POLITECNICO DI TORINO

Master of Science in Automotive Engineering

Master's Degree Thesis

Design and implementation of a trajectory planning algorithm using Dynamic Programming method on a Hybrid Electric Vehicles

Tutor

Prof. Giovanni Belingardi

Prof. Carlo Novara

Prof. Ali Emadi

Co-Supervisor

Ing. Pier Giuseppe Anselma

Candidate

Waiyuntian Lou

December 2019

ABSTRACT

In the recent years the automotive industry is in a revolution towards the electrified and intelligence mobility. The autonomous driving, which is the possible solution for the traffic congestion, accidents, and emission issues, is quickly becoming one of the hottest topics in the automotive field. However, the technology is still not mature enough for commercialization due to several technical challenges. The trajectory planning is one of them. Many researches regarding the trajectory planning have already been developed in the robotic field, however when it comes to the automotive field, it becomes extremely challenging due to the critical requirements on the system's real-time performance and on the trajectory generation with multi-objective optimization(Collision avoidance, occupant's comfort, fuel economy, etc). The two targets are usually contradictory since the planning algorithms with better optimization performance will degrade the real-time performance. In this thesis an approach is proposed to solve this problem. First an off-line optimization based on Dynamic Programming method is implemented on a Hybrid Electric Vehicle to obtain the optimum control sequence considering multi-optimization targets. Then a rule-based planner is established based on the results of the off-line optimization to realize the real-time application. Finally, the results obtained from different algorithm are compared (PID, MPC, Polynomial, DP, rule-based) and it proved that this approach could be a feasible solution for the contradictory requirements on real-time and optimality of the trajectory planning algorithm.

Key words: Trajectory planning, Multi-Objective Optimization, Hybrid Electric Vehicle, Dynamic Programming, Rule-based planner

Contents

List of figures

List of tables

Nomenclature

HEV	Hybrid Electric Vehicle
DP	Dynamic Programming
ADAS	Advanced Driving Assistance System
RRT	Rapidly exploring Random Tree
DOF	Degree of Freedom
BAS	Belt Alternator Starter
ICE	Internal combustion engine
SOC	State of Charge
EPS	Electric power steering
MPC	Model Predictive Control
PID	Proportional Integral Derivative
LKA	Lane Keeping Assist
ACC	Adaptive Cruise Control
PEARS	Power-weighted Efficiency analysis
IVD	Inter-vehicular distance
ECM	Equivalent circuit model

Chapter 1 Introduction

1.1 Introduction: Autonomous driving technology

Since the 1970s, the research on the autonomous driving technology began in countries such as the United States, Germany, etc. In 1984 the Carnegie Mellon University launched the Autonomous land vehicle project intended to build vision and intelligence for an autonomous vehicle[1]. In 2004, the United States Defence Advanced Research Projects Agency organized the DARPA Grand Challenge pushed the development of the autonomous driving technology[2]. In the recent decades with the revolution towards the intelligent mobility and electrified vehicles in the automotive industry. The autonomous driving technology is becoming one of the hottest topics nowadays. On May 7, 2019 GM Cruise Secures $1.15 billion of additional investment from T.Rowe Price, SoftBank Vision Fund, and Honda, Increasing its post-money valuation to $19 billion[3]. Ford claimed to have fully autonomous vehicle in commercial operation by 2021[4]. There are no doubts that the autonomous driving technology is becoming one of the most intensive research and development fields in the automotive industry.

There are several reasons why the autonomous driving technology is drawing such attention. First the autonomous driving technology can achieve a better fuel economy. The research of Taiebat shows that the automated vehicle can reduce the fuel consumption by optimizing the driving cycle[5]. According to Payre and Luettel, the autonomous driving cars can achieve a remarkable improvement on fuel economy through the optimization of the highway driving[6]-[7]. It would achieve 20%-30% fuel economy reduction implementing the platoon driving[8]. For instance, Volvo successfully realized partially automated highway platooning exploiting Cooperative Adaptive Cruise Control in California (CACC). Thanks to the V2V communication, it’s possible to have only one driver in the lead truck and realize the braking of all the vehicles at the same time. It enables vehicles to follow closer, not only reducing the aerodynamic drag, but also allows better highway utilization.

Second, it’s also beneficial for the overall social benefits. A study from INRIX revealed that Americans lost 97 hours a year due to congestion causing nearly $87 billion in 2018, which is an average of $1,348 per driver. The autonomous driving might be the possible solution. Ross pointed out that the autonomous driving can reduce the required cars to 15% of the current amount[9] and decline 25% percent of the of the current parking space[10]. Also, the idea of car-sharing service could possible reduce the number of vehicles needed and reduce the congestion.

Third the autonomous driving could be an ideal solution for the traffic accidents. World Health Organization reported that vehicle crashes kill more than 1.35 million people worldwide each year. While National Highway Traffic Safety Administration reported that the root of more than 90 percent of all crashes are caused by human error. According to AASHTO (American Association of States Highway and Transportation Officials), in 2006 16,000 serious lane departure accidents occured causing in total 25,082 death[11]. While the Lane Keeping Assisted System (LKAS) can effectively reduce the probability of the lane departure accidents and reduce the cost causing by the accidents[12]. The vehicles equipped with standard Automatic Emergency Braking (AEB) can achieve a reduction of striking rear-end crashed in 50km/h around 55%[13]. The advantages of autonomous driving system in reducing accidents are obvious. The processer could achieve a responding time faster than the human brain in several orders of magnitude, which could avoid or reduce the severity of accidents in conditions where emergency maneuver is needed. For example, the system could start to brake as soon as an emergency braking maneuver is required. And the stop distance is drastically reduced compared with the human driver. Also, the combination of sensors can achieve a perception ability much better than human. For instance, the radar is less influence by the weather. Thus, in critical conditions like fog or rain, the radar can perceive the obstacle in front which are not visible for human eyes. Theoretical speaking, the autonomous driving system has the potential to achieve much better performance in terms of safety.

However, the autonomous driving technology still has many challenges, making it difficult for commercialization. Some researchers pointed out that the autonomous vehicle will increase the difficulty for the courts, regulators, public to identify the responsibility of an accident involving a robot car[14] and cause serious legislation and insurance challenges[15]-[16]. Second, the high cost of the AV platform is another hinder for large-scale market adoption. For instance, Shchetko notes that the Light Detection and Ranging(LIDAR) systems might cost up to $85,000 each[17]. With large scale production, it's possible to reduce the cost of AV platform. Velodyne, one of the largest supplier for the Lidar, launched a $4,000 cost reduction for its most popular lidar sensor, the VLP-16 Puck thanks to its new megafactory in San Jose. While its original price was $8,000 when it first went on sale in 2016[18]. However, the current cost of the AV technology is still far to the customer's expectation. A survey from J.D. Power and Associates found that 37% of the persons would purchase a vehicle equipped with autonomous driving capabilities in their next vehicle. Though after being asked to assume an additional $3000 purchase cost, this share dropped to 20%[19]. Thus, further research and development is needed in order to cut down AV's technology cost. Third, security is another challenge for the connected and automated vehicles. In 2015, fellow white-hat hackers Charlie Miller and Chris Valasek remote hacked a Jeep. They managed to control the steering and even disable the brake[20]. Different attacks are also possible from the communication channel as well as sensor tampering of a connected vehicle[21]. Thus, it's necessary that the automotive manufacturer take a series of countermeasures to guarantee the cyber security of the connected vehicles.

In general, autonomous driving is a technology with huge potentials. It might be the solution for the safety, fuel economy, traffic congestion problems. However, further research effort is needed in the autonomous driving field to solve the current challenges.

1.1.1 Autonomous driving system structure

Generally, the autonomous driving system can be divided into three layers. The perception layer, the decision-making layer and the control layer. Shown as follows,

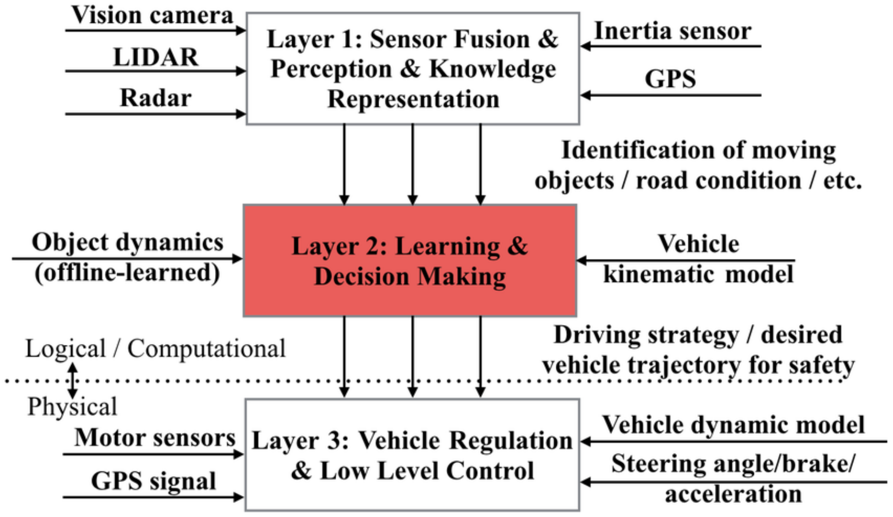

Vision camera
LIDAR
Radar
Layer 1: Sensor Fusion & Perception & Knowledge Representation
Inertia sensor
GPS
Identification of moving objects / road condition / etc.
Object dynamics (offline-learned)
Layer 2: Learning & Decision Making
Vehicle kinematic model
Logical / Computational
Physical
Driving strategy / desired vehicle trajectory for safety
Motor sensors
GPS signal
Layer 3: Vehicle Regulation & Low Level Control
Vehicle dynamic model
Steering angle/brake/ acceleration

Figure 1 Autonomous driving system structure

In the autonomous driving system, the environment information is gathered by the sensors. Typical sensors that autonomous vehicle equipped are camera, Radar(Radio Detection and Ranging), Lidar(Light Detection and Ranging), IMU(Inertia measurement unit), GPS(Global positioning system). Each sensor has its own advantages and disadvantages.

The function of the perception layer is to process the signals from the sensors (Radar, lidar, camera, GPS, IMU) and generate reliable information of the environment for the decision-making layer. For example, classifies the obstacle, monitors the road lane, identifies the traffic signals, etc. The technologies used in this layer includes image processing, sensor fusion, object tracking, environment modeling, simultaneous localization and mapping (SLAM) etc.

Then the decision-making layer will decide the behavior of the vehicle. The decisions can be divided into behavior planning, path planning, trajectory planning. The behavior planning layer will decide the current behavior of the vehicle according to the information from the perception layer. A common approach is the Finite State Machine method. For instance, for a Lane Keep Assist system, if the steering wheel torque sensor receive a large torque indicating the driver is willing to control the vehicle, then the lane keep assist system will enter the Off state to avoid interference with the driver. Once the behavior planning layer decide the maneuver task the vehicle need to perform, the trajectory planning layer will generate the corresponding trajectory to achieve the task. This paper focus on the trajectory planning layer, thus it will be introduced in detail in the next chapter.

Once the decision-making layer generated the target trajectory. The control layer will generate the throttle, brake pedal, and steering wheel commands so the vehicle could follow the target trajectory. During the trajectory generation vehicle dynamic constraints should be considered. The trajectory should be smooth enough so the vehicle can follow. Second a path tracking controller need to be properly designed. Different types of tracking algorithm are implemented, such as PID(proportional-integral-derivative), MPC(Model-predictive-controller), fuzzy controller, etc. A typical controller is the PID controller which is a negative feedback controller. The control signal is based on the error between the target trajectory from the decision-making layer and the actual trajectory obtained from the perception layer. At each time instant the PID controller sends control signals to the Electric Power Steering controller or the throttle/brake controller, which will adjust the vehicle’s current position. Then the vehicle current trajectory will be captured by the sensors, processed by the perception layer and then used to calculate the path error in the next control iteration. Other control algorithms like Model Predictive Control are also used, but there's always a trade-off between performance and computational time. Thus, the PID controller is still popular for its computational efficiency and robustness.

1.1.2 SAE levels

In order to define the level of automation, the Society of Automotive Engineer(SAE) formulate the standard in SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles[22]. It's shown as follow in table 1.

L	Desccription	Longitudinal and lateral vehicle motion control	Object and event detection and response	Dynamic driving task fallback	Operational design domain
0	Driver perform the entire DDT	Driver	Driver	Driver	N/A
1	The system performs partially the longitudinal and lateral vehicle motion control(ex.Adaptive cruise control, lane keeping assist)	Driver and system	Driver	Driver	Limited
2	The system performs the longitudinal/lateral vehicle motion control at the same time	System	Driver	Driver	Limited
3	The system performs the whole DDT, the DDT fallback is on the driver	System	System	Driver	Limited
4	The system performs the whole DDT, and the DDT fallback as well	System	System	System	Limited
5	Full driving automation	System	System	System	Unlimited

Table 1 SAE levels

In the table the OEDR(Object and event detection and response) means monitoring the driving environment and executing an appropriate response to such objects and events. For example, the vehicle should be able to detect the pedestrian suddenly crossing the road the brake the vehicle to avoid collision. While the DDT(dynamic driving task) includes the longitudinal and lateral motion control and the OEDR as well. The DDT fallback means the response when a DDT relevant system failure occur. For example, for Level 3 system the DDT fallback is on the driver, which means if a DDT relevant failure occurs, the system should warn the driver and the driver is responsible for taking control of the vehicle. While the Level 4 system the DDT fallback is on the system, which means the system itself has the ability to handle the DDT relevant failure and the driver is not responsible to handle the system failure. Thus, the level 4 automation has critical requirements on the reliability and robustness of the system. The typical solution includes increasing the redundancy(exploits more sensors) or use more expensive sensor(Lidar). However, either way will increase the cost of the system which is a challenge of the current technologies. K.P. Divakarla proposed an intermediate level 3.5 between level 3 automation and level 4 automation[23]. For the level 3.5 automation the primary DDT fall back is on the driver, if the driver failed to take control of the vehicle then the secondary DDT fall back is on the system. It's a smart solution for the potential risks of the level 3 automation since it's risky to ask the driver to control and release control of the vehicle frequently. Finally the ODD(Operational design domain) means the driving automation function designed operating condition. For the level 4 automation the vehicle can only drive autonomously in a certain region, while the level 5 automation the vehicle doesn't have this restriction.

1.2 Motion planning

1.2.1 Motion planning introduction

The main work of this thesis focused on the motion planning layer. In this chapter the current technologies of the motion planning are introduced. Planning means the decisions the autonomous vehicle made in order to travel from a certain starting point to a certain destination. The self-driving vehicle should avoid the obstacle while at the same time optimize the trajectory to achieve safety, comfort, fuel efficiency, maneuver efficiency, etc. The trajectory is generated based on the environment information obtained from the perception layer, and the generated trajectory is used as the reference path for the path tracking layer.

Usually the motion planning is divided into global and local planning. Global planning focuses on planning the shortest path from starting point A to destination point B. An example is the google map. It's not necessary to plan the speed trajectory in the global planning. While for the local planning the planner will generate an exact trajectory the vehicle will follow during the maneuver, for example the trajectory of lane changing, overtaking, emergency collision avoidance, etc. In the local planning not only the path trajectory but also the speed trajectory is determined. The factors that influence the trajectory includes collision avoidance, occupant's comfort, fuel efficiency, etc.

1.2.2 Motion planning State-of-the-art

Usually the most commonly used motion planning techniques can be divided into the following groups: graph search, sampling based, interpolating curves.

Graph search planners: The map which contains the starting point and destination is first discretized into lattice. Then the graph search algorithm will traverse the lattice from the starting point till the destination. Finally the cost is summarized and the best the path is the path with the lowest cost(For example minimum distance). Some typical graph search-based planners are introduced as follows.

Dijkstra Algorithm: The Dijkstra algorithm was forward by Dutch computer scientist Edsger W. Dijkstra in 1959[24]. The algorithm is widely used in finding the shortest path from the starting point A to the destination B. A simple example is shown in Figure 1.

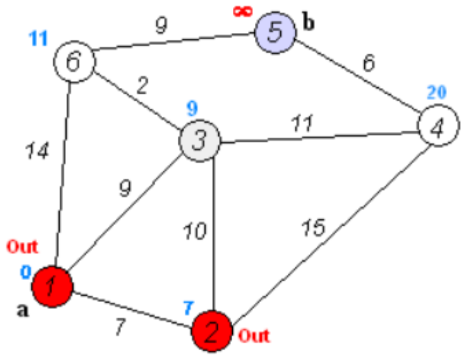


Figure 2 Dijkstra's algorithm example[25]

The nodes can be classified into unvisited set and visited set. The edge weights between each node are already known. At the beginning all the nodes are in the unvisited set, and the node's value are infinite.

First the starting node a will be add to the visited set and removed from the unvisited set. Since the distance of the starting node is 0, which is lower than infinite, thus the node value will be assigned as 0.

Next the values of the nodes close to node A are evaluated. If the cost is lower than the value of the node, then the node's value will be assigned as the total cost to reach that node. After that the node with the lowest value will be selected as the next current node and it will be added into the visited set and removed from the unvisited set.

In general, the basic idea of Dijkstra is that: the optimum solution of the partial path is independent from and will be part of the optimum solution of the whole path. For example, if the shortest path form node 1 to node 6 is 1->6 rather than 1->3->6, then in the final solution if the path starts from node 6, the previous path before 6 must be from 1->6. It tries to find the global optimum with the help of the local optimum.

However, the searching space of Dijkstra algorithm is large. Since the information of the destination is not used so the search process is quite inefficient. In order to solve this problem, the A* algorithm is forwarded.

A* Algorithm: The A* algorithm is the extension of Dijkstra's algorithm, thanks to the implementation of the heuristics, it reduces the calculation time compared with Dijkstra's algorithm. The main equation is shown as follows,

$$f(n)=g(n)+h(n)$$

In this equation n means node n, while g(n) is the actual cost from the original node to the node n, h(n) is the estimation of the optimum cost from node n to the destination node, f(n) is the estimation cost from the initial node to node n. The searching process is similar to Dijkstra's algorithm. The difference is that for Dijkstra's algorithm the next node gets explored is the node with the minimum actual cost while for A* the next node gets explored is the node with the minimum estimated cost, which is the sum of actual cost from initial node to node n and the estimation cost from node n to the destination. A simple example is that the estimated cost h(0) is the distance between node n and destination node. Then the estimation cost will penalize the node farther from the destination and thus the node closer to the destination are likely to get explored. The searching process is more target orientated and the searching time is reduced.

State Lattice Algorithm: Since the vehicle is a non-holonomic robot which means it's still under a certain constraint for example it cannot have translate movement perpendicular to the vehicle heading position. Thus, we should consider the dynamic constraints in the motion planning phase. M. Pivtoraiko proposed a novel approach to constrained path planning in 2005[26]. The planning space is a grid of hyper-dimensional states. For instance apart from the translational coordinates(x,y), the searching states also include the information of vehicle heading θ and the curvature κ. The problem can be described as finding the optimum solution from the initial set of states to the final set of states shown as follows,

$$[x_i, y_i, \ \theta_i, \ \kappa_i] \ \rightarrow \ [x_f, y_f, \ \theta_f, \ \kappa_f]$$

By implementing the state constraints, for example the limit variation rate of the heading θ, only a few states can be reached starting from the current states. Then vehicle dynamic constraints can also be considered and it's possible to make sure the generated trajectory is smooth enough which can be followed by the vehicle. It's also possible to implement a cost function to evaluate the cost of the path between each state. By evaluating the total cost it's possible to find the trajectory with minimum cost which is the optimum trajectory. Also, it's possible to implement the searching algorithm we described before in the state lattice. For example, M. Rufli and R. Y. Siegwart implemented the d* search algorithm to time-based planning on lattice graph[27].

Sampling Based Planners: Generally speaking, the graph-based search first discretizes the configuration space, next connect the adjacent lattice with edge based on a certain optimum path searching algorithm. However, with increasing the number of the state variables the searching space will increase exponentially, thus the graph searching process will take too long time. Sampling based planner is a possible solution since a random lattice is picked instead of the adjacent lattice.

Probabilistic Roadmap: The probabilistic roadmap method is proposed by L. E. Kavraki, P. Svestka in 1996[28], which then became the most popular path planning algorithm in the robotics field. The process is shown in the following figure. This method includes two phases, the first phase is to construct the roadmap. According to the map, several lattices are randomly selected and then connected to the closest lattice. To choose the closest lattice a distance function is implemented as follows,

$$D=\sqrt{(\Sigma_i(p_i - q_i)^2)}$$

In which p_i and q_i are the i-th state of the state lattice p and q. Then the lattice with the shortest distance are selected as the next lattice to be connected. While to connect the new lattice with the closest lattice a local deterministic planner is used. A common way of the local planner is to select a set of lattices evenly distributed on the straight line between the two lattices to be connected. Then a collision check function is used to check whether there is an obstacle on this path. If there is no obstacle on this path, then the two lattices are connected. This procedure will be repeated for several times until the map is dense enough. Once the path network is generated, the designated starting point and destination are added into the graph and connected to the existed network, then a certain graph-based searching algorithm(ex.Dijkstra) is implemented to find the optimum path between the starting point and the destination, this phase is called the query phase. The advantage of probabilistic roadmap searching method is that it's faster to find the possible path between the starting point and the destination since it doesn't search all the adjacent lattice of the previous lattice, also the cost of constructing the roadmap can be amortized by running multiple queries. Thus, the probabilistic roadmap method is quite suitable for operating robot in the same environment. The disadvantage is that it cannot guarantee to find the optimum path.

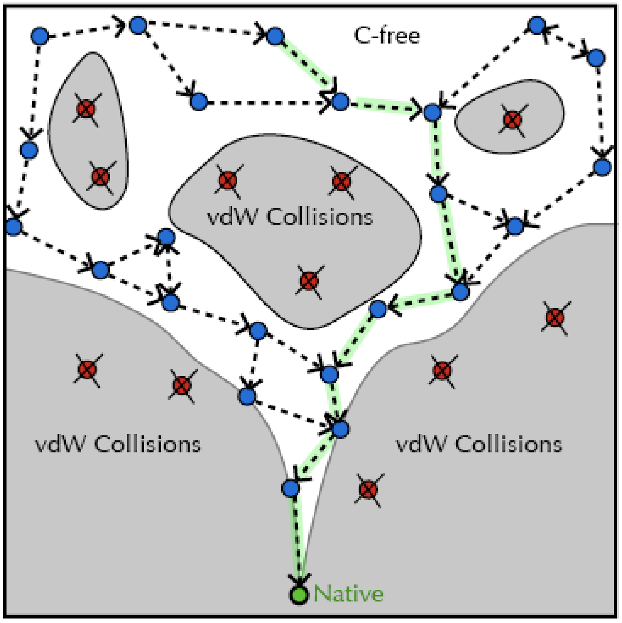


Figure 3 Probabilistic Roadmap trajectory planning algorithm [28]

Rapidly exploring Random Tree: In 2001 S. M. LaValle and J. J. Kuffner proposed the rapidly exploring random tree method [30]. Similar as probabilistic roadmap, first a random lattice is selected in the configuration space which is called seed. Instead of connecting the new lattice with the nearest lattice, a path will be extended from the existed lattice towards the new lattice for a pre-defined length, the end lattice of this path is called a child node, then this node is added onto the tree. The RRT method encourages to extend the path to cover the whole searching space in a short time. However, it's difficult to find the optimum path from the initial point to the destination.

