Department of Electrical Engineering

香港城市大學
City University of Hong Kong

Department of Electrical Engineering

EE6680D/EE6680 Dissertation

Project Title:

A Computation-aware Motion Planning and Control Architecture With Event-triggered MPC for Autonomous Vehicles

Student Name: ZHANG Li
Student ID: 56507997
Supervisor: Prof. NEKOUEI Ehsan
Assessor: Prof. CHEN Jie

Abstract

In this dissertation, an overtaking control architecture is developed for autonomous vehicles using Model Predictive Control (MPC). The proposed architecture is comprised of three components: a motion planning, a MPC and a low-level controller. The motion planner computes a trajectory for safe overtaking. Based on the output of the motion planner, the MPC generates a smooth trajectory which meets the vehicle's constraints. A proportional-integral-derivative (PID) controller ensures that the vehicle tracks the reference trajectory generated by the MPC. In order to reduce the computational cost of motion planning, we also proposed an event-triggered MPC. Two different methods of motion planning methods are proposed: the A* method and the jerk minimization method. The performance of the proposed architecture is investigated on the dry and icy road conditions where a kinematic model of the vehicle is used for motion planning by the MPC. The response of the vehicle is modeled by an accurate dynamic model with a nonlinear tire model.

The results indicate that the smooth reference trajectory, generated by the jerk minimization method, is more suitable for MPC, and the event-trigger MPC can save up to 50% computation cost compared with a periodic MPC. In addition, it is shown the event-trigger MPC can still control autonomous vehicles on icy roads.

Table of Contents

Acknowledgment

First of all, I would like to express my heartfelt thanks to Prof. NEKOUEI Ehsan for his encouragement, meticulous guidance, and patience. I was fortunate to be his student. His research and knowledge attitude not only expanded my ability to think about research but also affected my personality. Then, I would like to thank my assessor, Prof. CHEN Jie, for his suggestions and encouragement.

I owe a great deal of gratitude to my friends at City University of Hong Kong that changed my life into beautiful memories. I also want to thank my previous friends that gave me encouragement when I was confused.

In the end, I would like to give my special thanks and love to my family: my cousins, my aunts, my parents and my grandparents for their endless love and support.

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

The autonomous vehicles are no stranger to us, especially since 2020. Autonomous vehicles have been vigorously studied all over the world. From the 1920 to today, the autonomous vehicle technology has undergone rounds of development and innovation. In 2014, the Society of Automotive Engineers divided autonomous driving into five levels from Level 1 to Level 5, where Level 5 is always human's goal.

This technology brings great convenience to our lives when it can help us transport to goods, distribute express delivery, and travel by autonomous cars. When there are unmanned vehicles on the road, unified dispatching can be done, which greatly reduces traffic jams. Simultaneously, this technology can also fully comply with traffic rules and avoid the obstacle to reduce accidents.

At the same time, this technology is currently facing some problems. The first question is how to avoid obstacles. If the vehicle cannot quickly, stably, and accurately avoid obstacles on the road, severe accidents will occur, ranging from the damage of the passengers to the car's destruction. In practical applications, each vehicle is in a complex environment, and the shape, size, and movement of obstacles are different. Therefore, each unmanned vehicle needs to have a system of path planning and tracking control to meet unmanned driving requirements. Most of the technology is now more proficient, but there are still some problems when encountering particular scenarios such as overtaking. So, this dissertation will focus on solving the overtaking problem.

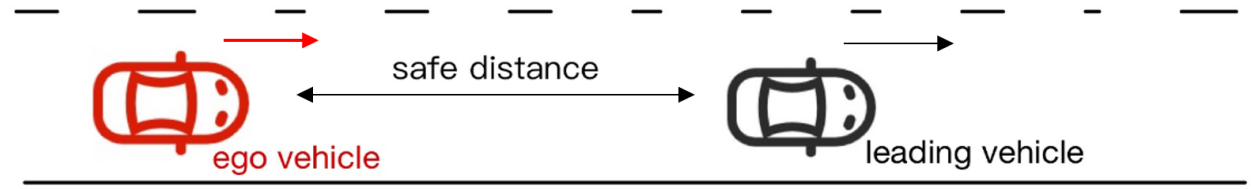

Fig. 1. Overtake problems

When driving a scene similar to Fig. 1, the driver always wants to overtake if the velocity of the leading car is slow. However, overtaking is a relatively dangerous driving action, and many traffic accidents occur during overtaking. Human drivers cannot overtake well because the behaviour of other cars is difficult to predict, the timing of changing lanes is difficult to grasp, acceleration cannot be determined precisely and the route of the overtaking vehicle cannot be well planned. Therefore, overtaking is significant and meaningful for autonomous driving. Autonomous driving can help human drivers solve the above problems and at same time save as much computational power as possible.

