speed collision and the implemented model has shown great potential in this regards. However, this goal can be better achieved by training a GMM model on a larger data set for better prediction accuracy under various conditions.

5.1 Future Work

The work done in this thesis has room for refinement and further development in several aspects. Following are some of the ideas that could be explored and implemented in future versions of the algorithm.

- Testing the performance of the trajectory planning algorithm with nonlinear filters (Unscented Kalman filter and Particle filter) for increased object detection and filtering accuracy.
- The clothoid set generated in the algorithm can be modified to be dynamic in nature i.e., that the number of tentacles generated can vary depending on the scenario. This might help reduce the overall computational cost of the algorithm throughout the maneuver.
- Machine learning algorithms can be used to develop inverse models which train on the naturalistic data set to calculate the weights for the cost functions in different scenarios such as lane change maneuvers and exiting express ways.
- CAN signals such as steering inputs, acceleration/braking inputs or lane change indications can be incorporated as additional parameters in the intent recognition model to improve the intent prediction accuracy of the surrounding vehicles.
- Ground truth/pseudo ground truth models used to verify the accuracy of the intent recognition system can be improved for better validation. For instance, a robust pre-trained machine learning model can be implemented to recognize the initiation of lane change maneuvers, using the steering input data instead of a simple kinematic model.
- The intent recognition results can be communicated to surrounding vehicles which might not be equipped with sensors, using V2V or V2I systems. This might aid the drivers to adapt to the changing driving conditions beyond their range of sight.

Bibliography

[1] R. Levien, "The Euler spiral: a mathematical history," p. 16.

[2] "White top car png image 3468, digital image, accessed : 2020-06-15, pluspng.com/png-57110.html."

[3] S. Gim, "Flexible and Smooth Trajectory Generation based on Parametric Clothoids for Nonholonomic Car-like Vehicles," p. 229.

[4] S. M. LaValle and J. James J.Kuffner, "Rapidly Exploring Random Trees: Progress and prospects," tech. rep., Iowa State University ,Ames,IA,USA, 2004. type: dataset.

[5] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments," *I. J. Robotic Res.*, vol. 29, pp. 485–501, Apr. 2010.

[6] D. J. White, "A survey of applications of markov decision processes," vol. 44, no. 11, p. 25.

[7] H. Mouhagir, R. Talj, V. Cherfaoui, F. Guillemard, and F. Aioun, "A Markov Decision Process-based approach for trajectory planning with clothoid tentacles," in *IEEE Intelligent Vehicles Symposium (IV 2016)*, (Göteborg, Sweden), pp. 1254–1259, 2016.

[8] *(2004) Holonomic and Nonholonomic Constraints. In: The Calculus of Variations. Universitext. Springer, New York, NY.*

[9] G. Vorotovic, B. Rakicevic, S. Mitić, and D. Stamenković, "Determination of cornering stiffness through integration of a mathematical model and real vehicle exploitation parameters," vol. 41, pp. 66–71.

[10] MATLAB, *9.7.0.1190202 (R2019b).* Natick, Massachusetts: The MathWorks Inc., 2019.

[11] "Robust perception from optical sensors for reactive behaviors in autonomous robotic vehicles."

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems." Library Catalog: www.semanticscholar.org.

[13] P. Closas and C. Fernández-Prades, "Particle filtering with adaptive number of particles,"

[14] Zhen Ding and B. Balaji, "Comparison of the unscented and cubature kalman filters for radar tracking applications," in *IET International Conference on Radar Systems (Radar 2012)*, pp. 82–82, Institution of Engineering and Technology.

[15] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125, Nov. 2018. ISSN: 2153-0017.

[16] "Piarc report 2001-2005, chapter v, traffic lanes and carriageway."

[17] J. Xu, K. Yang, Y. Shao, and G. Lu, "An experimental study on lateral acceleration of cars in different environments in sichuan, southwest china." ISSN: 1026-0226 Library Catalog: www.hindawi.com Pages: e494130 Publisher: Hindawi Volume: 2015.

[18] J. H. Reif, "Complexity of the generalized mover's problem.," p. 41.

[19] J. Canny and J. Reif, "New lower bound techniques for robot motion planning problems," in *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, SFCS '87, pp. 49–60, IEEE Computer Society.

[20] E. Glassman and R. Tedrake, "A quadratic regulator-based heuristic for rapidly exploring state space," in *2010 IEEE International Conference on Robotics and Automation*, pp. 5021–5028, IEEE.

[21] N. Gunantara, "A review of multi-objective optimization: Methods and its applications," vol. 5, no. 1, p. 1502242. Publisher: Cogent OA _eprint: https://www.tandfonline.com/doi/pdf/10.1080/23311916.2018.1502242.

[22] E. National Academies of Sciences, *Evaluation and Selection of Airport Capital Project Delivery Methods.*

[23] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*, pp. 2296–2301, IEEE. ISSN: 0743-1619.

[24] T. Toledo and D. Zohar, “Modeling duration of lane changes,” vol. 1999, no. 1, pp. 71–78.

[25] S. Haufe, J.-W. Kim, I.-H. Kim, A. Sonnleitner, M. Schrauf, G. Curio, and B. Blankertz, “Electrophysiology-based detection of emergency braking intention in real-world driving,” *Journal of Neural Engineering*, vol. 11, p. 056011, Oct. 2014.

[26] Y. Xing, C. Lv, W. Huaji, H. Wang, and D. Cao, “Recognizing driver braking intention with vehicle data using unsupervised learning methods.” ISSN: 0148-7191, 2688-3627.

[27] “Speed management in sweden.”

[28] A. K. Budhkar and A. K. Maurya, “Characteristics of lateral vehicular interactions in heterogeneous traffic with weak lane discipline,” vol. 25, no. 2, pp. 74–89.

[29] G. J. Forkenbrock and A. Snyder, “NHTSA’s 2014 automatic emergency braking test track evaluations,”