RRT*: Based on RRT, RRT* method is proposed by S.Karaman[31]. This method could solve the problem introduced above. First a lattice is randomly selected from the configuration space like RRT. Then instead the new child node is generated like RRT. Then each one of the nodes laid within a neighborhood of the newly added child is inspected. If the cost from the root to one of these nodes is lower, then this node will be the parent of the newly added child node. Thus, the path generated by RRT* will be much smoother since the algorithm tries to optimize the tree branches in each simulation step. However due to the reconnection process, the computational resources needed by RRT* is also larger. In case the optimum path is not required, RRT might be a better solution. The searching results of RRT and RRT* are shown as follows, it's clear that the graph generated by RRT* is more organized than RRT.

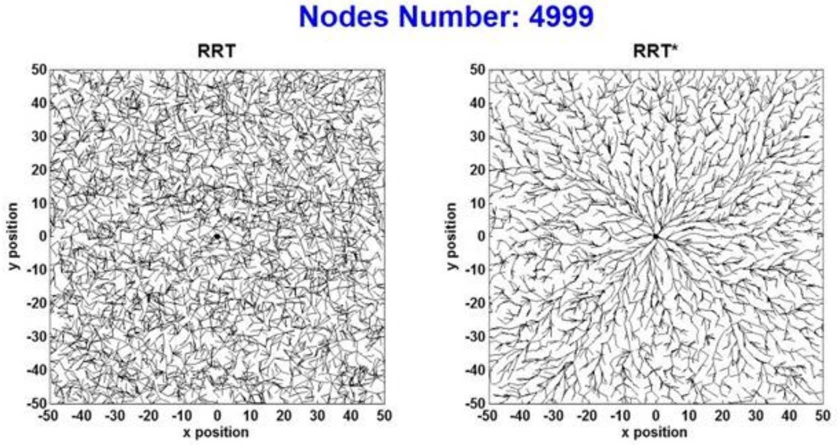


Figure 4 RRT RRT* comparison [32]

Interpolating curve planners: Instead of planning the path between two lattices, this set of planners will generate a segment of trajectory with pre-defined function. The parameters of the function are then tuned to fit the constraints of the two lattices (For example to guarantee the continuity of the connecting point). According to the different interpolating function used, commonly used algorithm includes *polynomial curves* [33], *Bézier Curves* [34], *Clothoid Curves*[35], *Spline Curves*, etc. The interpolating curve planners have advantages of low computational cost, since the segment of the trajectory is defined once the two lattices are defined. However, it cannot guarantee that the trajectory is the optimum and it's less versatile for complex real driving scenario. And also, it can generate the trajectory only if the initial and final condition are given, while it’s still necessary to find out the optimum final condition(displacement, velocity, acceleration, etc.)

Chapter 2 Dynamic programming

2.1 Dynamic programming introduction

Dynamic programming is an effective tool introduced by American mathematician Richard Bellman in 1953[36]. It's an effective tool which guarantees to find the global optimum solution of a complex problem with multiple states, constraints, and control inputs. However, the computational complexity increases exponentially with increasing the number of the state variables. Thus, the DP method is not suitable for real-time application (For example real-time path planning algorithm).

The dynamic programming function can be described by the following equations,

$$\dot{x}(t) = F(x(t), u(t), t) \qquad (1)$$

$$min\, J(u(t)) \qquad (2)$$

$$J(u(t)) = G\left(x(t_f)\right) + \int_0^{t_f} H(x(t), u(t), t)dt \qquad (3)$$

Equation(1) is the state update equation of the system, which means the states in the next time step is a function of the current states and the control input at the current time steps. While equation(2) shows the optimization goal, which is to find the best control serials u(t) to minimize the cost function J(u(t)). Equation (3) demonstrates that the cost function consists of two terms. The first term $G\left(x(t_f)\right)$ means the cost on the final states. While the second term represents the cost generated during each step.

In order to describe the system, we also need to implement constraints shown as follows,

$$x(t_f) \in [x_{f,min}, x_{f,max}] \qquad (4)$$

$$x(t) \in [x(t)_{min}, x(t)_{max}] \tag{5}$$

$$u(t) \in [u(t)_{min}, u(t)_{max}] \tag{6}$$

The equation(4) shows the states constraint on the final states, while equation(5) and (6) shows the constraints on states and control inputs at each time step.

For the problems with the state constraints on its final states, some researchers proposed the concept of the backward-reachable space[42]. Only the states lay inside the backward reachable space could reach the final states within the constraints, thus it's unnecessary to investigate the states outside the backward reachable region. This way, not only the computational effort is reduced but also the numerical errors caused by the states outside the backward-reachable region is reduced. As shown in the following figure, starting from the final states at k=N, only part of the states in the previous time instance are reachable.

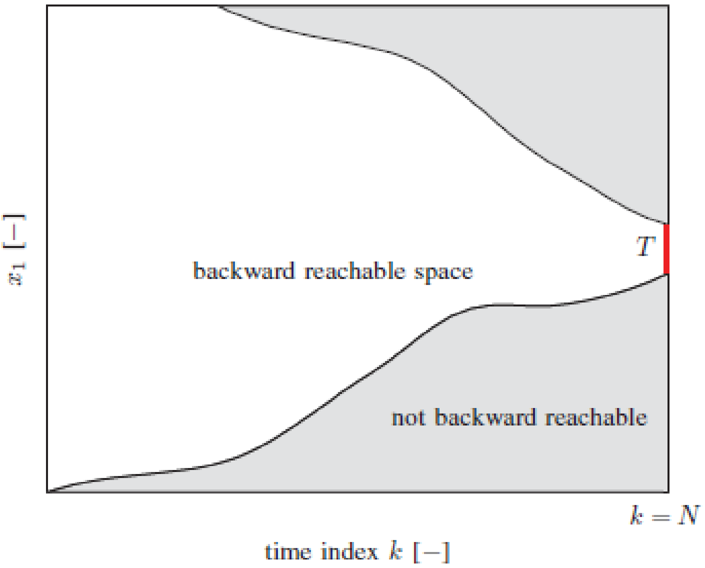


Figure 5 DP backward reachable space [42]

The backward evaluation process can be described as follows,

$$\mathcal{J}_N(x^i) = g_N(x^i) + \phi_N(x^i) \tag{7}$$

For k = N-1 to 0,

$$\mathcal{J}_k(x^i) = \min\{h_k(x^i, u_k) + \phi_k(x^i) + \mathcal{J}_{k+1}\left(F_k(x^i, u_k)\right)\}$$

(8)

The equation (7) is used to calculate the state cost on the final state while the equation (8) is the cost from states at step k to the final step N. The possible states of the previous step are evaluated from the final states within the constraints of the states and input. While the cost of each step is also evaluated backward from the final step. In equation (8) $h_k(x^i, u_k)$ represents the cost of implement control u_k on the *i*-th state at *k*-th step while $\phi_k(x^i)$ represents the cost of constraints on *i*-th state at *k*-th step. Finally, the term $\mathcal{J}_{k+1}(F_k(x^i, u_k))$ is the summation of the cost from the final step back to the *k+1*-th step. Once $\mathcal{J}_{k+1}(F_k(x^i, u_k))$ is calculated and stored, at each step we only need to calculate the cost at each step.

2.2 Dynamic programming application

Dynamic programming is a powerful tool to find the optimum solution of complex problem with multi-constraints on states and multi input. A typical example is the famous Lotka-Volterra fishery problem[39]. At the beginning a certain amount of fishes n_0 are in the lake. Everyday u_k amount of fish is caught from the lake. The relationship between the variation of the amount of fish and fishing volume is a non-linear function shown as follows in equation (9), then the amount of fish inside the lake in the k+1 time step can be described using equation (10), which in a general sense is considered as the state update equation,

$$f(x_k, u_k) = T_S \cdot (\frac{2}{100} \cdot \left(x_k - \frac{x_k^2}{1000}\right) - u_k) \quad (9)$$

$$x_{k+1} = f(x_k, u_k) + x_k \quad (10)$$

Then the dynamic programming method can be used to determine a control serial u_k at each day so that we've got the maximum total fishing volume. This target can be described by the objective function shown as follows in equation (11)

$$min \sum_{k=0}^{N-1} -u_k \cdot T_s \quad (11)$$

Another example is using DP to calculate the best torque split ratio control for the hybrid electric vehicle under given driving scenario to achieve the minimum fuel mass flow[40]-[41]. The state-of-charge (SOC) is the only state variable denoted as x_k. The longitudinal powertrain is modeled including the engine, electric motor, battery, vehicle longitudinal model, etc. With a given driving cycle (For example Japanese 10-15 driving cycle, NEDC), the operational condition of the vehicle is pre-defined. For example, the vehicle longitudinal speed v_k, longitudinal acceleration a_k, gear ratio i_k at each step k are known. Then the state update equation is shown in equation(12) shown as follows,

$$x_{k+1}=f(x_k, u_k, v_k, a_k, i_k)+x_k \tag{12}$$

Then the optimization target is expressed in equation (13), aiming to find the best power split ratio control serial to minimize the fuel mass consumption.

$$J=\sum_{k=0}^{N-1} \Delta m_f(u_k, k) \cdot T_s \tag{13}$$

In general, the characteristic of these two examples are: both of them are off-line optimization with future operation condition, disturbances already known, the computation time is quite long and not suitable for on-line application. They could find the global optimal control serial with the optimization target which can be used as a benchmark for other real-time controllers.

Chapter 3 Implementation: off-line optimization

3.1 Off-line optimization introduction

Since the dynamic programming method is used to find the global optimum solution for a complex multi-constraint, multi-control input problem. Many studies have already been done regarding the single state problem (For example fishery problem, power split ratio control to achieve the optimum fuel efficiency). P. Elbert and S. Ebbesen proposed the level-set dynamic programming method which is possible to solve the optimization problem accurately with n-dimension in an acceptable amount of time[42].

The trajectory planning problem can be also considered as a multi-states optimization problem with constraints on states at each step and on the final step, and with multiple optimization targets(For example maneuver efficiency, comfort, fuel economy, etc.) However, it's not feasible to implement this method due to the computational complexity as the trajectory planning algorithm which has strict real-time requirements.

In this chapter, the trajectory planning problem is abstracted into a multiple-states, multiple constraints optimization problem and the feasibility to implement the DP method in the trajectory planning is verified. The approach contains three steps, first a constant speed overtaking scenario is investigated to verify the collision avoidance ability of the algorithm. Next the longitudinal speed variation is also considered to study the overtaking efficiency and longitudinal comfort of the ego vehicle. Finally, an Adaptive Cruise Control scenario is studied and the longitudinal powertrain of the vehicle is also included and the fuel economy is also considered as the optimization objective.

Next in Chapter 4 other common algorithms including PID, MPC, polynomial are also established for the same given scenario to compare the performance of the DP-offline simulation algorithm.

All these optimization problems are solved off-line which requires a long computational time, then in the next chapter a method is proposed to realize the real-time implementation.

3.2 DP implementation for overtaking scenario

3.2.1 Driving scenario introduction

First a simple overtaking scenario is investigated to study the obstacle avoidance ability of the trajectory planning algorithm using dynamic programming method. The driving scenario is shown as follows in figure 6.

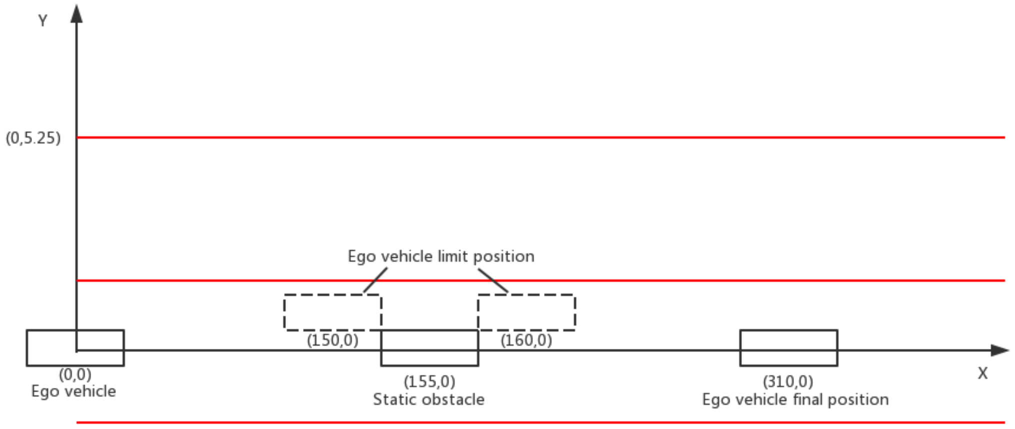


Figure 6 Obstacle avoidance scenario

The ego vehicle is travelling on a road with constant vehicle speed, and at this stage the speed variation of the ego vehicle is not considered. The static obstacle in front of the ego vehicle is in the same road lane with a certain initial distance. The initial and final position of the ego vehicle are known. The dash lane indicates the limit position of the ego vehicle for collision avoidance. The target of the ego vehicle is to perform a double-lane change maneuver to overtaking the obstacle in front and go back to the center of its original road lane.

3.2.2 Vehicle lateral dynamic model

In the first approach, in order to guarantee that the trajectory generated can be tracked by the vehicle, a 2 d-o-f vehicle lateral dynamic model is considered during the trajectory planning phase.

Since in the highway overtaking scenario, the tire's side-slip angle (The angle between the speed of the tire and the equatorial plane of the tire) is large and thus the bicycle model is no longer suitable to describe the vehicle lateral dynamics. Thus, the two degree of freedom model is introduced as follows.

Following figure shows the reference frame of the vehicle. The global reference frame XY is a inertial reference frame which is fixed. While the local reference frame x-y is a non-inertial reference frame move with the vehicle. Thus, the inertial forces should be considered during establishing the dynamic equation in the x-y reference frame.

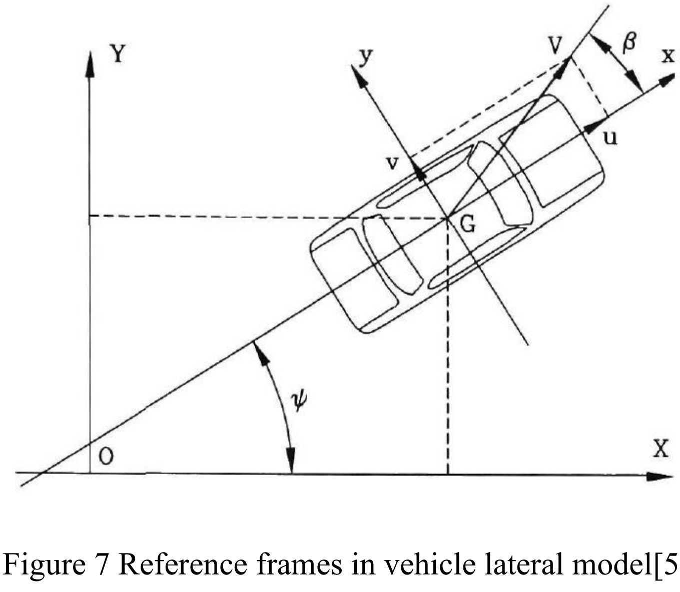


Figure 7 Reference frames in vehicle lateral model[54]

The dynamic equations considering the longitudinal/lateral acceleration and yaw moment are established as follows,

$$F_x = m(\dot{u} - r \cdot v) \tag{14}$$

$$F_y = m(\dot{v} + r \cdot u) \tag{15}$$

$$M_z = J_z \cdot \dot{r} \tag{16}$$

In this equation *u, v* represent the longitudinal and lateral vehicle speed along the local reference frame *x* and *y* direction. While *r* is the yaw rate of the vehicle.

In high speed scenario the lateral force exerted by the tire is proportional to the side slip angle, the equations of the lateral force are shown as follows,

$$F_{yf} = -C_f \cdot \alpha_f \tag{17}$$

$$F_{yr} = -C_r \cdot \alpha_r \tag{18}$$

The parameters all indicated is shown in the following figure, C_f and C_f are the cornering stiffness of the front and rear tires.

Given the side slip angle β and the yaw rate $\dot{\psi}$. It's possible to estimate the side-slip angle of the tire $\boldsymbol{\alpha}$[54], which is necessary to estimate the self-aligning torque of the tire. The geometric relationship between the front wheel center and the vehicle gravity center is reported as,

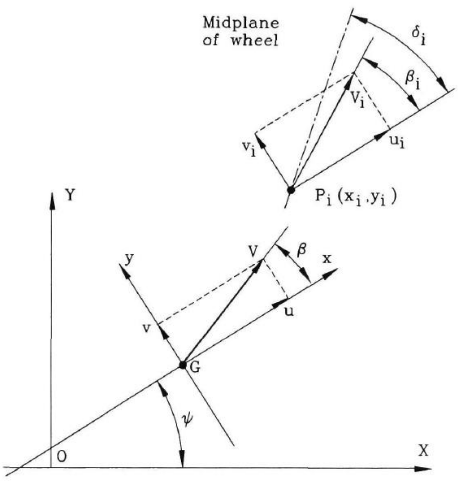


Figure 8 Front tire center vehicle gravity center relationship[54]

where,

- $XY -$ global reference frame
- $xy -$ vehicle local reference frame
- $G -$ vehicle gravity center
- $P_i -$ wheel ground contact center of i-th wheel
- $x_i, y_i -$ coordinate of the wheel center in the local reference frame(*m*)
- $V -$ vehicle speed(*m/s*)
- $u -$ vehicle center velocity component in longitudinal direction(*m/s*)
- $v -$ vehicle center velocity component in lateral direction(*m/s*)
- $V_i -$ speed of i-th wheel(*m/s*)
- $u_i -$ vehicle center velocity component in longitudinal direction(*m/s*)
- $v_i -$ vehicle center velocity component in lateral direction(*m/s*)
- $\beta -$ vehicle side slip angle(*rad*)
- $\beta_i -$ angle between v_i and *x* axis
- $\alpha_i -$ side slip angle of i-th wheel(*rad*)
- $\delta_i -$ steering angle of i-th wheel(*rad*)

Given the vehicle speed V, side slip angle β, and yaw rate $\dot{\psi}$, the velocity of the contact area of the i-th wheel, is

$$\overrightarrow{V_{P_i}} = \overrightarrow{V_G} + \dot{\psi}\Lambda\overline{(P_i - G)} \tag{19}$$

The velocity component of the i-th tire is,

$$v_i = v + \dot{\psi}x_i \tag{20}$$

$$u_i = u - \dot{\psi} y_i \tag{21}$$

The angle of the *i*-the tire β_i is,

$$\beta_i = \arctan\left(\frac{v_i}{u_i}\right) = \arctan\left(\frac{v+\dot{\psi}x_i}{u-\dot{\psi}y_i}\right) \tag{22}$$

Since the term $\dot{\psi}y_i$ is smaller than the vehicle longitudinal speed u in the order of magnitude, and β_i is a relatively small angle. The equation can be simplified as,

$$\beta_i = \frac{v+\dot{\psi}x_i}{u} = \beta + \frac{\dot{\psi}x_i}{u} \tag{23}$$

Finally, the side slip angle of *i*-th wheel is,

$$\alpha_i = \beta_i - \delta_i = \beta + \frac{\dot{\psi}x_i}{u} - \delta_i \tag{24}$$

where x_i is the front axle distance *a* and under the estimation of (25) it's possible to rewrite equation (24) in the form of equation (26),

$$u = V cos\beta \approx V \tag{25}$$

$$\alpha_i = \beta_i - \delta_i = \beta + \frac{\dot{\psi}}{V} \cdot a - \delta_i \tag{26}$$

Finally, the two degree of freedom lateral dynamic model is obtained.

$$mV(\dot{\beta} + r) = -C_f \cdot (\beta + \frac{l_f \cdot r}{V} - \delta) - C_r \cdot (\beta - \frac{l_r \cdot r}{V}) \tag{27}$$

$$J_z \cdot \dot{r} = -C_f \cdot l_f \cdot (\beta + \frac{l_f \cdot r}{V} - \delta) - C_r \cdot l_r \cdot (\beta - \frac{l_r \cdot r}{V}) \tag{28}$$

Finally the state space equation can be derived from equation (27) and (28) shown in equation (29), δ is the steering wheel angle which is the control input of the system while β and r are the vehicle's side slip angle and yaw rate which are the states of the system.

$$\begin{pmatrix} \dot{\beta} \\ \dot{r} \end{pmatrix} = \begin{bmatrix} -\frac{C1+C2}{mV} & \frac{C2b-C1a}{mV^2} - 1 \\ \frac{C2b-C1a}{Jz} & -\frac{C1a^2+C2b^2}{JzV} \end{bmatrix} \begin{pmatrix} \beta \\ r \end{pmatrix} + \begin{bmatrix} \frac{C1}{mV} \\ \frac{C1a}{Jz} \end{bmatrix} (\delta) \tag{29}$$

3.2.3 Dynamic programming method implementation (Constant speed scenario)

Under the assumption that the longitudinal speed is constant, the longitudinal position of the vehicle at each time instant is known thus it's not necessary to introduce one state variable to track the longitudinal displacement. While it's necessary to track the lateral displacement since the vehicle should change the road lane and avoid the obstacle. In order to guarantee the trajectory generated can be tracked by the vehicle the vehicle lateral dynamic model is also considered during the trajectory planning phase.