After such a long development period, four fundamental path planning algorithms have been developed in the previous literature: graph search-based methods such as A* [1] or Dijkstra, virtual potential methods, meta-heuristic based methods, and mathematical optimization based methods. The first three methods need a global map to achieve better performance. So, they are suitable to be global planners. The Model Predictive Control (MPC) is apart from mathematical optimization-based methods. When the car cannot obtain the global map, MPC can be used as a local planner. The vehicle prediction model is an essential part of MPC. If the model is too complex, it will require lots of time to obtain the optimal path that is not safe. However, if the model is too simple, the MPC cannot predict the vehicle motion. The four methods all can be not only the local planner but the global planner in the plan layer as the sole planner.

MPC is the current mainstream method, and it can be used in an open-loop method, planning an optimal path from an initial point to a target point offline followed by tracking with a feedback controller such as the PID controller. Besides, MPC is convenient when adding constraints. To take care of the passengers' feelings when the car is moving, MPC will limit the trajectory's speed, acceleration, and smoothness. In practical applications, the road boundary also needs to be determined. Therefore, MPC is now the most suitable method for obstacle avoidance in autonomous driving when the system has a reference trajectory. The reference trajectory can be generated by the high-level planner or a straight line from the start point to the goal point.

This dissertation presents a complete control architecture to overtake the leading vehicle for the autonomous vehicle. There are three main parts in this architecture. First, a reference trajectory is generated by A* or minimum-jerk method based on predicting the movement of the leading vehicle. Second, we use MPC to plan an optimal path and give the reference input to the next part. At last, PID controller will control the vehicle to overtake based on the reference input from MPC.

Generally, MPC has a heavy computational burden on cars. The MPC needs to calculate the optimal solution for all the prediction horizons in every run, but the ego vehicle will only use one solution as the input. This is called periodic MPC. To reduce the computation complexly of MPC, we propose a novel MPC named event-triggered MPC that can save computing resources as much as possible when the error is not much changed compared with the periodic MPC. Simulations in different MPC show the effectiveness of the event-triggered MPC.

The rest of the dissertation is organized as follows. Section II is literature survey that discusses the current progress on obstacle avoidance based on MPC. Section III gives an overview of the control architecture, including the trajectory generation, modelling vehicle and MPC formulate both periodic MPC and event-triggered MPC. Section IV presents the simulation showing the benefits of event-triggered MPC and its performance on smooth reference trajectories and icy road conditions. Section V is the conclusion of the dissertation.

CHAPTER 2 LITERATURE SURVEY

The vehicle is a complex system. There are three main ways to model a vehicle: point-mass model, kinematic model, and dynamic model [3]. Different models have different features that can model the vehicle depending on the different discrete-time and computational burdens.

There are two types of obstacles in the real world: static obstacles and moving obstacles. Stationary obstacles are fixed in the earth coordinate system. The position of moving obstacles such as pedestrians and other vehicles on the road is changing and the velocity of the obstacles is not constant.

2.1 Static obstacles

Based on these models, many ways for path planning and motion control of vehicles have been proposed, such as polynomial method and model predictive control with the obstacle information to be the hard constraint or the soft constraint. In [8] and [9], the authors use MPC to plan the motion and control a vehicle based on dynamic model. The author works with the parking problem. The obstacle information, the hard constraints, makes the MPC a non-convex problem and they use Hybrid A* to get the initial gauss to improve the accuracy of MPC. In [4], the author uses a novel way to define the obstacle information. The distance calculates by modified-parallax-based approach and adds it to the MPC's soft constrict. This method makes obstacles location more accurate and increases the calculation speed. In [10], the obstacle information is formulation as the time to collision (TTC), and the TTC is added into the MPC based on the dynamic model as a hard constraint. Hard constraints are more secure than soft constraints, but it is hard to solve the MPC when the safe area is not big.

Motion planning and control can be divided into two problems to consider and where obstacle information as Euclidean distance. So, a two-level or even more level structure is obtained, which can save computing costs. In [2], the first level uses a nonlinear MPC with a pointwise repulsive function based on the distance to add the obstacle information as soft constraint based on the point-mass model to generate a collision-free path and then a low-level controller such as LQR or PID to control the vehicle driving on the path. Another way is computing the distance between the ego vehicle and obstacle and using it as the soft constraint [6], with another MPC to follow the trajectory generated by high-level MPC. In the high-level MPC, the authors use point-mass model to predict the vehicle and use the dynamic model to control the car in low-level MPC. These two methods avoid being the non-convex problem, and they treat obstacles as points instead of objects with bounds. Compare with the one-level structure, two-level structure can save computation time and improve accuracy.