Since in highway driving scenario the side slip angle β is small. Thus, the following assumption is established,

$$V_x = V \cdot cos\beta \approx V \tag{30}$$

$$V_y = V \cdot sin\beta \approx V\beta \tag{31}$$

The state space equation is shown as follows, the state variables are side slip angle β, yaw rate *r*, lateral displacement *Y*. The control input is the steering wheel angle at the front wheel δ. *k* stands for the state variables at *k*-th simulation step.

$$\begin{pmatrix} \dot{\beta_k} \\ \dot{r_k} \\ \dot{Y_k} \end{pmatrix} = \begin{bmatrix} -\frac{C1+C2}{mV} & \frac{C2b-C1a}{mV^2}-1 & 0 \\ \frac{C2b-C1a}{Jz} & -\frac{C1a^2+C2b^2}{JzV} & 0 \\ V & 0 & 0 \end{bmatrix} \begin{pmatrix} \beta_k \\ r_k \\ Y_k \end{pmatrix} + \begin{bmatrix} \frac{C1}{mV} \\ \frac{C1a}{Jz} \\ 0 \end{bmatrix} (\delta_k) \tag{32}$$

Thus, the state update equation is shown as follows,

$$\begin{gathered} \beta_{k+1} = \beta_k + \dot{\beta_k} \cdot \Delta T \\ r_{k+1} = r_k + \dot{r_k} \cdot \Delta T \\ Y_{k+1} = Y_k + \dot{Y_k} \cdot \Delta T \end{gathered} \tag{33}$$

Since the dynamic programming method is a numerical optimization process, the problem, the state variables and the control inputs should be properly discretized. The problem time length is set as T, the discretization number is set as N, thus the time step can be derived as follows,

$$\Delta T = \frac{T}{N} \tag{34}$$

While the state constraints for each state at each time instant and the final time instant should be properly selected, as well as the constraints on the control input,

$$x(t) \in [x_{min}(t), x_{max}(t)] \tag{35}$$

$$x(t_f) \in [x_{f,min}, x_{f,max}] \tag{36}$$

$$u(t) \in [u_{min}(t), u_{max}(t)] \tag{37}$$

In order to describe the obstacle in front of the ego vehicle, a time variant constraint is implemented for the lateral displacement state variable. The following constraints on state Y described that when the longitudinal displacement of the ego vehicle X is close to the longitudinal position of the obstacle, the lateral displacement of the ego vehicle should be larger than the upper boundary of the obstacle in Y direction in order to avoid collision.

$$\text{if } \mathrm{X} \in [0, \mathrm{X_{obs}}) \quad \mathrm{Y} \in [\mathrm{Y_{lower}}, \mathrm{Y_{upper}}] \tag{38}$$

$$\text{if } \mathrm{X} \in [\mathrm{X_{obs}}, \mathrm{X_{obs}} + \mathrm{L_{obs}}] \quad \mathrm{Y} \in [\mathrm{Y_{obs,upper}}, \mathrm{Y_{upper}}] \tag{39}$$

$$\text{if } \mathrm{X} \in (\mathrm{X_{obs}} + \mathrm{L_{obs}}, \mathrm{X_f}] \quad \mathrm{Y} \in [\mathrm{Y_{lower}}, \mathrm{Y_{upper}}] \tag{40}$$

Finally, the cost function is specified as follows. Here $\Delta\beta_k$ is used to represent the optimization of the occupant's comfort and α_1 is the weight coefficient of this term.

$$\mathrm{C}{=}\alpha_1 \cdot |\Delta\beta_k| = \alpha_1 \cdot |\dot{\beta}_k \cdot \Delta T| \tag{41}$$

3.2.4 Dynamic programming method implementation (Varied speed scenario)

Many researches have been done regarding the trajectory planning in the lane change or overtaking scenario. Usually the longitudinal speed variation is not considered by assuming a constant longitudinal speed of the ego vehicle. However, in actual scenario usually the driver will decide to accelerate the vehicle to reduce the overtaking time. Some studies consider the trajectory planning and speed planning as two separate problem. First the trajectory is generated without considering the vehicle speed. Then the speed trajectory is planned based on the existed vehicle path. For instance, the speed is planned on the s-t graph shown as follows,

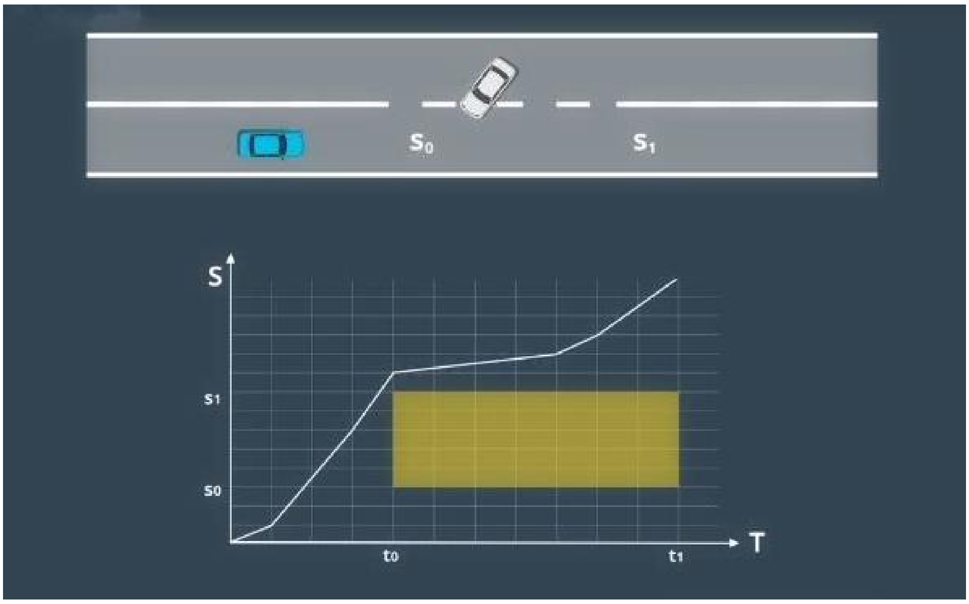


Figure 9 S-T graph for speed planning

In this case the ego vehicle is travelling on a straight road, thus the generated path trajectory is a straight line. S stands for the station, which is the distance that the ego vehicle travelled along the path trajectory. While T stands for the time. In this scenario the white vehicle will occupy the path trajectory of ego vehicle between station S0 and S1 during time interval from t0 to t1, which is represented as the yellow region in the S-T graph. Thus, the speed planner of the ego vehicle should generate a speed trajectory avoid the yellow region. The result is the ego vehicle tends to accelerate and travel pass S1 before t0 to avoid the obstacle.

Since the path trajectory is planned without considering the vehicle speed, the global optimum solution is not found. Matthew McNaughton and Chris Urmson proposed the spatiotemporal lattice planner which could consider the path trajectory and speed trajectory planning at the same time[43]. However, at each state lattice a set of velocities are assigned in order to reduce the searching space. Then a secondary optimization process is implemented to find the best speed profile. The real-time application is guaranteed after reducing the searching space however it still cannot guarantee to find the global optimum solution.

For scenario considering varied longitudinal speed, vehicle's position in the longitudinal direction is not known. Thus, additional states should be added into the state space to track the vehicle's longitudinal displacement. The longitudinal displacement X is added as an additional state variable. However, if only the longitudinal displacement is monitored and the longitudinal speed is considered as a control input, the algorithm might return solution jerky speed profile with large acceleration. Thus, it's also necessary to include the longitudinal speed V_x as another state variable and the longitudinal acceleration as a control input. By implementing the boundary limit on the longitudinal acceleration, we can guarantee the speed trajectory is smooth enough which is able to be followed by a vehicle.

However, adding two more state variables will result a system with five state variables. It's critical to notice that the searching space and computational time increases exponentially with increasing the number of state variables. Five state variables might result a simulation time more than ten hours. Thus, it's important to avoid unnecessary state variables. In the first attempt, the vehicle dynamic model is considered during the trajectory generation phase to guarantee the trajectory can satisfy the vehicle dynamic. However, another possible solution is to generate a trajectory smooth enough in the planning phase without considering the vehicle dynamics, then a trajectory tracking algorithm and a vehicle dynamic model are developed to verify that the trajectory can be followed by a vehicle.

Thus, in order to reduce the computational time, the vehicle dynamic model is not considered during the trajectory planning phase. The state variable yaw rate is cancelled and instead of the steering wheel angle, the variation rate of the side slip angle is the control input of the system. The state update equations are shown as follows,

$$V_{x,k+1} = V_{x,k} + a_{x,k} \cdot \Delta T \tag{42}$$

$$X_{k+1} = X_k + V_{x,k} \cdot \Delta T \tag{43}$$

$$\beta_{k+1} = \beta_k + \dot{\beta}_k \cdot \Delta T \tag{44}$$

$$Y_{k+1} = Y_k + \beta_k \cdot V_{x,k} \cdot \Delta T \tag{45}$$

Since now the it's able to control the longitudinal speed variation of the ego vehicle, thus the longitudinal comfort and overtaking efficiency can be considered as the cost term influencing the trajectory generated. The cost function is shown in the following equation,

$$C{=}\alpha_1 \cdot |\Delta Y| + \alpha_2 \cdot \left|a_{x,k}\right| + \alpha_3 \cdot \left|\dot{\beta}_k\right| \tag{46}$$

In the cost term $|\Delta Y|$ represent the overtaking efficiency. As long as the vehicle is not back into the original road center this term will add cost, thus it encourages the vehicle to finish the overtaking maneuver faster. While the term $\left|a_{x,k}\right|$ is the longitudinal acceleration representing the longitudinal comfort. The term $\left|\dot{\beta}_k\right|$ is used to describe the lateral comfort. α_1, α_2 and α_3 are the weight coefficients of each term.

3.3 DP implementation for ACC scenario

The trajectory planning considering the obstacle avoidance, lateral comfort and longitudinal comfort have been discussed in the previous chapter. However, for the real-time trajectory planner it's quite difficult to consider the fuel economy. In this chapter, the longitudinal powertrain model of a hybrid electric vehicle is established and the fuel consumption is taken into consideration as a influencing factor of the trajectory generation.

3.3.1 Longitudinal powertrain modelling

In order to consider the fuel economy, it's necessary to model the longitudinal powertrain so that the energy consumption at each vehicle operating condition can be derived (Longitudinal speed, acceleration, gear number, etc.). The P0-P4 hybrid electric vehicle longitudinal powertrain of the McMaster University EcoCar competition team is selected as the reference in this research. The architecture of the powertrain is shown as,

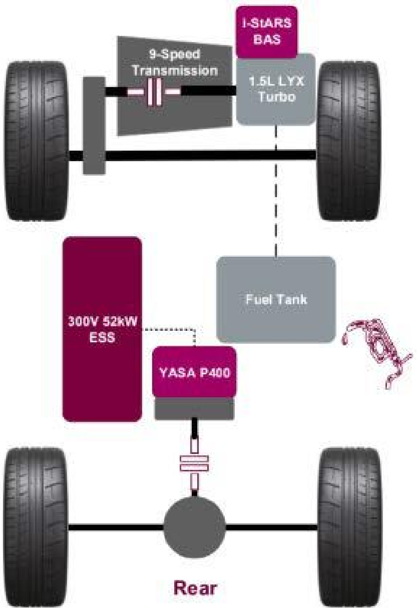


Figure 10 P0-P4 Hybrid electric powertrain

The vehicle is equipped with a 1.5L turbo engine and a Belt Alternator Starter (BAS) directly connected with the engine shaft. The BAS will work in the electric motor mode and drive the engine crank shaft. While in driving phase, it will generate an auxiliary torque to adjust the engine torque to its optimum efficiency region. And it will either charge or discharge a 12V low voltage battery.

The rear axle is connected to the YASA P400 motor. During performance driving if high torque command is received from the acceleration pedal which exceed the torque limit of the turbo-engine and the BAS, the YASA motor will engage and supply the rest of the torque requested.

The block diagram of the longitudinal powertrain is shown as,

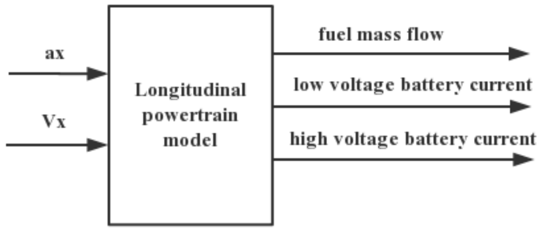

ax
Vx
Longitudinal powertrain model
fuel mass flow
low voltage battery current
high voltage battery current

Figure 11 Longitudinal powertrain model

The inputs are the longitudinal acceleration and longitudinal speed at each operating condition, while the outputs are the fuel mass flow, low voltage battery current, and high voltage current. The outputs are used to evaluate the energy consumption.

Longitudinal vehicle model: In order to evaluate the engine fuel mass flow, it's necessary to investigate the engine torque, which usually starts from the torque at the wheel. The longitudinal vehicle dynamic equation is described as,

$$R = \left(mgcos(\alpha) - \frac{1}{2}\rho V^2 SC_z\right)(f_0 + kV^2) + \frac{1}{2}\rho V^2 SC_x + mgsin(\alpha) \qquad (47)$$

$$\frac{T_{wheel}}{R_e} = \left(R + m_{equ} \cdot a_x\right) \qquad (48)$$

R is the term representing the total resistance force, in which the term $f_0 + kV^2$ representing the rolling coefficient, $\frac{1}{2}\rho V^2 SC_x$ is the aerodynamic drag. In the longitudinal dynamic equation, m_{equ} is the total equivalent mass considering the inertial of the rotating component, which is described in equation as follows.

$$m_e = m + \frac{J_\omega}{R_e^2} + \frac{J_t}{R_e^2 \tau_f^2} + \frac{J_e}{R_e^2 \tau_f^2 \tau_g^2} \qquad (49)$$

where J_ω, J_t, J_e are the moment of inertia of the wheel, the moment of inertial of the powertrain and of the engine. While τ_f and τ_g are the gear ratios of the final transmission and the gearbox.

Engine model: Since engine is a complex mechanical and thermodynamic system, usually instead of modeling the engine in detail, the engine map is implemented to obtain the fuel mass flow. The engine map is a 2-D look-up table, given the engine rotation speed and engine torque it's possible to obtain the fuel mass flow rate $\dot{m}_f$

interpolating the look-up table.

Given the vehicle operating condition (longitudinal speed V, longitudinal acceleration a_x, gear ratio of gearbox τ_g), the engine operating condition can be directly obtained. The engine speed is reported as:

$$n_e = \frac{60 \cdot V}{2\pi \cdot R_e \cdot \tau_f \cdot \tau_g} \tag{50}$$

Belt Alternator Starter (BAS) model: BAS is an electric motor connected to the engine crankshaft, it's an economic solution which could add mild-hybrid feature to the powertrain. It can provide assist torque to compensate the engine torque and guarantee the engine is working in its optimum region. It can either work in the power assist mode or the regenerative braking mode. A 12V low voltage battery is connected to the BAS system and available for charging and discharging. Similar as the engine map, the electric motor is modeled using the motor map. Given the battery voltage, motor speed, and torque command, the battery current and motor torque can be directly obtained from the motor map. Since the BAS efficiency is different when working in motor mode or in alternator mode. It's necessary to implement two maps,

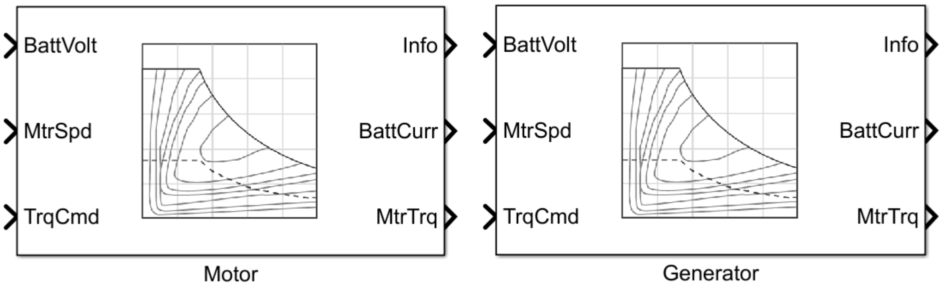


Figure 12 BAS motor map

Low voltage battery model: The equivalent circuits models (ECM) are adopted to model the low-voltage battery [45]. Since the Dynamic programming method requires a large computational effort, it's important to properly establish the model to reduce the computational complexity. The equivalent circuits models can capture nonlinear electrochemical phenomena while avoid lengthy electrochemical process calculations. Which made it the perfect candidate for system-level modeling. The equivalent circuit is shown as,

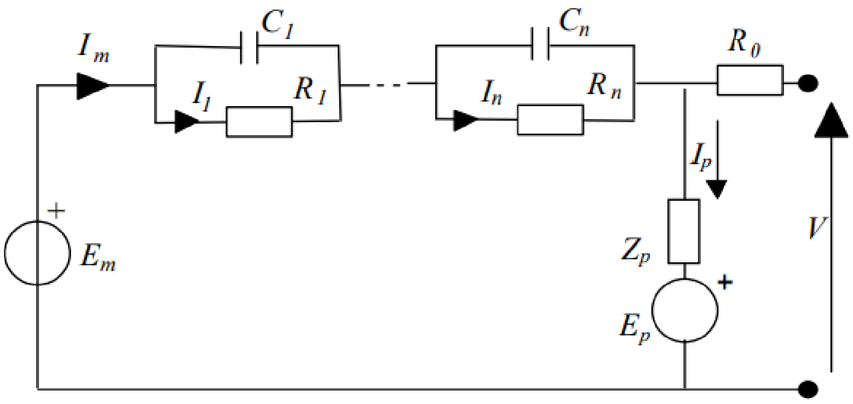


Figure 13 Equivalent circuit model for low voltage battery [45]

where,

- $E_m -$ Battery open-circuit voltage, $E_m = f(SOC, T)$
- $R_o -$ Series resistance, $R_o = f(SOC, T)$
- $C_{batt} -$ Series resistance, $C_{batt} = f(T)$
- $R_n -$ Network resistance, $R_n = f(SOC, T)$
- $C_n -$ Network capacitance, $C_n = f(SOC, T)$

The equivalent circuit is formulated in the way that several resistance-capacitance blocks are connected in serial with a series resistance. The resistance, capacitance, and the open circuit voltage are dependent on the battery state of charge and the temperature. Compared with modelling the electrochemical reaction in the battery in detail, the equivalent circuit can fit the experimental data quite well without increasing too much the computational effort.

The voltage on each RC pair V_n, output voltage V_T, SOC can be derived in equation the following equation,

$$V_n = \int_0^t (\frac{I_{batt}}{C_n} - \frac{V_n}{R_n C_n})\, dt$$

$$V_T = E_m - I_{batt} R_o - \sum_1^n V_n$$

$$SOC = \frac{-1}{C_{batt}} \int_0^t I_{batt} dt$$

The low voltage can be charged and discharged since the BAS can either work in electric motor mode or the generator mode. Given the battery current and temperature, the battery SOC and the battery voltage can be estimated.

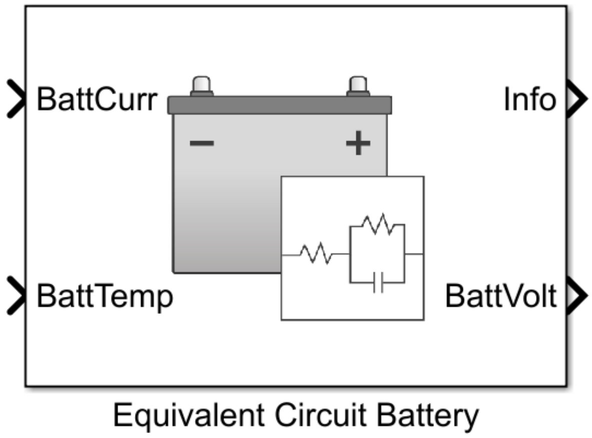


Figure 14 Simulink Equivalent circuit battery mode for low voltage battery

3.3.2 Torque split rules

For a hybrid electric vehicle with P0-P4 powertrain structure, it's critical to decide the torque split rules to achieve the optimum fuel economy. The function of the Belt-Alternator-Starter is to generate an assist torque to guarantee that the engine is working in the region with optimum fuel efficiency. While the YASA motor connected to the rear axle could supply torque command exceed the limit of engine and BAS in performance driving. The logic diagram representing the torque split rules is reported as follows,

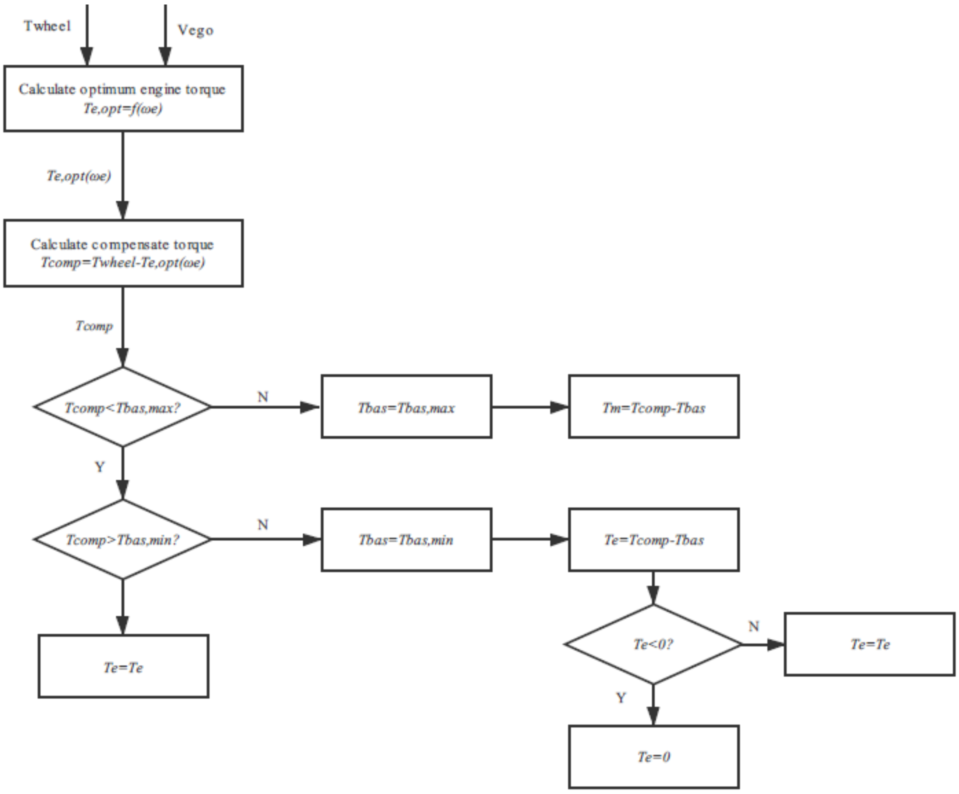


Figure 15 Torque split rules

Given the torque request at the wheel and the ego vehicle speed, the engine speed is obtained. Then the optimum engine torque is determined based on the engine speed. However the difference between the wheel torque command T_{wheel} and the and the optimum engine torque $T_{e,opt}$ should be compensated by the BAS and the YASA motor. Thus, after the theoretical optimum engine torque is obtained, the compensating torque T_{comp} is calculate and compared with the upper and lower torque limit of the BAS. If the compensating torque is larger than the upper limit of the BAS, it means the torque command at the wheel is too larger, the engine and BAS are not able to provide enough torque. Then the YASA motor will engage and provide the rest of the torque. While if the compensating torque is less than the lower limit of the BAS, the BAS will operate at its lower torque limit and the rest of the torque will be provided by the engine, which means it's not always feasible for the engine to work in its optimum region. It's also necessary to consider the torque limit of the BAS.

Once the torque command of the engine, BAS, and YASA motor are known, the fuel mass flow $\dot{m}_f$ and the current of the low voltage battery and high voltage battery can be determined using the motor and battery model discussed in the previous chapter. Finally, it's possible to evaluate the energy consumption.

3.3.3 Dynamic programming method implementation

Since the overtaking and lane changing maneuver are transient maneuvers and last relatively a short time which has a negligible impact on the fuel economy. Also, the safety is most important consideration in these scenarios. Thus, the Adaptive-Cruise-Control (ACC) scenario is a better candidate to study the speed trajectory planning considering the fuel efficiency.

In the ACC driving scenario, the lateral motion of the vehicle is not considered, thus only the states related to the longitudinal motion are considered, the state updating equation are shown as follows,

$$a_{x,k+1} = a_{x,k} + \dot{a_{x,k}} \cdot \Delta T \tag{51}$$

$$V_{x,k+1} = V_{x,k} + a_{x,k} \cdot \Delta T \tag{52}$$

$$IVD_{k+1} = IVD_k + (V_{lead,k} - V_{x,k}) \cdot \Delta T \tag{53}$$

First the acceleration is described as an additional state instead of input in order to limit the sharp variation of the acceleration. While the variation rate of the acceleration is the new input of the system. Then instead of longitudinal displacement of the ego vehicle, the relative displacement is monitored as a state. Since in ACC scenario only the relative displacement matters. It’s also possible to implement a finer discretization tracking the relative displacement which guarantee a better DP performance.

The energy consumption model has already been established in the previous chapter shown in figure 10, thus the fuel consumption is evaluated at each simulation step.

$$m_{fuel,k} = f(a_{x,k}, V_{x,k}) \tag{54}$$

The cost function is formulated in the following form. The first term represents the target to minimize the fuel consumption. The second term is used to limit the variation of acceleration, which is used to improve the comfort of the occupant. Here α_1 and α_2 also represent the weight coefficient of the fuel mass flow and the longitudinal comfort.

$$C{=}\alpha_1 \cdot \left|\dot{m_f}\right| + \alpha_2(if\,|a_{x,k+1} - a_{x,k}|{>}p) \tag{55}$$

Chapter 4 Dynamic programming performance comparison

In order to compare the performance of the dynamic programming method. A polynomial based path trajectory planning algorithm is established for the overtaking scenario. While for the ACC scenario, the PID and MPC control algorithm are established as the references for the DP controller.

4.1 Performance comparison for Overtaking scenario

The polynomial path planning algorithm is commonly used for highway scenario. Neville Hogan proposed the jerk minimizing theory [44]. It's proven that a jerk minimum trajectory should satisfy,

$$\frac{d^m s}{dt^m} = 0,\ \forall m \geq 6 \tag{56}$$

$$Total\ squared\ jerk = min(\int_0^{t_f} \dddot{s}(t)^2 dt) \tag{57}$$

Here *s* represents the longitudinal or lateral displacements, while m represents the order of derivation of the polynomial function s. According to the power series expansion, a function can be written in the following form,

$$s(t) = \int_{n=0}^{\infty} \alpha_n t^n \tag{58}$$

Thus, the function satisfying the minimum jerk trajectory is reported as follows, which can satisfy the requirements reported in equation (56)

$$s(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5 \tag{59}$$

$$\dot{s}(t) = \alpha_1 + 2\alpha_2 t + 3\alpha_3 t^2 + 4\alpha_4 t^3 + 5\alpha_5 t^4 \tag{60}$$

$$\ddot{s}(t) = 2\alpha_2 + 6\alpha_3 t + 12\alpha_4 t^2 + 20\alpha_5 t^3 \tag{61}$$

The polynomial equation has six unknown parameters and the unknown variable time t. Theoretically speaking, once the initial and final condition s_i, $\dot{s}_\imath, \ddot{s}_\imath$, s_f, $\dot{s}_f, \ddot{s}_f$, initial and final time instances t_i, t_f are given. It's possible to solve all the parameters. And solve the trajectory. However, it's possible to simplify the solving process by assuming that $t_i = 0,\ t_f = T$. The coefficient $\alpha_0,\ \alpha_1,\ \alpha_2$ are directly determined based on the initial condition,

$$s_i = s(0) = \alpha_0 \tag{62}$$

$$\dot{s}_i = \dot{s}(0) = \alpha_1 \tag{63}$$

$$\ddot{s}_i = \ddot{s}(0) = 2\alpha_2 \tag{64}$$

Substituting the parameters $\alpha_0,\ \alpha_1,\ \alpha_2$ in equation (59)-(61), given the maneuver duration T and the final condition, $s_f,\ \dot{s}_f, \ddot{s}_f$. It's possible to organize the equations above in the matrix form shown as follows, then the parameters $\alpha_3,\ \alpha_4,\ \alpha_5$ are solved inverting the matrix,

$$\begin{bmatrix} T^3 & T^4 & T^5 \\ 3T^2 & 4T^3 & 5T^4 \\ 6T & 12T^2 & 20T^3 \end{bmatrix} \times \begin{bmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} s_f - (s_i + \dot{s}_i T + \frac{1}{2}\ddot{s}_i T^2) \\ \dot{s_f} - (\dot{s}_i + \ddot{s}_i T) \\ \ddot{s_f} - \ddot{s}_i \end{bmatrix} \tag{65}$$

Thus, given the initial and finial position, speed, acceleration, and maneuver time, it's possible to solve the trajectory that minimize the jerk. However, the problem is how to find a good final state. Even the final lateral position is given by the decision layer, an appropriate lateral speed and acceleration are still unknown and need to be solved.

While for the advantage of the DP algorithm is that it could derive the global optimum trajectory with the information of speed and acceleration at each point. Thus, the DP-based trajectory planner might be an appropriate candidate to overcome the drawback of the polynomial algorithm as is introduced above.

Then the polynomial trajectory planning algorithm and the DP trajectory planning algorithm are established and compared. The maneuver efficiency which is represented by the maneuver duration, and the occupant's comfort, which is represented by the accumulation of acceleration and jerk, reported as,

$$a_{cum} = \int |y\ddot{(t)}|^2 dt \tag{66}$$

$$J_{cum} = \int |y(\dddot{t})|^2 dt \quad (67)$$

However, since the maneuver duration of the polynomial planning algorithm is a tunable variable. Longer duration will result lower maneuver efficiency but also a lower jerk accumulation which is good from the comfort point of view. How to specify the maneuver duration is critical for the performance comparison.

The approach adopted to solve this problem in this thesis is the control variants method. First the DP planner runs off-line to solve the optimum path trajectory, the initial, overtaking, final positions, and maneuver time of polynomial algorithm are set to be the same as the trajectory obtained from the DP algorithm. The lateral speed and acceleration of the overtaking position are set as zero since the polynomial trajectory cannot find the optimum speed and acceleration by itself. By setting the maneuver duration as the same, the maneuver efficiency of the two approaches is the same. Then the accumulation of jerk and acceleration are compared to compare the performances of these two algorithms in terms of comfort. And we could verify the optimality of the algorithms.

4.2 Performance comparison for ACC scenario

Typical control algorithm implemented in ACC scenario include PID control, MPC(Model Predictive Control) algorithm, fuzzy control, etc. Each type of control algorithm has its advantages and drawbacks.

4.2.1 Performance comparison PID algorithm

The PID control algorithm is still widely used in the industry with intuitive principle. It’s easy to satisfy the real-time requirement of the system due to its simplicity. The PID control algorithm is a suitable candidate for the ACC driving scenario. A common approach is the multi-mode PID control. When the ego vehicle is travelling without any vehicle in front or with a vehicle in front of large distance. The control variable is the pre-defined vehicle speed. When the ego vehicle approaches the lead vehicle in front, the cantor variable will change to the pre-defines safety distance.

Though it's adaptable to a wide range of system, the performance is not optimal when applying it to a specific system. For the Adaptive Cruise Control system, even the PID algorithm could simply satisfy the control target, it's difficult to consider the occupant's comfort and the fuel economy, which are the crucial factors influencing the customer's experience.

A predictive PID controller for ACC scenario is established as follows,

$$T_{wheel,tar} = k_p(IVD_{tar} - IVD_{pre}) \tag{68}$$

$$IVD_{pre} = \left(X_{lead} + V_{lead} \cdot t_{pre}\right) - (X_{ego} + V_{ego} \cdot t_{pre}) \tag{69}$$

The request torque at the wheel is proportional to the error between predictive inter-vehicular distance and the target inter-vehicular distance. The predictive inter-vehicular distance is the relative distance after the prediction time t_{pre} if both the vehicles are travelling at the same speed.

Given the target torque, the ego vehicle longitudinal acceleration is obtained and the speed is updated. Then the speed will be the feedback of the predictive PID controller to determine the target acceleration in the next time step. The block diagram of the system is shown in the following figure.

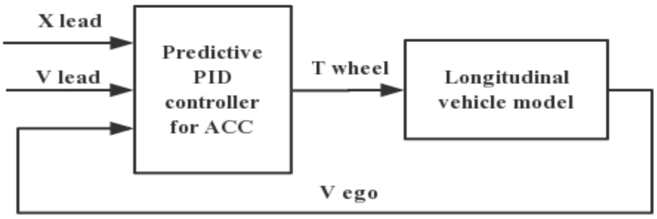


Figure 16 PID controller for ACC scenario

Eventually the same energy consumption model of the DP optimization is exploited here for the PID ACC controller to compare the performance. The whole system is reported as follows,

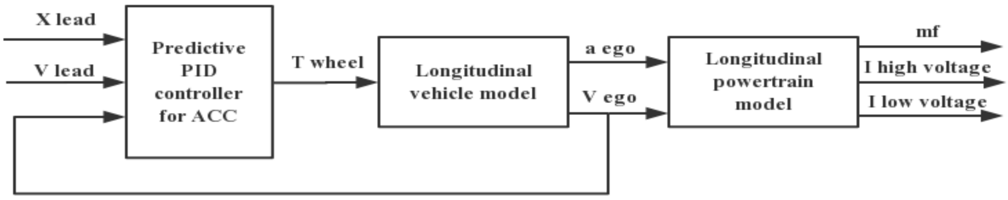

X lead
V lead
Predictive PID controller for ACC
T wheel
Longitudinal vehicle model
a ego
V ego
Longitudinal powertrain model
mf
I high voltage
I low voltage

Figure 17 Structure of PID ACC system

The whole system is realized in Simulink as,

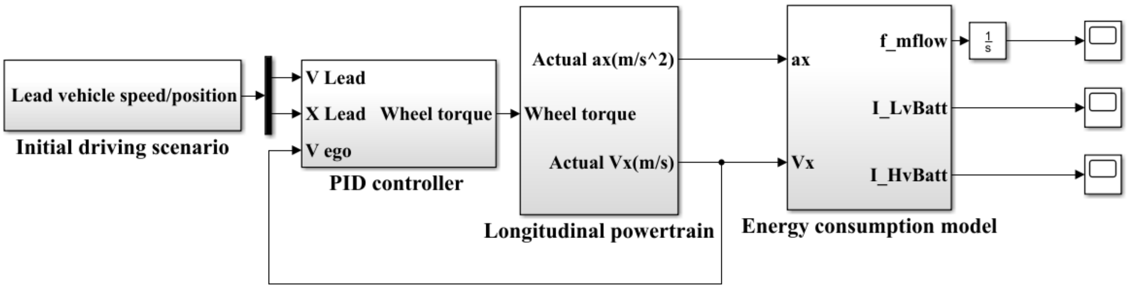


Figure 18 Simulink PID ACC control system

4.2.2 Performance comparison with MPC algorithm

MPC introduction: Generally speaking, the trajectory planning problem is an optimization problem with multiple constraints on each state and input, aiming to minimize the cost function. The MPC(Model-Predictive-Controller) algorithm is also a common solution for this category of optimization problem. Thus, a MPC controller is developed to compare with the DP controller.

MPC is an advanced control technology widely used in industry since 1980s[46]. This technology is popular in multiple industrial field such as chemical reaction control, energy efficiency control, flight control, satellite attitude control, blood glucose control, etc.[47].

Like PID controller, the MPC algorithm also exploits the results obtained from the control plant through a negative feedback loop. Then the control sequence is generated to reduce the difference between the current state variables and its reference value. However, for PID controller the optimality of the control sequence is not guaranteed since it's not possible to predict the future state of the plant. While the MPC controller could guarantee the optimality of the control since an internal model of a control system is evaluated at each time step to predict the future states of the system.

The logic chart of the MPC control algorithm is reported,

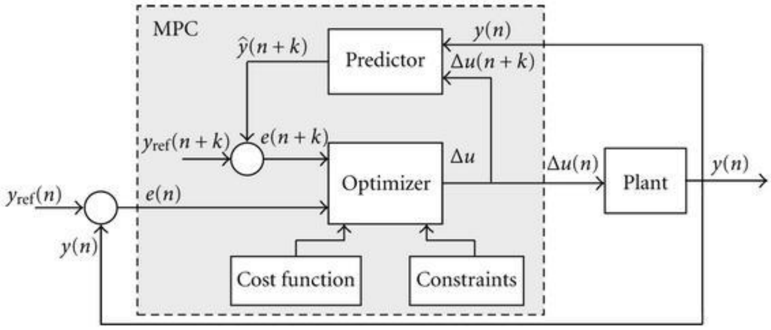


Figure 19 Structure of MPC controller

Like the other controller, the input of the MPC controller is the actual output of the plant at n-th time step $y(n)$, the target value of the output at n-th time step $y_{ref}(n)$, the control input at the n-th time step $u(n)$. Different from the PID controller, the MPC controller has a plant model(predictor), and an optimizer inside.

First of all, the problem is discretized into several step, the sampling time is T_S. All the states variables and the inputs, outputs are assumed to maintain the same value during each time step. The plant at n-th time step can be described using the state space equation as,

$$\dot{x}(k \cdot T_s) = f(x(k \cdot T_s), u(k \cdot T_s)) \tag{70}$$

$$y(k \cdot T_s) = h(x(k \cdot T_s), u(k \cdot T_s)) \tag{71}$$

The time period from current time step n till the (n+k)-th step is called prediction period. While the number of time step within the prediction period k is called the prediction horizon. Starting from the current time step n, a serial of control sequence is generated from n till n+k at each time step($u(n)$, $u(n+1)$,... $u(n+k)$) by the optimizer. Then a serial of estimated output $\hat{y}(n)$ to $\hat{y}(n+k)$ is evaluated using the internal plant model with the actual plant state $x(n)$ as the initial condition based on the control sequence from $u(n)$ to $u(n+k)$. From n-th to $(n+k)$-th time step, the error between the reference output y$_{ref}(n+k)$ and estimated output $\hat{y}(n+k)$ is evaluated as $e(n+k)$. Finally, the optimizer can evaluate a control serial from $u(n)$ to $u(n+k)$ that minimize the total cost during the prediction period using the cost function within the constraints on the states and the inputs. A typical example of cost function is shown as,

$$J_y = \sum_{j=1}^{n_y} \sum_{i=1}^{k} \{w_{i,j}[r_j(n+i) - \hat{y}_j(n+i)]\}^2 \tag{72}$$

Here,

- $n_y -$ number of outputs
- $k -$ prediction horizon
- $w_{i,j} -$ weight coefficient of j-th output on n+i-th step
- $r_j(n+i) -$ reference value of j-th output on n+i-th step
- $r_j(n+i) -$ reference value of j-th output on n+i-th step
- $\hat{y}_j(n+i) -$ estimated value of j-th output on n+i-th step

The calculated control sequence within the prediction horizon is the sequence that the minimum of the cost function is achieved,

$$z_n = [u(n)^T\ u(n+1)^T \ldots\ u(n+k)^T]^T \tag{73}$$

$$J_{y,min} = J_y(z_n) \tag{74}$$

This is a typical optimization problem with constraints on states can be considered as a Quadratic Programming (QP) problem. Different kinds of algorithm can be implemented to solve the optimization problem such as,

- interior point
- active set

- augmented Lagrangian

- conjugate gradient

- gradient projection

Once the control sequence z_n is evaluated, only the control signal at the current time step $u(n)$ is implemented to the plant. Then the state variables of the plant are updated and $x(n+1)$ is obtained. Then the optimization process will be repeated in the next time step.

In general, the results obtained from the MPC controller has a better performance compared with the PID controller since the future states of the model is predicted and the cost function is solved to obtained better control sequence. However, it still has several drawbacks as follows,

- The performance of the MPC controller depends on the fidelity of the plant model, which is usually difficult to obtain.

- The computational complexity of the MPC is high. It's applicable to slow dynamic process with high performance computer such as chemical plants and oil refineries. Nowadays with the increase of the computational power it's possible to implement the MPC on system with fast response requirements. However, the large computational power required by the algorithm is still a challenge.

- The MPC algorithm doesn't guarantee to find the global optimum solution since the control signal at time step $u(n)$ is derived by solving the optimization equation within the prediction horizon. While the DP method introduced above guarantees to obtain the global optimum solution. Thus, it's interesting to compare the results obtained from DP and MPC.

Establishing MPC controller: The same state space equation used to describe the adaptive cruise control scenario in the dynamic programming method is adopted in order to make the comparison, it's described as follows,

$$\dot{a_{x,k}} = u_k \tag{75}$$

$$\dot{V_{x,k}} = a_{x,k} \tag{76}$$

$$I\dot{V}D_k = V_{lead,k} - V_{x,k} \tag{77}$$

With the state variables, input, output reported as follows,

$$x = \{IVD, V_x, a_x\}^T \tag{78}$$

$$u = \dot{a_x} \tag{79}$$

$$y = IVD \tag{80}$$

In order to compare the performance of the DP controller, the term representing the fuel economy is included in the cost function of the MPC controller. Given the operation condition of the vehicle (*Vx, ax*), the fuel mass flow rate $m_{\dot{f}uel}$ can be obtained using the longitudinal powertrain model described in the previous chapter. The cost function for the MPC controller is shown as follows,

$$J = \sum_{i=1}^{k}\{\frac{w_y}{s_y}[r(n+i) - \hat{y}(n+i)]\}^2 + \sum_{i=1}^{k}[\frac{w_m}{s_m}(m_{\dot{f}uel})]^2 \tag{81}$$

here,

- $w_y -$ weight coefficient of the output error
- $k_y -$ scale factor of the output error
- $w_m -$ weight coefficient of fuel mass flow rate
- $k_m -$ scale factor of the fuel mass flow rate

The scale factor is introduced in the cost function. The scale factor is the range of a state variables, for instance for the scale factor s_y,

$$s_y = y_{upper\ limit} - y_{lower\ limit} \tag{82}$$

Since that different state variables might have difference in the order of magnitude. While the scale factor can scale different cost term into the same order of magnitude, which could avoid the dominance of a certain cost term due to its large order of magnitude.

While the weight coefficient can be used to emphasize the importance of a certain cost term, since it's common that two cost term has contradictory requirement on the control input. For example, from the comfort point of view a small acceleration is preferred while from the efficiency point of view a large acceleration is preferred to reduce the maneuver time.

Once the internal model and the cost function are specified, the MPC planner for vehicle speed trajectory in the ACC scenario is established as,

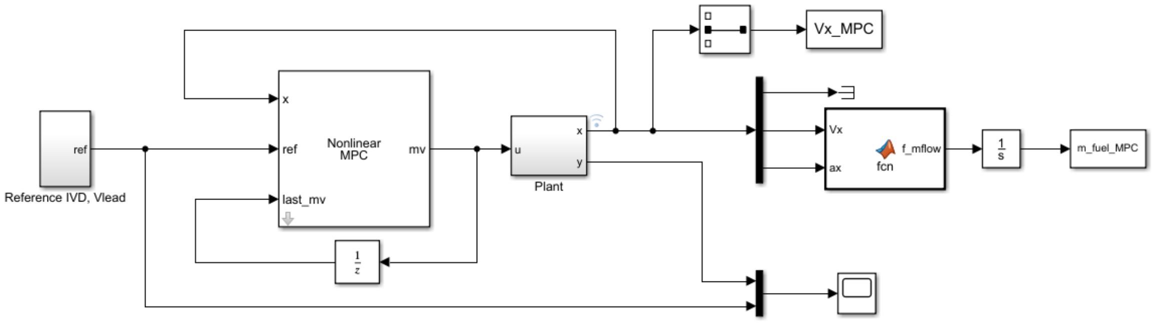


Figure 20 MPC planner for ACC

The input of the MPC planner is the target inter-vehicular distance, lead vehicle speed. The output is the variation rate of the longitudinal acceleration of ego vehicle. Then the longitudinal speed v_x and longitudinal acceleration a_x are used to evaluate the fuel mass flow rate $\dot{m}_f$. Then the fuel mass flow is integrated and the total fuel consumption is obtained.

Finally, the speed profile and the total fuel consumption obtained using the Dynamic Programming, PID, and the Model Predictive Control are plot together and the performance are compared. The results are shown in Chapter 5.

Chapter 5 Rule-based real time controller

5.1 Rule-based controller introduction and implicit MPC analogy

Since the principle of DP method is to search all the possible control solution and guarantee to find out the global optimal solution. A solution was forwarded for a similar problem in the Model Predictive Control application: The explicit MPC[48], which could be applied to realize the DP algorithm in this thesis as well.

The traditional MPC require running on-line optimization algorithms to solve the optimization problem at each time step. The computational effort for the MPC is large thus is has been labeled as a technology for slow processes. In field having critical real-time requirements, such as automotive field, the MPC application is quite limited.

The idea of the explicit MPC is that, instead of solving the optimum control using the MPC online, the optimum control for all the x(t) is computed off-line. Then the optimum control input for each state are recorded. Then the explicit relationship between the state and the control input are found out which could be then implemented in the real-time application. Thus, instead of solving the optimization problem on-line, the optimum control is directly searched based on the rules given the system's current state.

For instance, the relationship of the optimum control and system states obtained off-line can be plotted into a look-up table as,

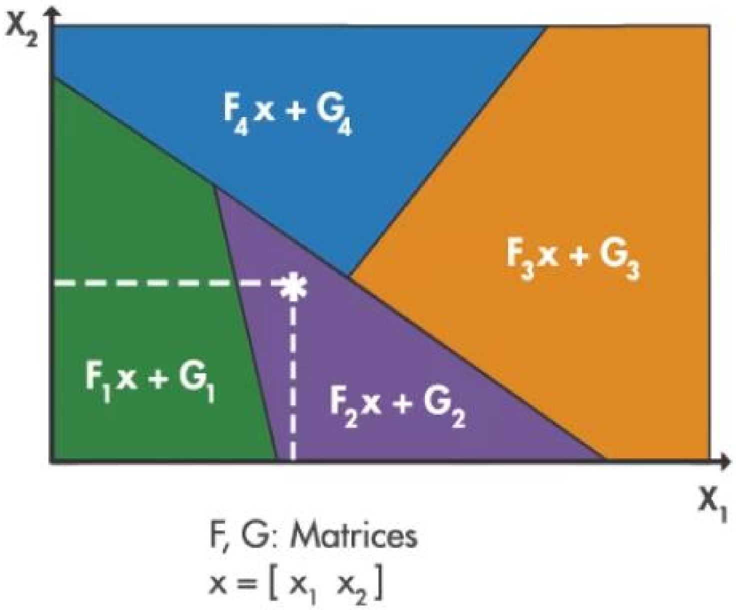


Figure 21 Explicit MPC look-up table[49]

During real-time implementation, for give states satisfying the inequity constraints, (the states are within the $i\text{-}th$ polyhedron as shown in figure above

$$A_0^i x \leq b^i \tag{83}$$

The optimum control inputs are obtained solving the linear equation, F_i and G_i are constants obtained from the off-line DP optimization.

$$u = F_i x + G_i \tag{84}$$

Once the states of the system are estimated, then it’s possible to derive the control exploiting the look-up table. The computational effort is reduced and it’s possible to implement the DP in the real-time application.

Thus, it’s also possible to adapt the principle of explicit MPC to the application using DP. First runs the DP optimization algorithms off-line and obtain the global optimum solution. Then a control rule is generated from the obtained data. For example, the rule between the states variables and the control input can be tabulated into a n-D look-up table. Finally, a real-time controller is established using the obtained rule and ready for real-time implementation. Several studies have implemented this method to build the real-time controller for the hybrid electric vehicle powertrain control to optimize the fuel consumption. Pier Giuseppe proposed a based real-time control algorithm for an HEV derived from an offline energy management strategy using the Power-weighted Efficiency Analysis for Rapid Sizing (PEARS) algorithm[50]. ZHANG Ya Hui introduced a hybrid dynamic programming-rule based algorithm to achieve energy optimization on plug-in hybrid electric bus[51], which exploiting the result obtained off-line from the dynamic programming method to update the control rules of the real-time controller.