A two-level controller works with the non-convex problem directly [11]. The authors add a preview distance block to reduce the error of the trajectory generated by the nonlinear MPC based on point-mass model and use a PI controller to follow the path. This method can accommodate more information from the obstacles, but it still needs more time to solve the non-convex problem.

In some two-level structures, the MPC in high-level is not necessary. The trajectory can generate by some other method and use MPC to follow the trajectory and make the trajectory smooth in the low-level. The [12] uses 3D potential field method to get a reference trajectory and then use MPC to track the trajectory based on the dynamic model. The potential field can help the vehicle to avoid the obstacle, so there is no need to add the obstacle information in the MPC. In [6], graph-based route selection algorithm can build a probability graph and select a route with the lowest collision probability. In low-level control, the authors use a longitudinal MPC and a lateral

MPC to track the trajectory based on the kinematic model. In [4] where a lidar-based method was proposed. The obstacle information is obtained from lidar, and the authors divide the area into safe and dangerous. Using Dijkstra or A* method is used to generate a collision-free path based on the safe area. In the low-level controller, an MPC is used based on dynamic model to track the path. Compared with MPC, this method can save more time and respond faster, but some of the generated reference trajectories are hard for low-level MPC to track because the reference paths are not smooth enough.

2.2 Moving obstacles

In the real-life, most obstacles are not static. There are many moving obstacles, especially when we consider the overtaking problem. There are significant differences between a static obstacle and a moving obstacle in overtaking problem. The paper [13] divides the overtake problem into three phases. The first phase is diverting from the lane, and the second phase is driving straight in the adjacent lane. The last phase is returning to the lane. In [13], the authors only consider the first phase. They use a 5th-order polynomial method to optimize the trajectory based on minimizing the jerk of the ego vehicle with a point mass model. Thus, they ignore the obstacle information, which is not safe for vehicle. The paper [14] and paper [15] only considered the second phase. These methods also use 5th-order polynomial method to optimize reference trajectory in longitude direction based on the point-mass model. The methods are based on minimizing the acceleration or jerk. The methods are robust in different weather, but they did not consider the obstacle information in every part, which is not safe enough.

A new control method for the automated overtaking vehicle allows the vehicle to track the reference trajectory [16]. This method sets a goal point around the obstacle based on the information of the obstacle at each phase, and the continuous reference trajectories for every phase using third-order polynomial method based on minimizing the jerk has been generated in real-time. The feedback controller is based on the kinematic model in [16].

Similar to dealing with static obstacles, [19] and [20] also use the soft constraint to deal with moving obstacles. In [20], the Euclidean distance between ego car and leading car is added to the cost function and the prediction model is based on the dynamic model. In [19], the TTC is obstacle information and the MPC is based on kinematic model. It is easy for soft constraints to ensure avoiding the moving obstacles when the MPC cost function is well defined, but it is not easy to avoid obstacles in non-convex problems.

The previous methods are based on time-domain calculations. In [17] and [18], in order to avoid a non-convex formulation, they turned time-domain into spatial-domain based on a point-mass model. So, the hard constraint becomes the convex constraint and the MPC problem is easier to solve without the initial guess. However, the equations become less intuitive, and it is more difficult to modify them in a targeted manner.

The Trajectory boundaries of the vehicle were added in the MPC as the hard constraint in [21]. At first, the boundaries can be drawn based on the prediction of leading vehicle's movement in different situations. Then the boundaries' information to the MPC and the vehicle is controlled in the boundaries. Because the vehicle is driving on the road, a safe distance is also added to the constraints. The MPC is based on the kinematic model and is divided into two directions: longitude and lateral to control the vehicle.

The same as avoiding static obstacles, when the ego car overtakes the leading car, it also can use a two-level controller. The potential field was used to generate a collision-free path in real-time and a risk map in [22]. Based on the risk map and the collision-free path, the MPC was used

to control the ego vehicle to overtake the leading vehicle. The prediction model is a kinematic bicycle model, and it also considers incoming vehicles in another lane. In the paper, [23] uses a search method to obtain a safe path and uses a law which defines by the distance to obstacle generate lots of blocks, and these blocks make up the collision-free areas. The MPC can control the vehicle in the block by adding the information in the constraint. The kinematic model was used for prediction. However, these methods cannot decide when the ego car should overtake because it is not safe to overtake on the road in many dangerous situations. So, it will be better if the ego vehicle can predict the environment and deal with some emergencies.