Since nowadays the most common approach to guarantee real-time trajectory planner is to reduce the searching space to achieve the real-time target, and the global optimum solution is not guaranteed. In this paper the possibility of exploiting the results from the off-line optimization results to build the real-time controller for the trajectory planner is studied and verified.

5.2 Rule-based controller implementation for Overtaking scenario

With the results obtained from the dynamic programming optimization for overtaking scenario. A rule-based trajectory planner is established. In order to test the planner, a trajectory tracking algorithm and the vehicle models are developed. The model build in Simulink is shown as,

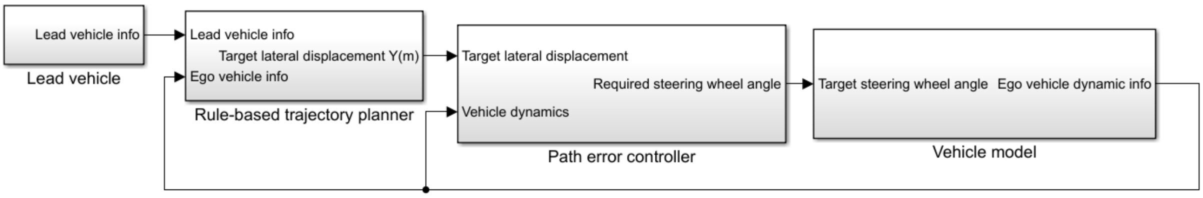

Lead vehicle info
Lead vehicle
Lead vehicle info
Target lateral displacement Y(m)
Ego vehicle info
Rule-based trajectory planner
Target lateral displacement
Required steering wheel angle
Vehicle dynamics
Path error controller
Target steering wheel angle
Ego vehicle dynamic info
Vehicle model

Figure 22 Model in loop test for real-time rule-based planner

Assuming the lead vehicle information (for example the lead vehicle speed, the inter-vehicular displacement) are known conditions from the perception layer. The rule-based real time trajectory planner will then plan the target lateral displacement at each time step based on the lead vehicle and ego vehicle dynamic information.

Then a predictive path error controller is established to control the vehicle model in order to track the trajectory generated by the trajectory planner.

Finally, the vehicle model is built. The electric power steering (EPS), 2 degree of freedom vehicle lateral model, and the tire model are established. The ego vehicle dynamic information will then be updated at each time step and will be the input of the rule-based planner in the next time step. The whole loop is closed to realize the feedback control.

5.2.1 Real time path planner

Since in the previous chapter the optimum control solutions in an overtaking scenario with different initial speeds and inter-vehicular distances are obtained using off-line dynamic programming method. With the known state history at each simulation step, it's possible to plot the lateral displacement and longitudinal acceleration with respect to the inter-vehicular distance and longitudinal speed difference. It’s shown as,

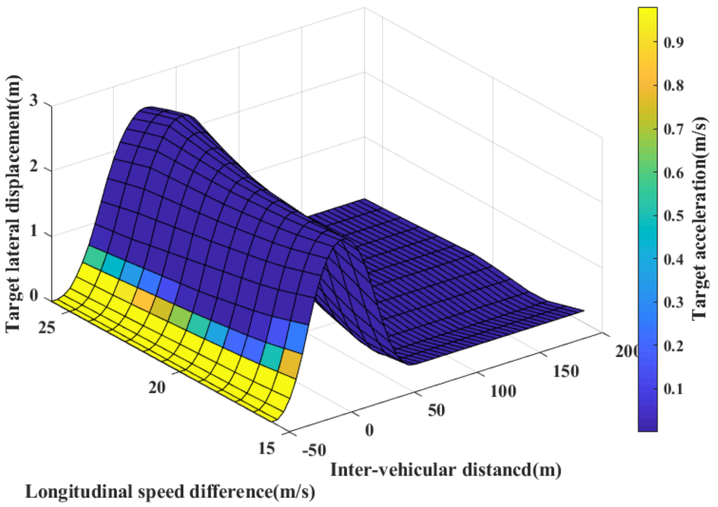


Figure 23 2-D look-up table for target lateral displacement and longitudinal acceleration

where, the z-axis represents the target lateral displacement, while the color indicates the target longitudinal acceleration at each operation point.

By interpolating the curve a 2-D look up table is tabulated. For each inter-vehicular distance and longitudinal speed difference a target lateral displacement can be found, corresponding to the optimum lateral displacement considering all the optimization target. While the color at each step represents the longitudinal acceleration of the ego vehicle.

Once the 2-D look up table is built, it's possible to implement it in the rule-based trajectory planner shown as follows. The input are the inter-vehicular-distance and speed difference between the ego vehicle and the obstacle. While the outputs are the target lateral displacement Y_{target} and the target longitudinal acceleration a_{target}.

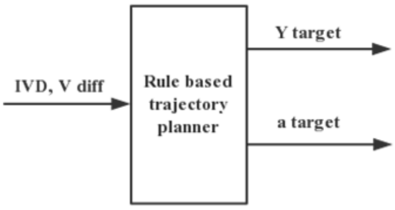


Figure 24 Real-time rule-based controller

Based on the current driving condition, the rule-based trajectory planner could generate the target lateral displacement and longitudinal acceleration profile in real-time at each time step. Then the planned trajectory will be the reference of the motion control layer.

5.2.2 Trajectory tracking algorithm

Since during the trajectory planning phase the vehicle dynamic model is not considered, the trajectory tracker and a vehicle dynamic model are established to verify that the trajectory can be followed by the vehicle.

The PID(Proportional-integral-derivative) control algorithm is adopted to build the tracking controller. For the lateral position control a predictive PID controller is adopted. The block diagram of the controller is shown as,

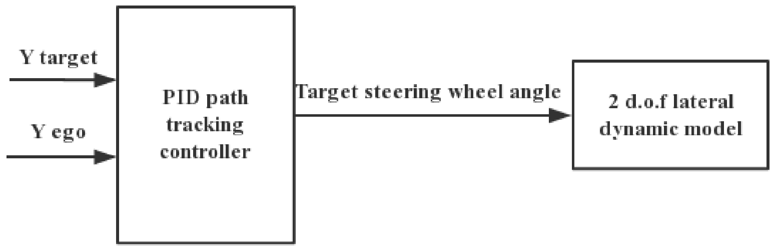

Y target
Y ego
PID path tracking controller
Target steering wheel angle
2 d.o.f lateral dynamic model

Figure 25 Predictive path tracking controller

The target steering angle is proportional to the error between the target lateral displacement and actual ego vehicle displacement.

Since the response of the lateral dynamic model has a delay, the correction will be too late if the target steering wheel angle is proportional to the current lateral displacement error. It's necessary to exploit a predictive PID controller. The equations are shown as follows,

$$Y_{pre} = Y_{current} + V_Y \cdot t_{pre} \tag{85}$$

$$\theta_{target} = K_y(Y_{target} - Y_{pre}) \tag{86}$$

The target steering wheel angle is proportional to the predictive path error instead of the current path error. The predicted lateral displacement is the lateral displacement after prediction time t_{pre} assuming the vehicle maintaining the current lateral speed. Finally, the predictive PID controller could generate a steering correction in real time to guarantee trajectory tracking ability.

5.2.3 Establishing control models.

In order to verify the effectiveness of the control algorithm, it’s necessary to establish the model of the control plant. In the overtaking scenario, it’s assumed that the vehicle lateral motion is controlled by the Electric Power Steering. Thus, the path tracking controller will send the target steering wheel angle request to the EPS controller. Then the front wheel is steered and the reaction force between the ground and the tire will steer the vehicle. Thus, the EPS, vehicle lateral dynamic, and tire models are established. In real time application the lateral displacement information is obtained from the perception layer (for example the camera) to perform the feedback lateral motion control. In this research, instead of the camera a state estimation block is built to estimate the lateral displacement of the ego vehicle based on the vehicle dynamic information from the vehicle model. The model is controlled by the path error tracker. The output of the model is the ego vehicle dynamic information, is exploited by the ego vehicle trajectory planner and tracker to perform the feedback control, the system is shown in figure below.

While the subsystem of the model containing EPS module, vehicle dynamic model, tire model and state estimation block are shown as,

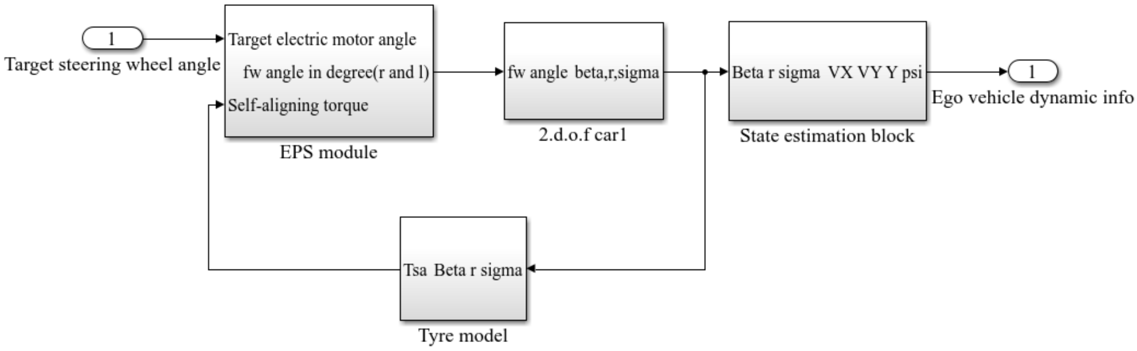


Figure 26 Vehicle system model

Vehicle dynamic model: The vehicle dynamic model adopted in the real-time control loop is the same as the vehicle dynamic model implemented in the offline DP optimization int Chapter 3. The state variables are the vehicle side slip angle β and the yaw rate $\dot{\psi}$. And the input is the steering wheel angle at the front wheel δ.

$$\begin{pmatrix} \dot{\beta} \\ \dot{r} \end{pmatrix} = \begin{bmatrix} -\frac{C1+C2}{mV} & \frac{C2b-C1a}{mV^2}-1 \\ \frac{C2b-C1a}{Jz} & -\frac{C1a^2+C2b^2}{JzV} \end{bmatrix} \begin{pmatrix} \beta \\ r \end{pmatrix} + \begin{bmatrix} \frac{C1}{mV} \\ \frac{C1a}{Jz} \end{bmatrix} (\delta) \tag{87}$$

Tire model: In highway overtaking scenario it’s necessary to take into consideration also the tire model. Since we need to estimate the self-aligning torque induced by the side-slip angle in order to solve the dynamic equation of the steering system.

Considering the side slip angle of a vehicle travelling on a highway is small, it’s reasonable to assume that the tire works in the linear region, thus the self-aligning torque is proportional to the tire side-slip angle, described as[54],

$$T_{sa} = M_{z(\alpha)}\alpha \tag{88}$$

where,

- $\boldsymbol{M_{z(\alpha)}} -$ self-aligning torque coefficient (Nm/rad)
- $\boldsymbol{\alpha} -$ tire side slip angle (rad)

EPS model: The structure of the Electric Power steering is described in the following figure. The steering rod is connected to the electric motor via a reduction gear. Then the steering rod is connected to a rack and pinion transmission mechanism to control the front wheel steering angle. Thus, the transmission ratio between the electric motor angle and steering rod angle, and the transmission ratio between the steering rod angle and the front wheel angle should be taken into consideration. A potentiometer is exploited to monitor the angular position of the steering wheel. The steering wheel angular position information is then used by the EPS controller to perform the close loop control.

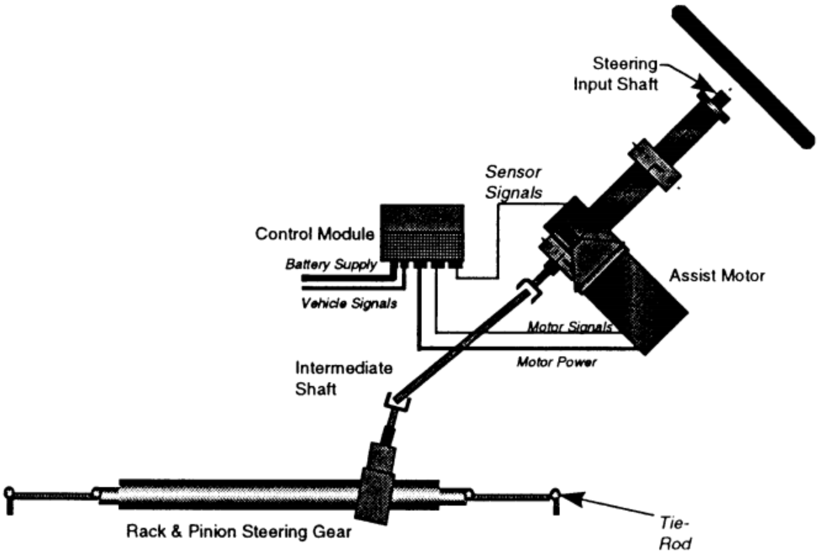


Figure 27 EPS structure[52]

The EPS model contains two parts, the mechanical part and the electrical part. For the mechanical part, assuming the steering mechanism is a rigid body. The deformation, damping, and friction are neglected, the dynamic equation of the steering system is described as,

$$J_{tot}\ddot{\theta}_e = T_e - \frac{T_{sa}}{K_e \cdot K_{sw}} \tag{89}$$

here,

- $\boldsymbol{J_{tot}} -$ total equivalent moment of inertia at the electric motor side (kgm^2)
- $\boldsymbol{\ddot{\theta}_e} -$ electric motor angular acceleration ($\frac{rad}{s^2}$)
- $\boldsymbol{T_e} -$ electric motor torque (Nm)
- $\boldsymbol{T_{sa}} -$ self-aligning torque (Nm)
- $\boldsymbol{K_e} -$ transmission ratio between electric motor and steering rod
- $\boldsymbol{K_{sw}} -$ transmission ratio between steering rod and front wheel steering angle

The steering wheel angle θ_e can be derived from the steering system dynamic equation given the previous states, self-aligning torque $\boldsymbol{T_{sa}}$, and electric motor torque $\boldsymbol{T_e}$. The self-aligning torque can be evaluated using the tire model, while the electric motor torque can be obtained from the electric motor model.

Electric motor model: The equivalent circuit of DC brush motor can be described as,

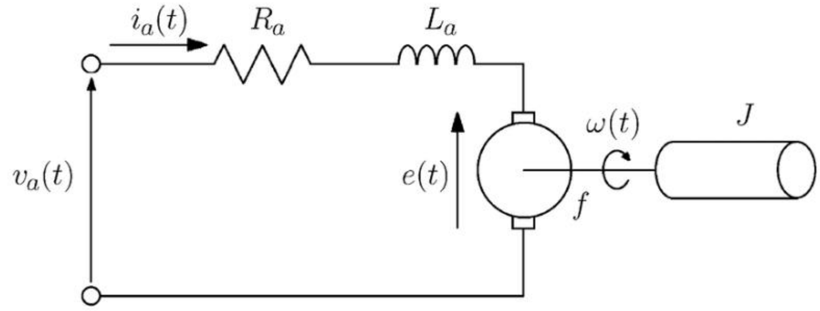


Figure 28 Motor equivalent circuit [53]

which consists of a resistance, inductance and electric motor. The system can be described using the following electric circuit equation,

$$L_a\frac{dI_a}{dt} + R_aI_a + v_e = v_a \tag{90}$$

$$v_e = K_U\frac{d\theta_e}{dt} \tag{91}$$

$$T_e = K_TI_a \tag{92}$$

where,

- $\boldsymbol{L_a} -$ electric motor inductance (*H*)
- $\boldsymbol{R_a} -$ electric motor resistance (*Ω*)
- $\boldsymbol{v_e} -$ electric motor voltage (*V*)
- $\boldsymbol{K_U} -$ electric motor voltage coefficient ($\frac{V}{rad/s}$)
- $\boldsymbol{K_T} -$ electric motor torque coefficient ($\frac{Nm}{A}$)

State estimation block: In order to estimate the longitudinal and lateral displacement of the vehicle, it's necessary to transform the vehicle speed from the local reference frame into the global reference frame.

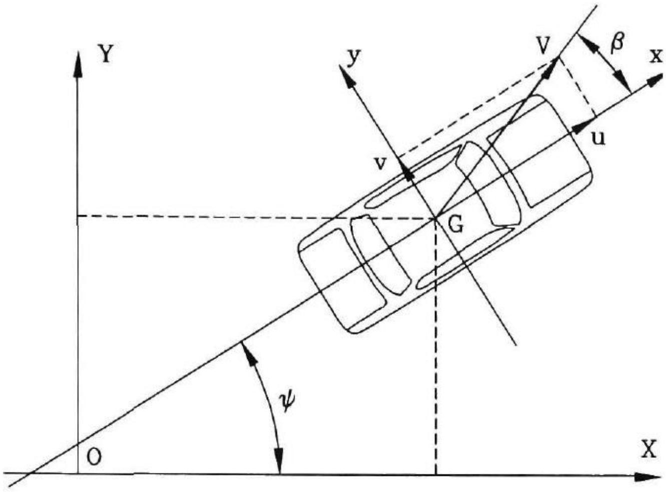


Figure 29 Vehicle local and global reference frame[54]

The side slip angle of the vehicle β is a small angle, thus it's reasonable to introduce the following assumptions,

$$u = Vcos(\beta) \approx V \tag{93}$$

$$v = Vsin(\beta) \approx \boldsymbol{V\beta} \tag{94}$$

The longitudinal and lateral speed int the global reference frame $\boldsymbol{V_X}$ and $\boldsymbol{V_Y}$ are derived as (95), (96). Then the longitudinal and lateral displacement are obtained integrating the speed as equations (97), (98). The yaw angle in the global reference frame $\boldsymbol{\psi}$ is estimated integrating yaw rate $\boldsymbol{\dot{\psi}}$ as equation (99).

$$V_X = Vcos(\psi) - \ V\beta sin(\psi) \tag{95}$$

$$V_Y = Vsin(\psi) + V\beta cos(\psi) \tag{96}$$

$$X_{ego} = \int_0^t V_X(t)dt \tag{97}$$

$$Y_{ego} = \int_0^t V_Y(t)dt \tag{98}$$

$$\psi(t) = \int_0^t r(t)dt \tag{99}$$

The state estimation block is realized in Simulink as,

Then the estimated states of the ego vehicle will be exploited by the rule-based trajectory planner and path error controller to perform the feedback control.

Finally, it’s possible to run the model-in-loop test of the real-time rule-based planner. Given initial condition with different inter vehicular distance and different vehicle speeds, the real time trajectory planner could generate the optimum trajectory in real-time. The results are shown in Chapter 6.

5.3 Rule-based controller implementation for ACC scenario

Similar as the trajectory planner in the overtaking scenario, the results obtained solving off-line DP for the ACC scenario is recorded and processed. Given different initial speed difference and inter vehicular distances. The target acceleration solved using off-line DP are plotted as,

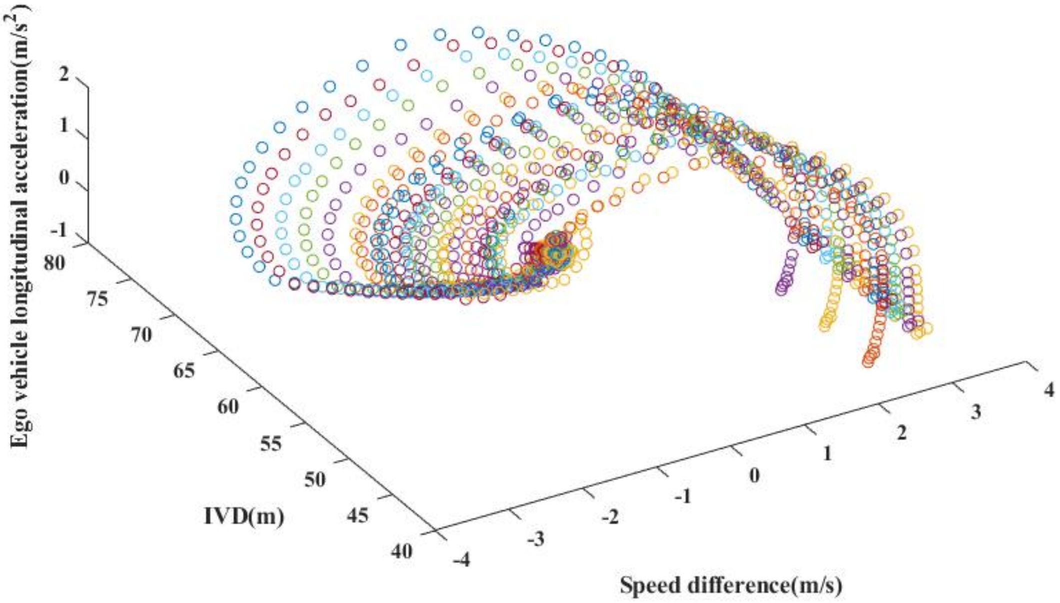


Figure 30 DP off-line optimization results for ACC scenario (oblique view)

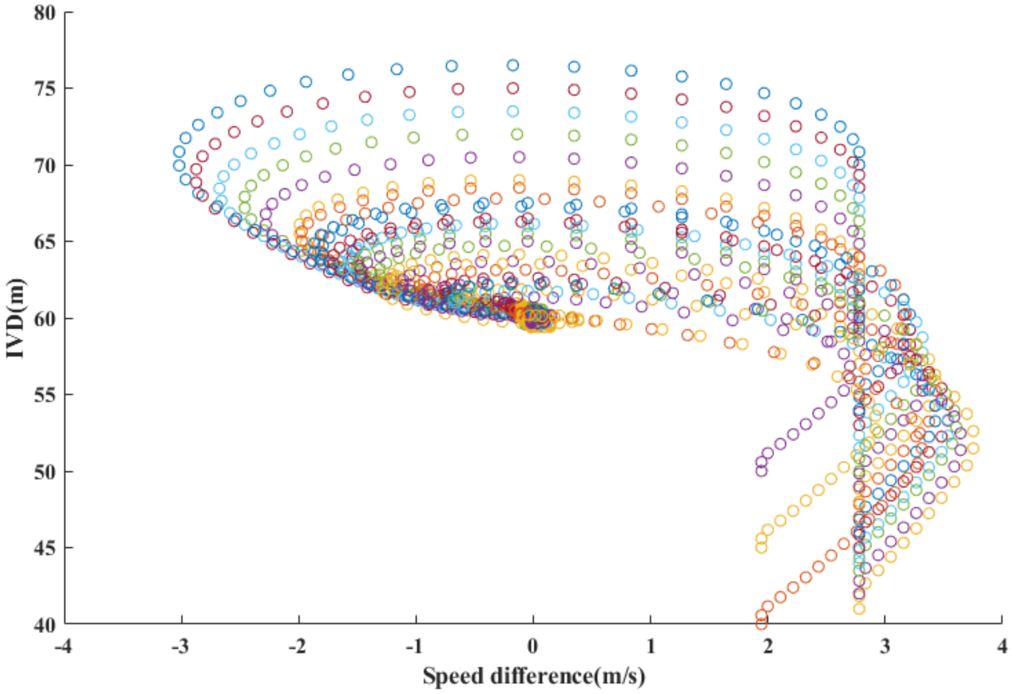


Figure 31 DP off-line optimization results for ACC scenario (top view)

It obvious that the acceleration trajectory with different initial conditions form a regular surface, thus a control rule can be extracted from the surface for real-time implementation. However, in order to tabulate a 2-D look-up table, the data on the x and y axle need to be gridded. An interpolation algorithm is implemented to traverse each data lattice of the x and y axis. Next all the operating points of the DP optimization are traversed and the closest point to the grid lattice is selected, then the target acceleration value is assigned to the grid lattice. Once all the grid latticed are traversed, the 2-D look up table is established. It's reported as,

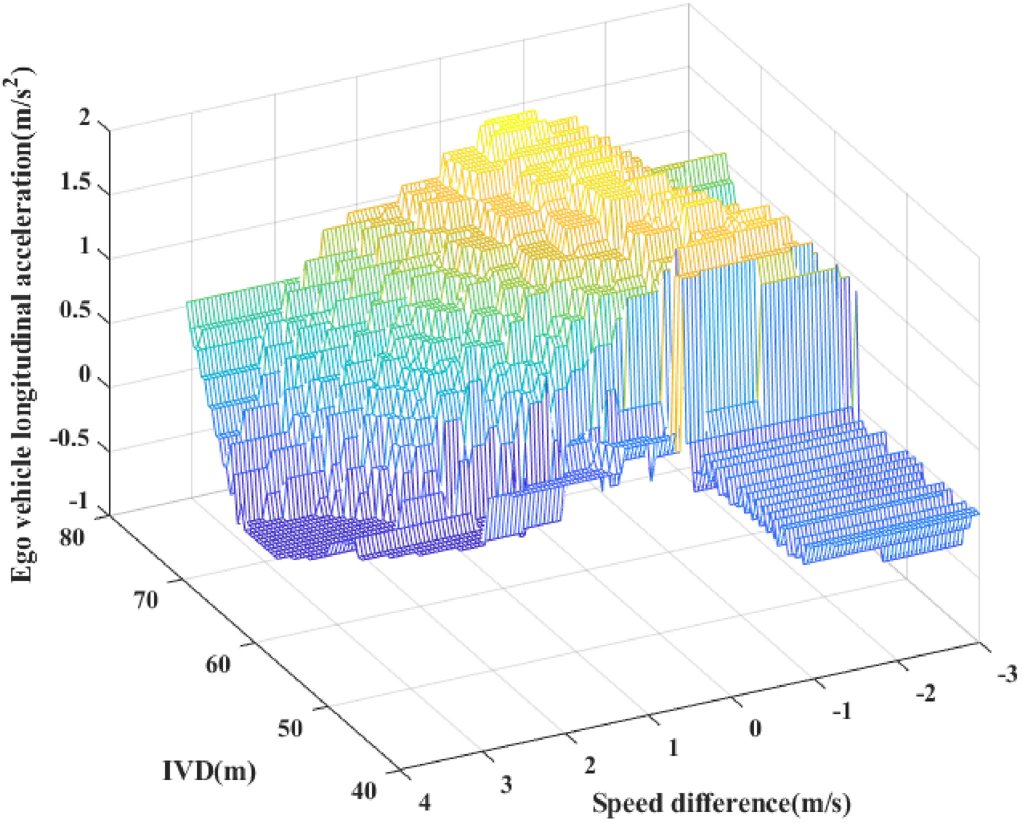


Figure 32 2-D look-up table for target acceleration

Increasing the number of DP offline datasets and discretization of the grid data (discretization of the axle of the look-up table), the results obtained from the real-time controller will approach the result obtained from the offline DP. Since the DP optimization is solved off-line, it's possible to increase the performance of the real-time controller by increasing the number of off-line DP tests. It will not cause any increase of the online computational complexity. While increasing the discretization of the axle-of the look-up table will increase the searching effort for the on-line controller and increase the computation time.

Next the rule-based trajectory planner can be implemented in the loop, the same longitudinal powertrain and energy consumption model are established as the ACC scenarios exploiting the PID and MPC controller. Finally, the longitudinal speed trajectory and the fuel consumption are plotted to compare with the performance of PID, MPC, DP off-line optimization results.

Chapter 6 Results and conclusion

6.1. Results of DP offline simulation

The simulation results obtained from the DP offline algorithm are presented in this chapter. The results of the overtaking scenario and the adaptive cruise control scenario are presented separated.

6.1.1 Overtaking scenario considering varied vehicle speed

As mentioned in the chapter 3, during the overtaking scenario considering the varied vehicle speed, the longitudinal displacement X and speed V_x, lateral displacement Y and side slip angle β are considered state variables. While the longitudinal acceleration a_x and side slip angle variation rate $\dot{\beta}$ are considered as control inputs, the state updating equation is shown as,

$$V_{x,k+1} = V_{x,k} + a_{x,k} \cdot \Delta T$$

$$X_{k+1} = X_k + V_{x,k} \cdot \Delta T$$

$$\beta_{k+1} = \beta_k + \dot{\beta}_k \cdot \Delta T$$

$$Y_{k+1} = Y_k + \beta_k \cdot V_{x,k} \cdot \Delta T$$

The cost term is formulated as follows to represent the overtaking efficiency, longitudinal comfort and lateral comfort,

$$C{=}\alpha_1 \cdot |\Delta Y| + \alpha_2 \cdot |a_{x,k}| + \alpha_3 \cdot |\dot{\beta}_k|$$

Considering a driving scenario with a lead vehicle in front of the ego vehicle in the same road lane. The lead vehicle is travelling at a constant speed and the ego vehicle is travelling at a certain initial speed. The scenario is shown in the following figure,

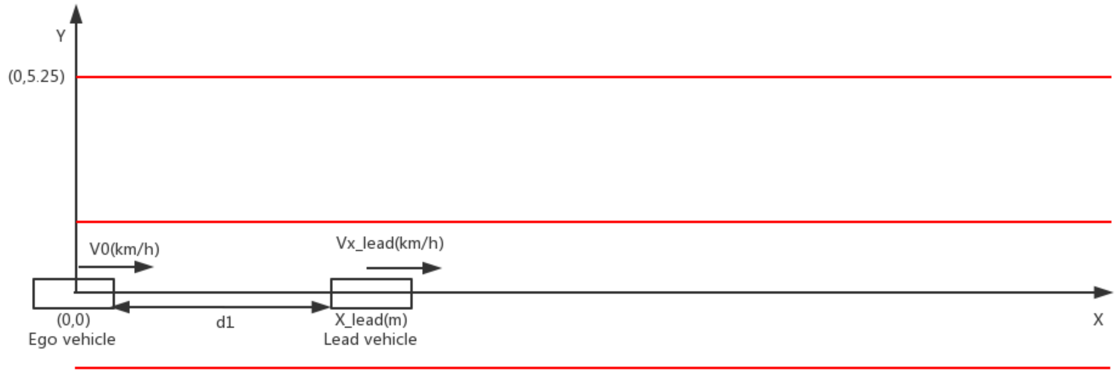


Figure 33 Overtaking scenario

The data of the lateral dynamic model of the vehicle and the electric power steering model are presented in the following table,

Vehicle dynamic data	
Front wheel base $a[m]$	**1.04**
Wheel base $l[m]$	**2.6**
Sprung mass $m_s[kg]$	**1111**
Unsprung mass $m_{us}[kg]$	**120**
Moment of inertia $J_z[kg \cdot m^2]$	**2031.4**
Front tire cornering stiffness $C_1[\frac{N}{rad}]$	**98194**
Rear tire cornering stiffness $C_2[\frac{N}{rad}]$	**69318**
Self-aligning torque coefficient Nm/rad	**1606.6**

Table 2 Data of vehicle dynamic model

Electric power steering data	
Equivalent resistance $R_m(\Omega)$	**0.167**
Equivalent inductance $L_m(H)$	**0.001**

Voltage coefficient $K_v(V/(\frac{rad}{s}))$	**0.02**
Torque coefficient $K_t(Nm/A)$	**0.04**
Steering wheel front wheel transmission ratio G_{sw}	**16.5**
Electric motor steering wheel transmission ratio G_m	**15**
Electric motor moment of inertia $J_m[kg \cdot m^2]$	**0.0024**
Steering wheel moment of inertia $J_m[kg \cdot m^2]$	**0.1**
Steering moduel equivalent mass $M_r[kg]$	**32**

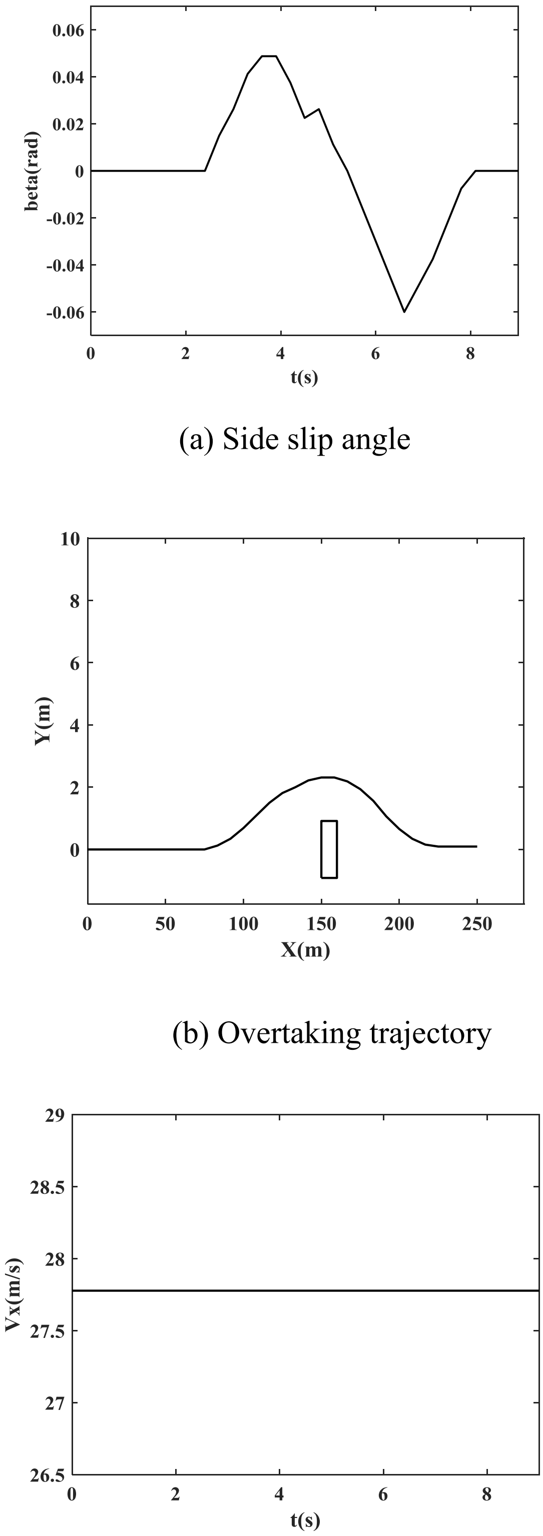
Table 3 Data of electric power steering model

In order to study the influence of the initial condition on the trajectory planning, two sets of tests are designed. For one set the speed difference of the ego vehicle and lead vehicle is fixed while the initial inter-vehicular distance is changed. While for the other set the initial inter vehicular distance is fixed and the speed difference is varied. The tests are designed according to the following table,

Test	**1**	**2**	**3**	**4**	**5**
$V_{diff}(km/h)$	100	100	100	90	70
$IVD(m)$	150	110	50	50	50

Table 4 Initial variables of overtaking scenario test

The side slip angle, longitudinal speed profile, and the overtaking trajectory are plotted as follows,

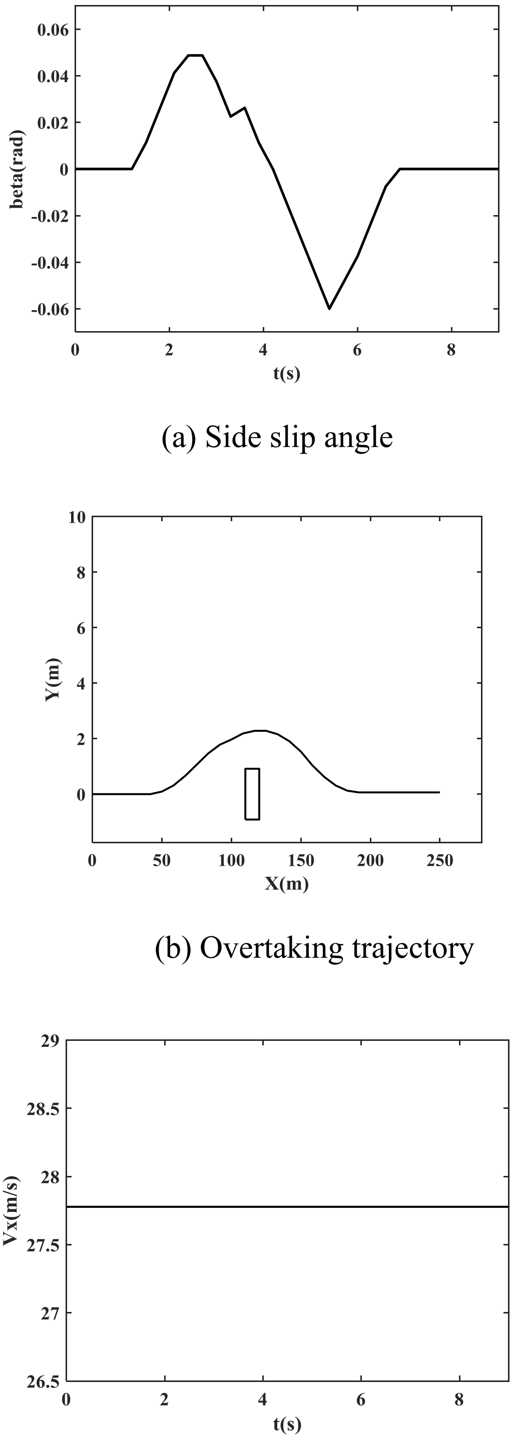


(a) Side slip angle

(b) Overtaking trajectory

(c) Longitudinal speed

Figure 34 Test 1 $V_{diff} = 100$km/h $IVD = 150m$

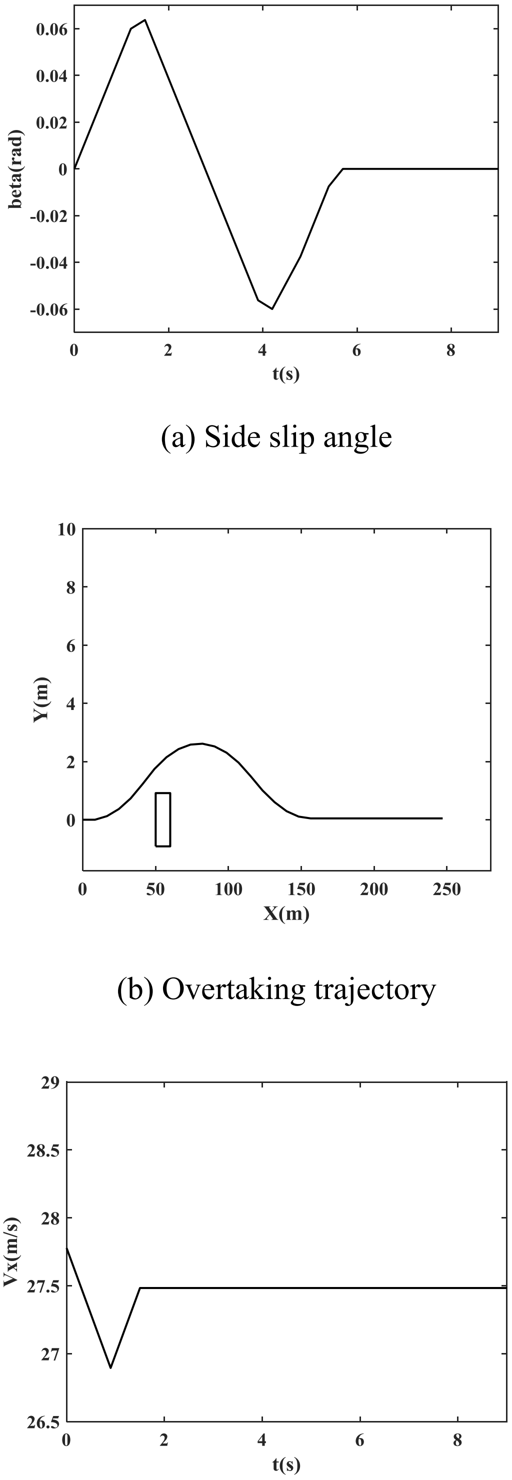


(a) Side slip angle

(b) Overtaking trajectory

(c) Longitudinal speed

Figure 35 Test 2 $V_{diff} = 100$km/h $IVD = 110m$

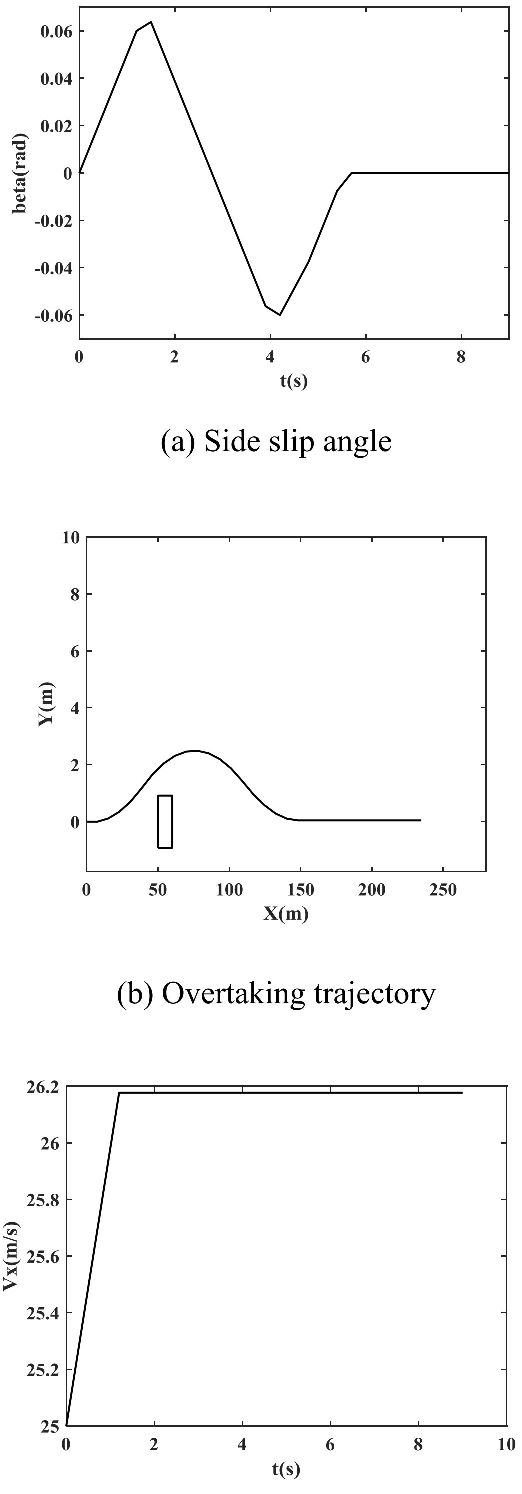


(a) Side slip angle

(b) Overtaking trajectory

(c) Longitudinal speed

Figure 36 Test 3 $V_{diff} = 100$km/h $IVD = 50m$

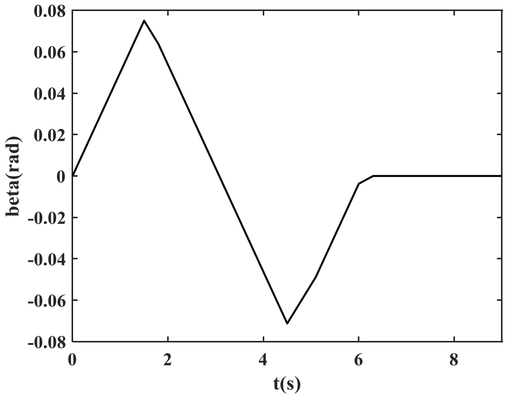


(a) Side slip angle

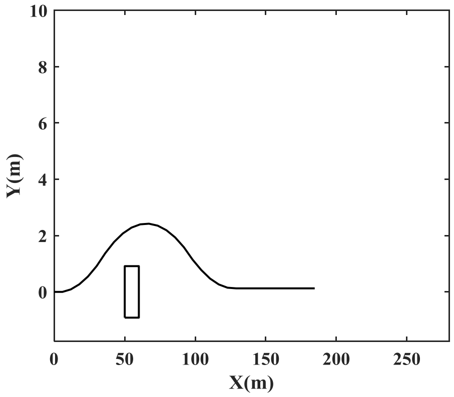
(b) Overtaking trajectory

(c) Longitudinal speed

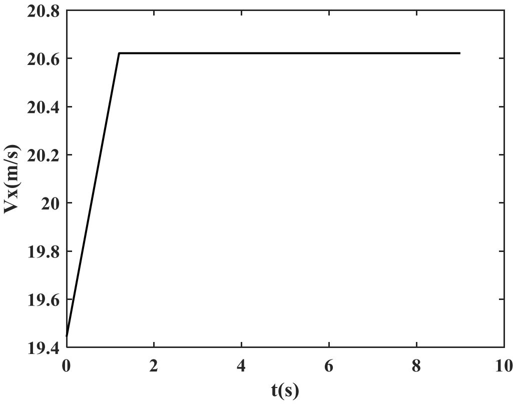
Figure 37 Test 4 $V_{diff} = 90$km/h $IVD = 50m$



(a) Side slip angle



(b) Overtaking trajectory



(c) Longitudinal speed

Figure 38 Test 5 $V_{diff} = 70$km/h $IVD = 50m$

Analyzing the results obtained from the test above, it's easy to conclude that when the initial inter vehicular distance is long enough, the ego vehicle tends to travel straight, and starts to change the lane only at a certain distance behind the ego vehicle.

During the lane changing maneuver it also tends to maintain its speed. Since the initial inter vehicular distance is large and the ego vehicle has enough space to perform the lane change maneuver, the DP algorithm finds out it's unnecessary to vary the speed which could degrade the occupant's comfort.

While for the case that the initial inter-vehicular distance is short, the ego vehicle tends to accelerate to increase the lateral speed in order to shorten the lane change maneuver time and avoid the collision.

Then simulations with different longitudinal speed difference and inter vehicular distance are run, the optimization results are recorded which could be exploited to build the rule-based real time controller. It will be introduced in the next chapter.

6.1.2 Adaptive Cruise Control scenario considering fuel economy

The state space equation and cost function of the ACC scenario are established as introduced in the Chapter 3. The driving scenario is reported as follows,

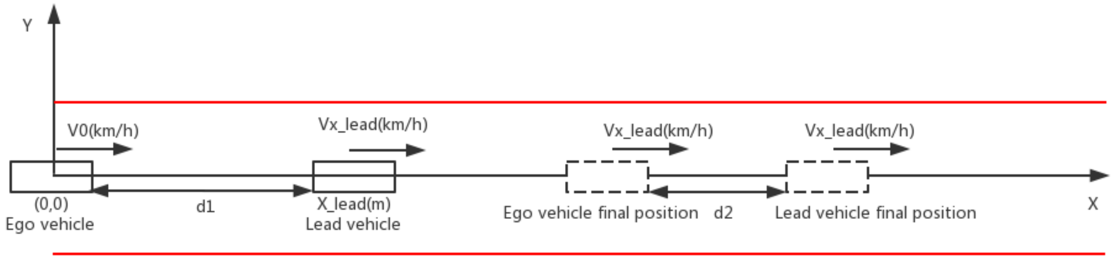
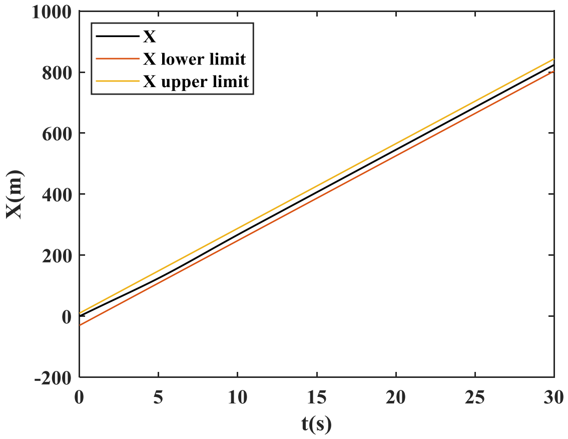
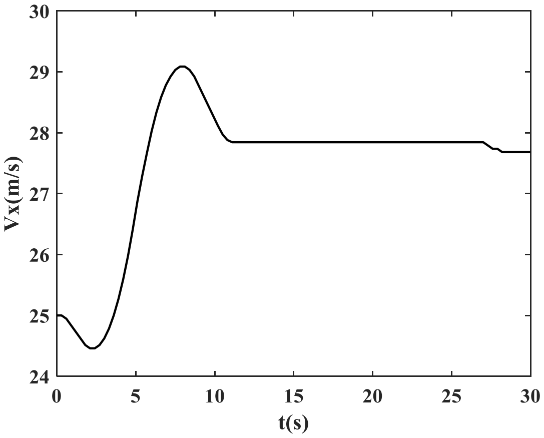


Figure 39 Adaptive Cruise Control driving scenario

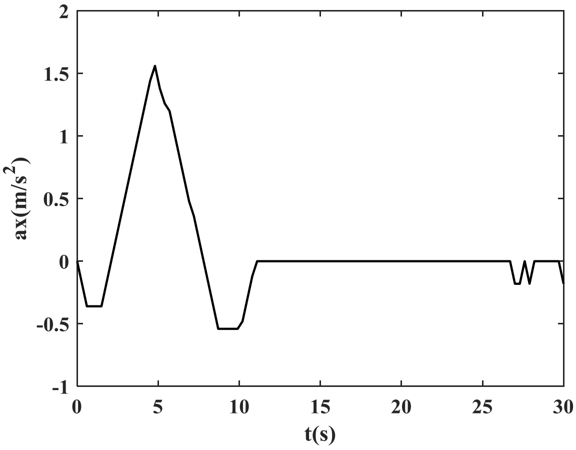
Given the different initial ego vehicle speeds V_0, lead vehicle speeds $V_{x,lead}$, and initial inter-vehicular distance d_1, both of the vehicles will arrive at a known final states with ego vehicle speed equals to the lead vehicle speed $V_{x,lead}$ and a certain inter vehicular distance d_2. Tests with different initial speed are shown as follows,



(a)Longitudinal displacement

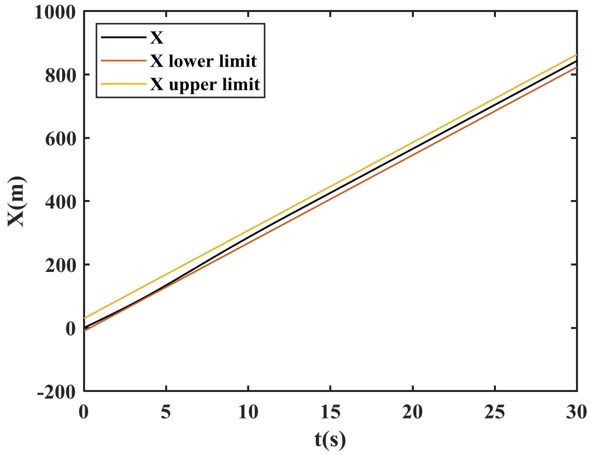


(b) Longitudinal speed

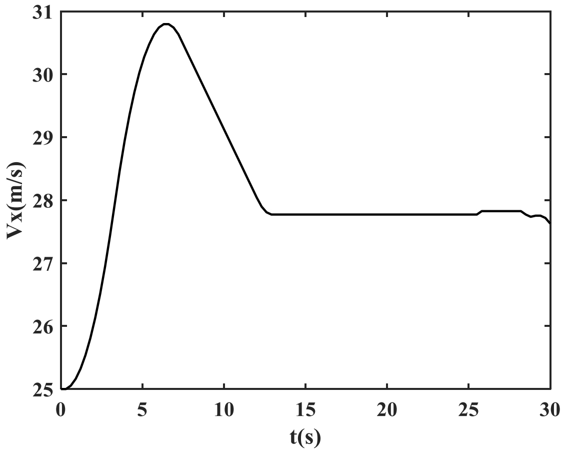


(c) Longitudinal acceleration

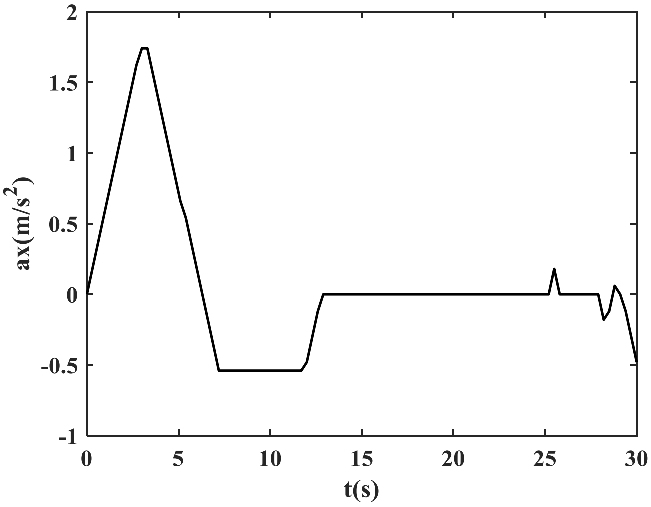
Figure 40 Test 1 $V_{diff} = 10$km/h $IVD = 50m$



(a) Longitudinal displacement

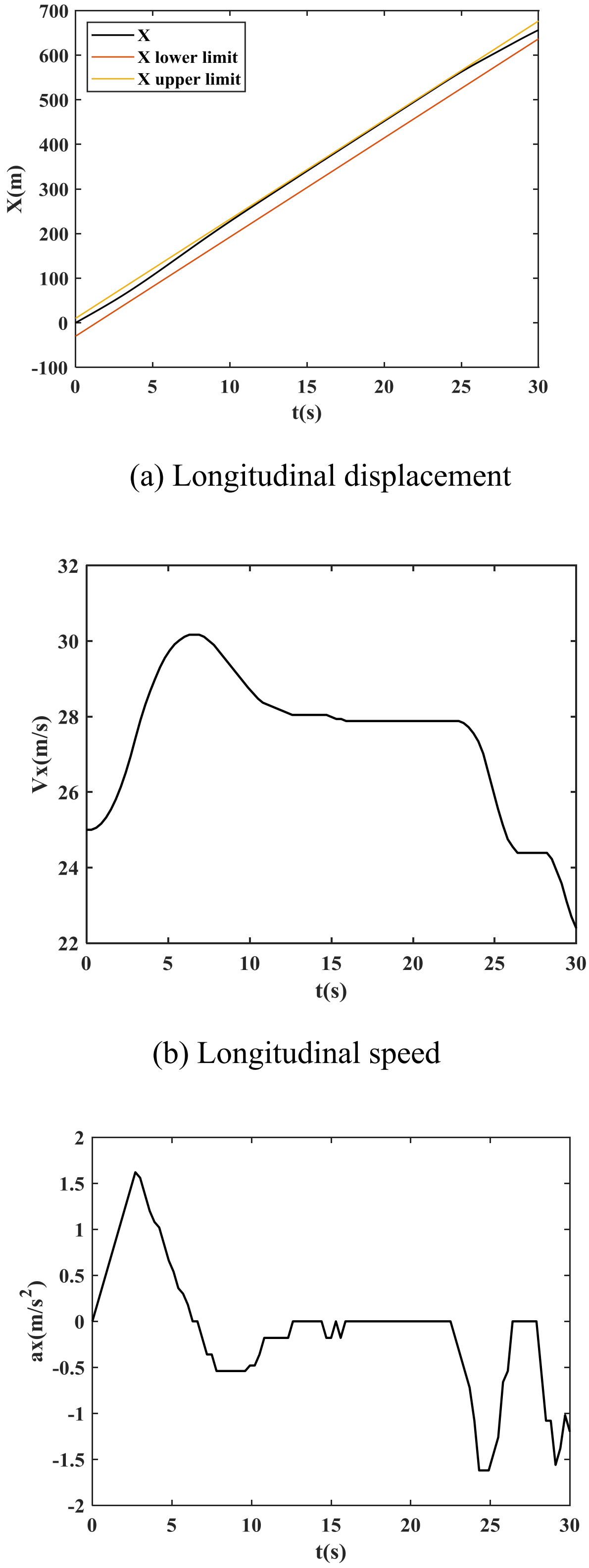


(b) Longitudinal speed



(c) Longitudinal acceleration

Figure 41 Test 2 $V_{diff} = 10$km/h $IVD = 70m$

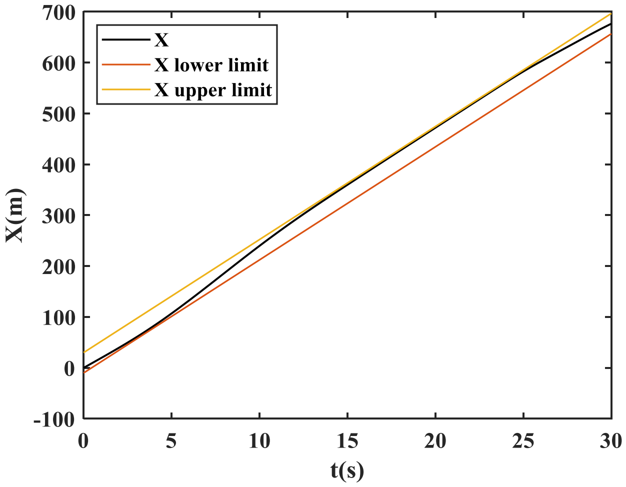


(a) Longitudinal displacement

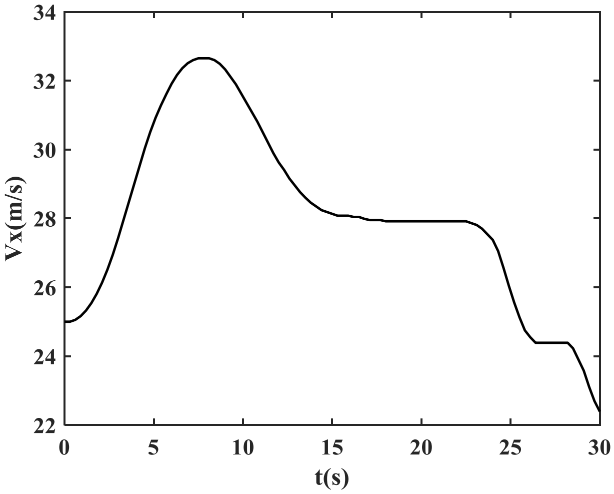
(b) Longitudinal speed

(c) Longitudinal acceleration

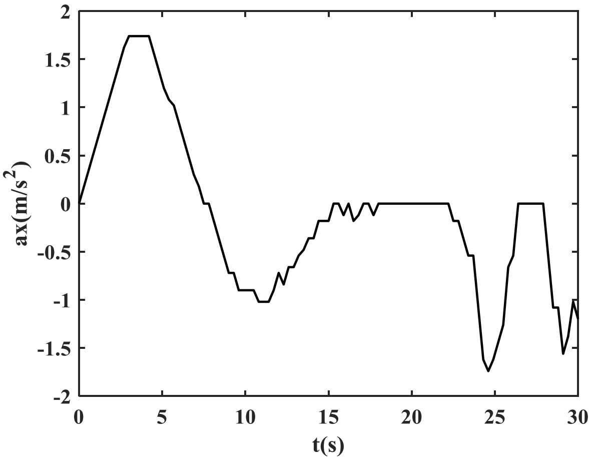
Figure 42 Test 3 $V_{diff} = -10$km/h $IVD = 50m$



(a) Longitudinal displacement



(b) Longitudinal speed



(c) Longitudinal acceleration

Figure 43 Test 4 $V_{diff} = -10$km/h $IVD = 70m$

6.2 Performance comparison of DP and common planning algorithm

Theoretically speaking, the DP method can find the global optimum control sequence. Thus, in order to verify the optimality, several other types of mainstream algorithms are established to verify the performance of the results obtained from the DP controller.

6.2.1 DP result comparison with polynomial for overtaking scenario

The polynomial trajectory planning algorithm is adopted to compare the performance of the DP off-line algorithm as introduced in chapter 3. First the DP trajectory planning algorithm run offline and the optimum overtaking trajectory is determined. Then the initial, overtaking, and final position, and the maneuver time of the two phases of the DP algorithm will be the input of the polynomial solver. By doing this the maneuver efficiency of both algorithms is kept as the same. Finally, the square of the accumulation jerk derived from the results of both algorithms are compared.

The path trajectory is reported as follow. The initial position, overtaking position, and final position the both trajectories have the same lateral displacement.

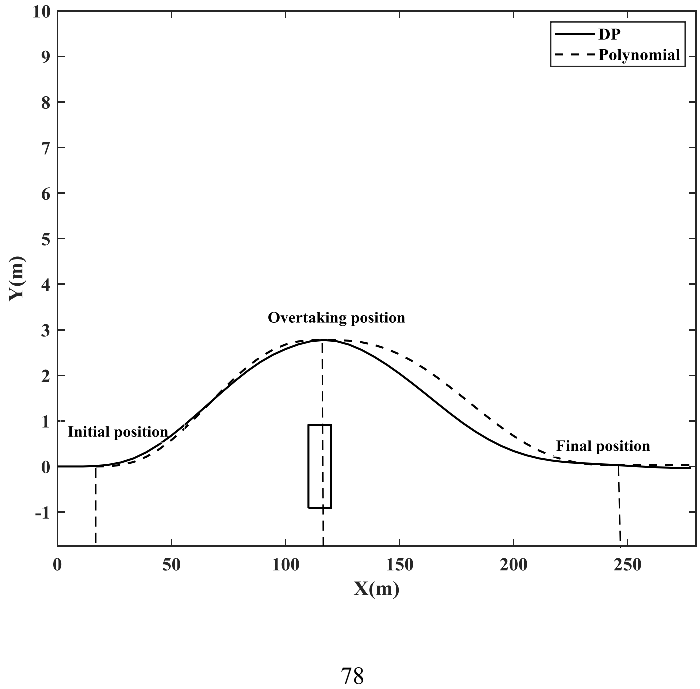


Figure 44 Path trajectory comparison

The history of the side slip angle is reported as,

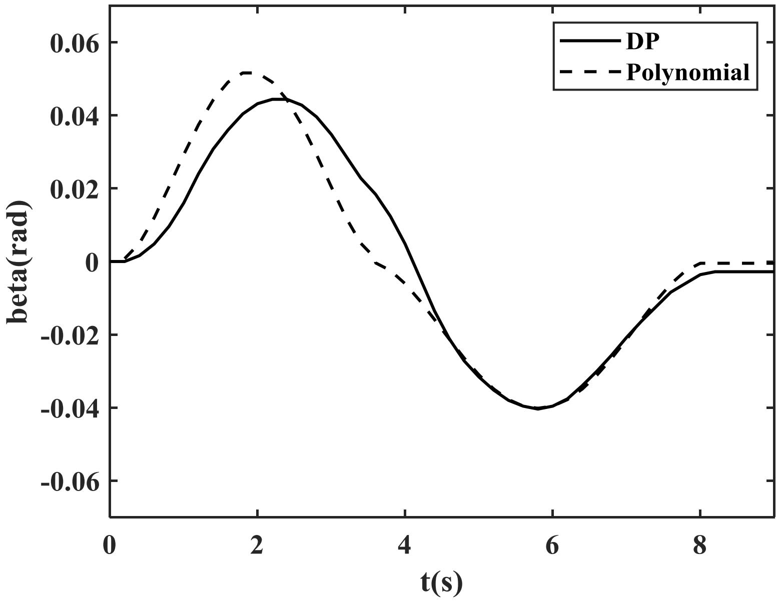


Figure 45 Side slip angle comparison

From figure above we learnt that at the overtaking position the side slip angle β doesn't equal to 0 and the peak is also smaller than the trajectory obtained from the polynomial solver. While for the second phase the trajectories obtained from both algorithms almost coincident.

	DP	Polynomial
$\int \lvert y(\ddot{t})\rvert^2 dt$	17.6	20.1
$\int \lvert y(\dddot{t})\rvert^2 dt$	32.6	35.3

Table 5 Jerk and acceleration accumulation comparison

The results show that the DP method could achieve a smaller jerk accumulation, which means it has a better performance in terms of comfort. The reason is that it could overcome the drawback of the polynomial algorithm. The DP algorithm can find an optimum overtaking position as well as the speed and acceleration. While for the Polynomial solver this information should be the input and it's necessary to find a way to define the optimum value. Thus, DP might also be a reasonable candidate to solve this problem.

6.2.2 DP result comparison with PID and MPC for ACC scenario

In order to compare the performance of the speed trajectory planning algorithm DP, PID, and MPC. An adaptive cruise control scenario with same initial and final states are specified for each condition. Shown as the following table,

	IVD(m)	V_{ego}(km/h)	V_{lead}(km/h)
Initial condition	70	90	100
Final condition	60	100	100

Table 6 ACC scenario initial and final condition

Then the results obtained from these three different controllers are compared. The speed profile and the total fuel consumption are plotted as,

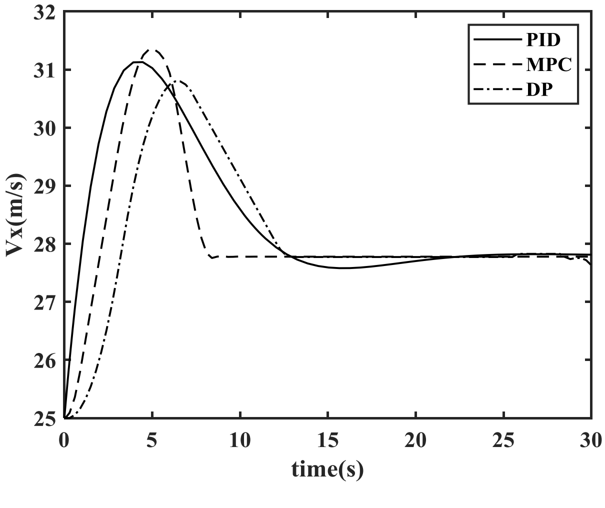


Figure 46 Speed profile of DP and PID controller

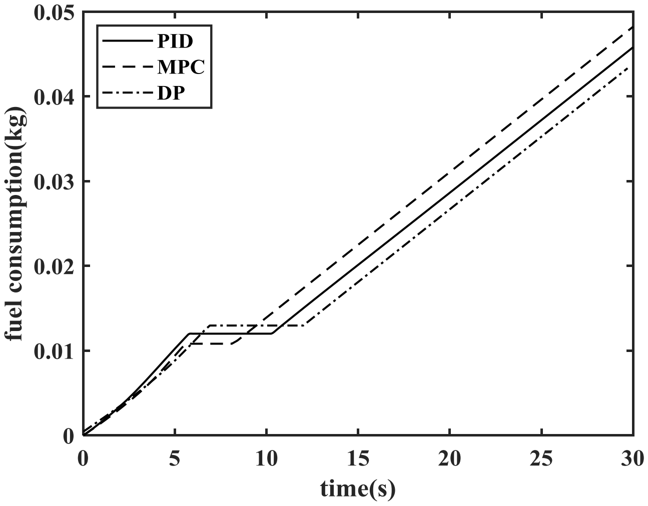


Figure 47 Fuel consumption plot of DP and PID controller

Figure 47 shows that given the same initial and final condition, the MPC controller has the highest maneuver efficiency. However, the speed variation is also the highest. While the DP controller can achieve a smaller speed variation with the same maneuver time compared with the PID controller.

Even if the MPC controller has the highest maneuver efficiency, figure 48 shows that it also has the highest fuel consumption. While the DP controller can achieve the lowest fuel consumption.

In general, the dynamic programming method can possibly achieve a better performance in terms of fuel economy and occupant’s comfort. It can be exploited to find the optimum control sequence which can be used as a benchmark of the other controllers.

6.3 Rule based real time planner

6.3.1 Overtaking scenario

The control sequences obtained from the dynamic programming offline optimization under different driving scenario are obtained in chapter 4. Then all the data are gathered together and a interpolation algorithm is implemented to tabulate a 2-D look up table based on the DP optimization results, in which the target lateral displacement can be found given the current inter-vehicular distance IVD and the vehicle speed difference V_{diff}. Then a predictive PID trajectory tracking controller is developed to control a vehicle dynamic model to follow the trajectory.

The trajectory planned by the DP off-line optimization, rule-based real-time planner, and actual trajectory of the vehicle model are plot on the same figure. The results with different initial conditions are shown as follows,

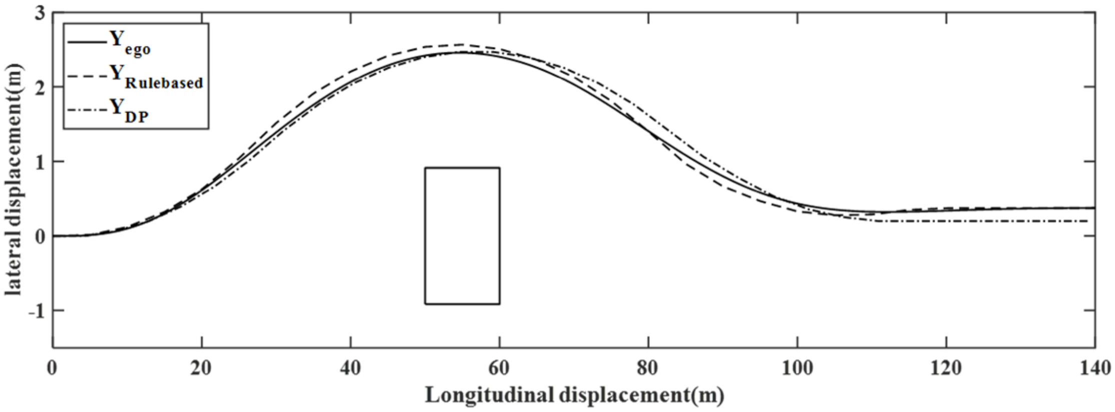


Figure 48 IVD=50m V_{diff}=50km/h

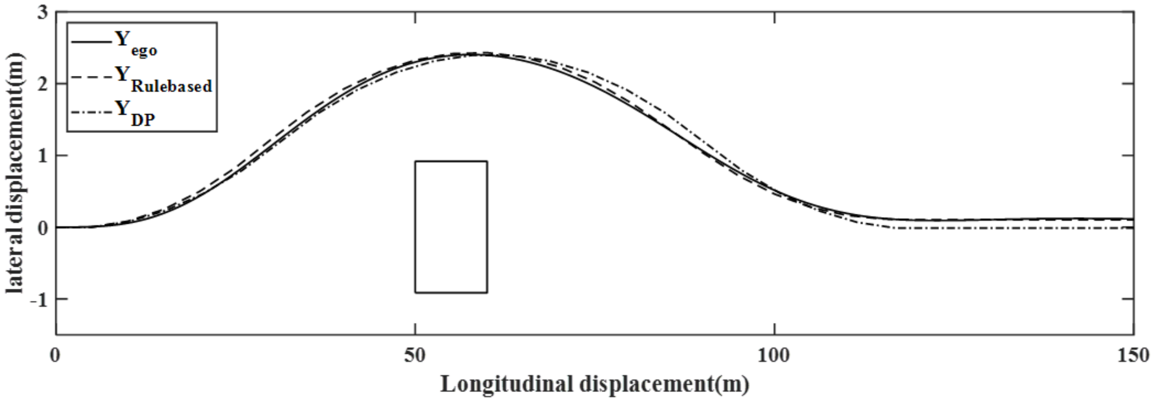


Figure 49 IVD=50m V_{diff}=60km/h

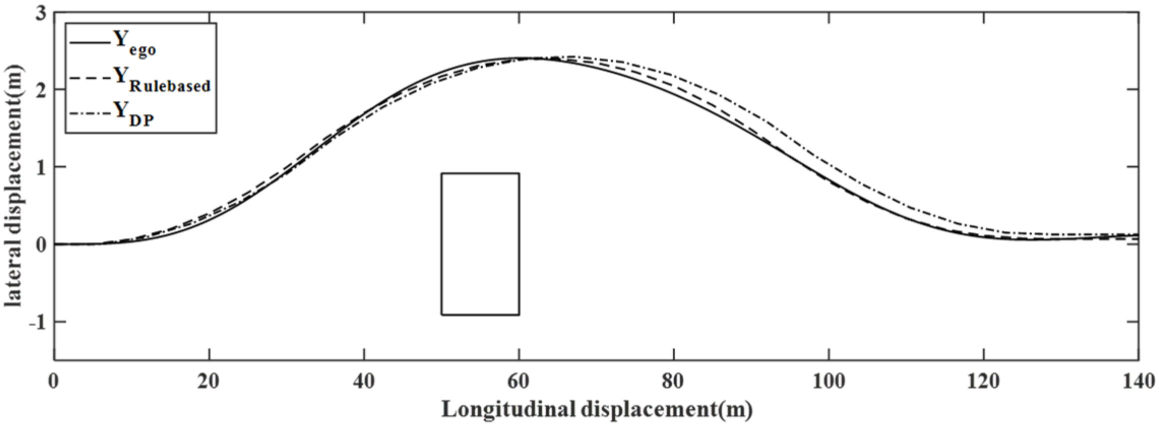


Figure 50 IVD=50m V_{diff}=70km/h

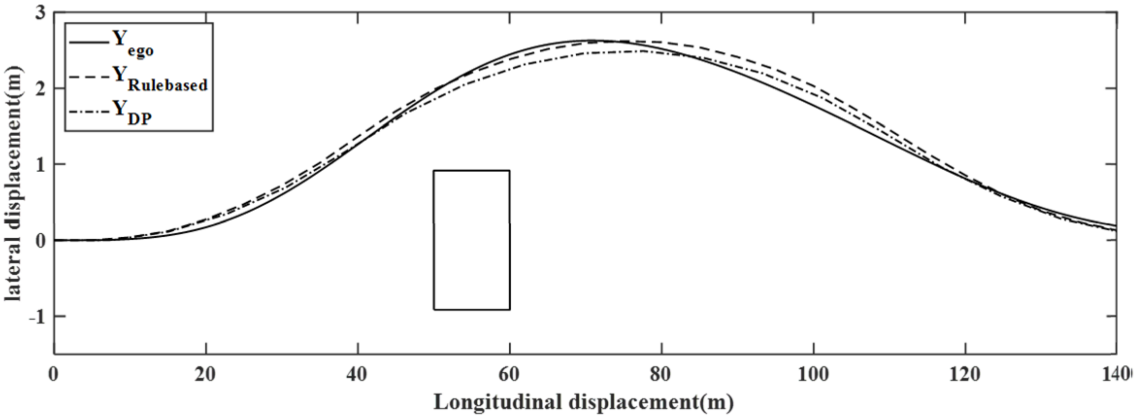


Figure 51 IVD=50m V_{diff}=90km/h

The plot shows that the trajectory generated by off-line dynamic programming, rule-based controller and ego vehicle model are close to each other but are not exactly coincided on the same curve. It's intuitive that the poor performance of the PID path tracking controller results the deviation between the ego vehicle trajectory and the trajectory generated by the real-time rule-based planner. While for the deviation between the trajectory generated by DP optimization and rule-based planner. There are two main reasons. First, during the interpolation process some deviation occurs, which makes the 2-D look up table not the same with the results obtained from DP optimization. Second, the rule-based trajectory planner will re-plan the trajectory at each time instant. Even if the initial state is the same with DP optimization, any difference in the state will cause the different planned trajectory, which will further result the difference in the state. Finally, the trajectory obtained from the DP off-line optimization will deviate from the trajectory generated by the real-time rule-based controller.

6.3.2 Adaptive cruise control scenario

Similar as the overtaking scenario, the fuel consumption and the longitudinal speed trajectory obtained from off-line DP, rule-based, MPC, and PID are plotted on the same figure to compare the performance, the results are reported as,

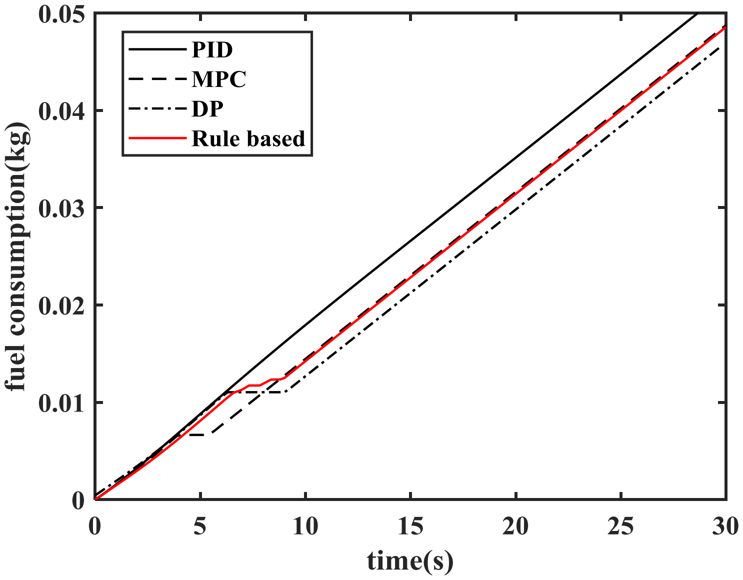


Figure 52 Fuel consumption of different controllers in ACC scenario

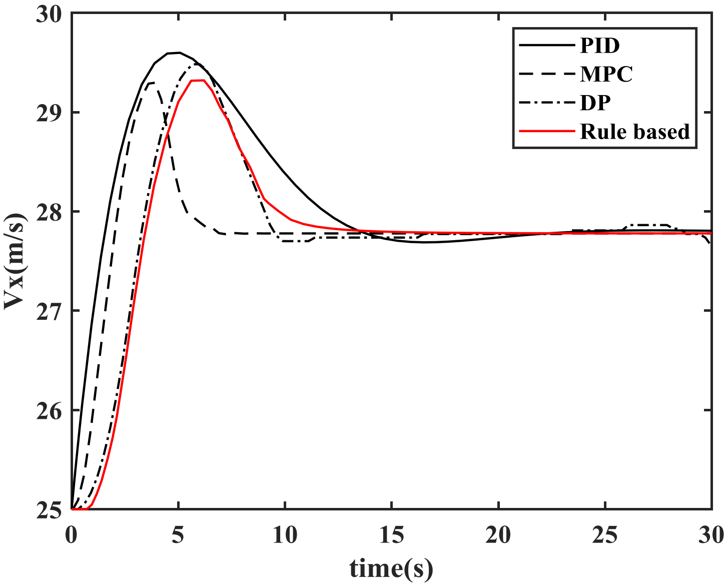


Figure 53 Longitudinal speed of different controllers in ACC scenario

The fuel consumption, peak acceleration value, and maneuver time are reported in as,

	m_f(kg)	a_{peak} (m/s)	t(s)
PID	0.05176	2.3049	21.07
MPC	0.04876	1.7658	7.2
DP	0.04698	1.5600	10.2
Rule based	0.04856	1.4969	12.07

Table 7 Performance comparison of different approach

The parameters of the PID controller is tuned to have the best trade-off between overshoot and rise time. However, it's not possible to have the control results close to that obtained from off-line DP. And it's obvious that the PID controller has the worst performance, since it has the highest fuel consumption, the highest speed variation indicating a poor performance in terms of occupant's comfort, the longest maneuver time representing a poor performance in terms of maneuver efficiency.

The MPC controller shows a quite good performance. It has the shortest maneuver time and lowest longitudinal speed variation. And a better fuel economy compared with the PID controller.

The DP off-line controller achieves a speed profile with the lowest acceleration, which is good from the occupant's comfort point of view. Also, it achieves the best fuel economy.

The rule-based controller is established based on the results obtained solving DP in off-line. It's exploited to express the explicit function between the system's states and the target control input. However, due to the sampling process when establishing the look-up table, some numerical issues occurs making the results of the real-time rule-based controller deviated from the DP off-line controller, and the performance is also reduced. It's possible to reduce the deviation by increasing the number of data sets obtained from off-line DP, increasing the grid density of the look-up table, or adopting better interpolation algorithm during data processing.

The DP off-line controller achieves a speed profile with the lowest acceleration, which is good from the occupant's comfort point of view. Also, it achieves the best fuel economy.

6.4 Conclusion

In this paper, the possibility of implementing an off-line optimization algorithm based on Dynamic Programming in the trajectory planning is studied. Then the overtaking and adaptive cruise control scenarios are studied. The vehicle lateral model and a longitudinal powertrain model of a hybrid electric vehicle are established for each case. Then the DP algorithm is implemented to plan the path trajectory and speed trajectory for each case considering the occupant's comfort, maneuver efficiency, collision avoidance, and the fuel economy off-line. Next, the results obtained from the off-line DP are processed using an interpolation method and the real time controller are established for each scenario. Then it is embedded in the loop. Tests with different initial states are run and the effectiveness of the real-time controller are verified. In order to compare the performance, the polynomial algorithm is implemented in the overtaking scenario to compared with the path trajectory generated by DP algorithm and real-time rule-based planner. While for the ACC scenario a PID and a MPC controller are established to compare the performance.

It's proved that it's possible to adopt the off-line DP in trajectory planning for simple scenario like overtaking and adaptive cruise control, and established the rule-based control based on the results. The rule-based controller can achieve a good control performance at the same time overcome the drawback of computational effort of the off-line DP controller.

However, several aspects could be discussed further regarding this approach. First, even the Dynamic Programming is a powerful tool to solve the optimization problem of system with multi-states, multi-constraints, and multi-optimization targets. A deep comprehending of the vehicle dynamic is required to specify the constraints in order to guarantee that the optimization problem is solvable. Thus, a large effort is required in tuning of the DP algorithm, especially with increasing the number of the state variables. Also, the specification of the sampling time and the whole simulation time need to be properly specified. Second, it's proven that the rule-based controller works well in relatively simple scenario with fewer disturbances. However, it's still necessary to verify the effective of the rule-based controller in scenario with fast varying real-time disturbances. Finally, the performance of the rule-based controller depends on the amount of data obtained from the DP-offline simulation and the performance of the interpolation algorithm. A better interpolation algorithm should be adopted to reduce the deviation of the rule-based controller from the results obtained from off-line DP.

References

[1] H. P. Moravec, The Stanford Cart and the CMU Rover, in Proceedings of the IEEE, vol. 71, no. 7, pp. 872-884, July 1983.

[2] URMSON C, ANHALT J, BAGNELL D, et.al. Autonomous driving in urban environments： boss and the urban challenge[J].Journal of Field Robotics，2008，25(8)：425-466.

[3] «Cruise Secures $1.15 Billion of Additional Investment» [Online]. Available: https://investor.gm.com/news-releases/news-release-details/cruise-secures-115-billion-additional-investment.

[4] « Ford Looking further: ford will have a fully autonomous vehicle in operation by 2021» [Online]. Available: https://corporate.ford.com/innovation/autonomous-2021.html.

[5] Taiebat, Morteza; Brown, Austin; Safford, Hannah; Qu, Shen; Xu, Ming. "A Review on Energy, Environmental, and Sustainability Implications of Connected and Automated Vehicles". Environmental Science & Technology. 52(20): 11449–11465, 2018.

[6] Payre, W., Cestac, J. & Delhomme, P., Intention to use a fully automated car; Attitudes and a priori acceptability. Transportation Research Part F: Traffic Psychology and Behaviour, Band 27, Part B, pp. 252-263, 2014..

[7] Luettel, T., Himmelsbach, M. & Wuensche, H.-J., Autonomous Ground Vehicles—Concepts and a Path to the Future. PROCEEDINGS OF THE IEEE , 100 (Special Issue: SI), pp. 1831- 1839, 2012.

[8] Weyer, J., Fink, R. D. & Adelt, F.. Human–machine cooperation in smart cars. An empirical investigation of the loss-of-control thesis. Safety Science, Band 72, pp. 199-208, 2015.

[9] Ross, P. E.. Robot, you can drive my car; Autonomous driving will push humans into the passenger seat. IEEE SPECTRUM , 51(6), pp. 60-90, 2014.

[10] Alessandrini, A., Campagna, A., Delle Site, A. & Filippi, F.. Automated Vehicles and the Rethinking of Mobility and Cities. Transportation Research Procedia, Band 5, pp. 145-160, 2015.

[11] American Association of States Highway and Transportation Officials, Driving Down Lane-Departure Crashes: A National Priority. [Online].Available: https://rosap.ntl.bts.gov/view/dot/6232/dot_6232_DS1.pdf

[12] Eichelberger A H, Mccartt A T. Toyota drivers' experiences with dynamic radar cruisecontrol, pre-collision system, and lane-keeping assist [J]. Automobile Drivers, 56, 2016.

[13] Rizzi, Matteo, Anders Kullgren, and Claes Tingvall. "The injury crash reduction of low-speed Autonomous Emergency Braking (AEB) on passenger cars." Proc. of IRCOBI Conference on Biomechanics of Impacts. 2014.

[14] Beiker S A. Legal aspects of autonomous driving[J]. Santa Clara L. Rev., 52: 1145, 2012.

[15] Anderson J M, Nidhi K, Stanley K D, et al. Autonomous vehicle technology: A guide for policymakers[M]. Rand Corporation, 2014.

[16] Bagloee S A, Tavana M, Asadi M, et al. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies[J]. Journal of modern transportation, 2016, 24(4): 284-303.

[17] Shchetko, Nick, 2014. Laser Eyes Pose Price Hurdle for Driverless Cars, Wall Street Journal. July 21.

[18] « Velodyne Just Cut the Price of Its Most Pupular Lidar Sensor in Half » [Online]. Available: https://www.thedrive.com/tech/17297/velodyne-just-cut-the-price-of-its-most-popular-lidar-sensor-in-half

[19] J.D. Power and Associates, 2012. 2012 U.S. Automotive Emerging Technology Study

[20] Mccluskey, B. (2017). Connected cars–the security challenge [Connected Cars Cyber Security]. Engineering & Technology, 12(2), 54-57.

[21] Amoozadeh, M., Raghuramu, A., Chuah, C. N., Ghosal, D., Zhang, H. M., Rowe, J., & Levitt, K. (2015). Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. IEEE Communications Magazine, 53(6), 126-132.

[22] SAE J3016

[23] K. P. Divakarla, A. Emadi, and S. Razavi, "A cognitive advanced driver assistance systems architecture for autonomous-capable electrified vehicles," IEEE Trans. Transport. Electrific., vol. 5, no. 1, pp. 48–58, Mar. 2019.

[24] Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische mathematik, 1959, 1(1): 269-271.

[25] « Dijkstra's algorithm» [Online]. Available: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

[26] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in Proc. Int. Symp. Artif. Intell., Robot., Autom. Space, 2005, pp. 1–7.

[27] M. Rufli and R. Y. Siegwart, "On the application of the d∗ search algorithm to time-based planning on lattice graphs," in Proc. ECMR, 2009, vol. 9, pp. 105–110.

[28] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 566–580, Aug. 1996.

[29] Girdhar, Yogesh & Bystroff, Christopher & Akella, Srividyaneeharika. (2005). Efficient sampling of protein folding pathways using HMMSTR and probabilistic roadmaps. 222 - 223.

[30] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," Int. J. Robot. Res., vol. 20, no. 5, pp. 378–400, 2001.

[31] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in Proc. 49th IEEE CDC, Dec. 2010, pp. 7681–7687.

[32] « What Is The Difference Between RRT And RRT*» [Online]. Available: https://www.youtube.com/watch?v=JeEk_CWcRFI

[33] A. Piazzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, “Quintic g2-splines for the iterative steering of vision-based autonomous vehicles,” IEEE Trans. Intell. Transp. Syst., vol. 3, no. 1, pp. 27–36, Mar. 2002.

[34] J.-W. Choi, R. Curry, and G. Elkaim, “Path planning based on bézier curve for autonomous ground vehicles,” in Proc. IEEE IAENG Spec. Ed. WCECS, 2008, pp. 158–166.

[35] M. Brezak and I. Petrovic, “Real-time approximation of clothoids with bounded error for path planning applications,” IEEE Trans. Robot., vol. 30, no. 2, pp. 507–515, Apr. 2014.

[36] R. E. Bellman, Dynamic programming. Princeton - N.J.: Princeton University Press, 1957.

[37] P. Varaiya, “Reach set computation using optimal control,” in Proc. KIT Workshop Verification Hybrid Syst., Grenoble, France, 1998, pp. 377–383.

[38] A. Kurzhanski and P. Varaiya, “Dynamic optimization for reachability problems,” J. Opt. Theory Appl., vol. 108, no. 2, pp. 227–251, 2001.

[39] M. Schaefer, “Some aspects of the dynamics of populations important to the management of the commercial marine fisheries,” Bulletin of Mathematical Biology, vol. 53, pp. 253–279, 1991, Reprinted from the Bulletin of the Inter-American Tropical Tuna Commission, 1(2):27–56, 1954.

[40] H. Mosbech, “Optimal control of hybrid vehicle,” in International Symposium on Automotive Technology & Automation, vol. 2. Turin, Italy: Automotive Automation Ltd, 1980, pp. 303–310.

[41] A. Sciarretta and L. Guzzella, “Control of hybrid electric vehicles,” IEEE Control Systems Magazine, vol. 27, no. 2, pp. 60–70, 2007.

[42] P. Elbert, S. Ebbesen and L. Guzzella, "Implementation of Dynamic Programming for n-Dimensional Optimal Control Problems With Final State Constraints," in IEEE Transactions on Control Systems Technology, vol. 21, no. 3, pp. 924-931, May 2013.

[43] M. McNaughton, C. Urmson, J. Dolan, and J.-W. Lee, “Motion planning for autonomous driving with a conformal spatiotemporal lattice,” in Proc. IEEE ICRA, May 2011, pp. 4889–4895.

[44] Hogan N. An organizing principle for a class of voluntary movements[J]. Journal of Neuroscience, 1984, 4(11): 2745-2754.

[45] Huria, Tarun & Ceraolo, Massimo & Gazzarri, Javier & Jackey, Robyn. High fidelity electrical model with thermal dependence for characterization and simulation of high power lithium battery cells. Electric Vehicle Conference (IEVC), 2012 IEEE International. 2012.

[46] Qin, Joe & Badgwell, Thomas. (2003). A Survey of Industrial Model Predictive Control Technology. Control engineering practice. 11. 733-764. 10.1016/S0967-0661(02)00186-7.

[47] XI, Yu-Geng & Li, Dewei & Lin, Shu. (2013). Model Predictive Control — Status and Challenges. Acta Automatica Sinica. 39. 222–236. 10.1016/S1874-1029(13)60024-5.

[48] Alessio, A., & Bemporad, A. (2009). A survey on explicit model predictive control. In Nonlinear model predictive control(pp. 345-369). Springer, Berlin, Heidelberg.

[49] « Understanding Model Predictive Control, Part 5: How To Run MPC Faster» [Online].Available: https://www.youtube.com/watch?v=1MpbPd6Ri-8&t=356s.

[50] Anselma, Pier Giuseppe & Huo, Yi & Roeleveld, Joel & Belingardi, Giovanni & Emadi, Ali. (2018). Real-time rule-based near-optimal control strategy for a single-motor multimode hybrid electric vehicle powertrain.

[51] Badawy A, Zuraski J, Bolourchi F, et al. Modeling and analysis of an electric power steering system[R]. SAE Technical Paper, 1999.

[52] Baharom M B, Hussain K, Day A J. Design of full electric power steering with enhanced performance over that of hydraulic power-assisted steering[J]. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering, 2013, 227(3):390-399.

[53] State–Space Parameter Identification in a Second Control Laboratory - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Equivalent-circuit-of-an-armature-controlled-dc-motor_fig1_3050910

[54] Genta, G. (1997). Motor Vehicle Dynamics: Modeling and Simulation. Series on Advances in Mathematics for Applied Sciences.

Acknowledgement

I would first like to thank Dr. Ali Emadi and Prof. Giovanni Belingardi for offering me such wonderful chance at McMaster University in Canada and the support on my thesis.

I would also like to thank Prof. Carlo Novara for his guidance and suggestions on my thesis which helps me solving my doubts.

Besides my advisors, I would also like to offer my special thanks to Ing. Pier Giuseppe Anselma on all his instructions and willingness to help on every step of my thesis.

Thanks to the EcoCar team, Megan, Mike, Josh for giving suggestions on my thesis topic. Thanks to Sam, Augustino, Matt, Arthur for helping me set up the test bench. A special thanks to Andrew for lending me his bike. And also Joe, who always discussed with me and gave suggestions on my project.

My sincere thanks also go to my roommate Zhituo for debating with me on my thesis and giving me advice, also for offering me a wonderful place to study.

Thanks to my friends, Baoqing, Alessandro, Mattia for sharing with me a fun time in Canada.

Last but not the least, I would like to offer my deepest gratitude to my family, especially my grandma, for always being my solid support and my harbor, for giving me the courage to overcome all the difficulties in the journey of my life.