The surrounding vehicle motion can be estimated based on using probabilistic methods. The ego vehicle obtained the probability density function based on the information from the environment in [24]. The method divides the driving style into three clusters and uses probability density functions fitted to each cluster's data. Based on the function, when the ego car knows the leading vehicle's moving, it can find a way to avoid them. The controller uses point-mass as the prediction model. Another way is to use the normal distribution to estimate the leading vehicle's movement [25]. Based on the normal distribution, the probability density function can be defined. According to the probability density function, the probability of collision, the probability of overtaking, and the probability of keeping following every position on the road can also be defined. Authors build a directed graph based on the road and find an optimized path on the graph based on the probabilities and use MPC based on the point-mass model to follow the trajectory. In [26], the authors also estimate leading vehicle's movement using the normal distribution. Based on the probability of the leading vehicle moving to every location, the ego vehicle can decide whether overtake or not. If it decides to overtake, the ego vehicle uses MPC to add the boundaries information to overtake based on point-mass model.

CHAPTER 3 CONTROL ARCHITECTURE

There are three layers for the architecture, as shown in Fig.2, to control the ego vehicle overtaking the leading vehicle: 1) generating the reference trajectory, 2) using an MPC controller as the high-level controller to generate the control input, and 3) using a PID controller as the low-level controller to control the ego vehicle based on the dynamic model to overtake the leading vehicle.

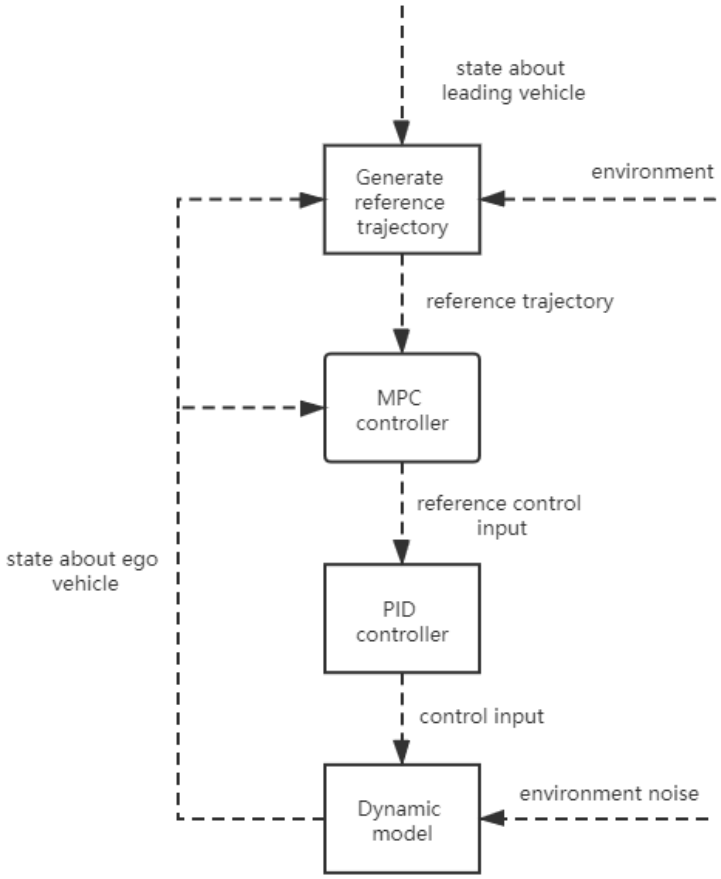

Fig. 2. Control Architecture

3.1 Reference trajectory planning

The first layer is trajectory planning. It will generate the reference trajectory for the MPC. we use two different methods for trajectory generation: A* and minimum-jerk method.

3.1.1 A* method

The A* method is a graph search-based method [1]. It is based on the grid graph. The cost function of A* is the sum of the distance travelled and the heuristic function which is defined by the Euclidean distance from the ego vehicle position to the goal position. In this dissertation, the leading vehicle is a moving obstacle while most A* methods focus on the static obstacle. So, in

our implement A* method, the leading vehicle movement will be predicted by the point-mass model. The area where the leading vehicle may move in the future is used as a planned obstacle. Second, the reference trajectory will be replanned before every MPC step, and each trajectory generated by the real-time A* algorithm will be a straight line.

3.1.2 Minimum-jerk method

The second method to generate the reference trajectory is the minimum-jerk method. This method is a trajectory generation method based on optimizing [13]. Each segment of the trajectory between two points can be transformed into a 7-order polynomial. The cost function to generate the coefficients of the polynomial is:

$$\mathrm{J} = \sum_{k=0}^{T} \|\dot{a}\|^2 \tag{1}$$

$\dot{a}$ is the jerk of ego vehicle, T is the time allocated between every point, and we need to minimize the cost function to get the coefficients of the polynomial. The way to set the two endpoints of each polynomial is vital for overtaking the leading vehicle. First, we define a safe distance d_s, When starting and ending overtaking, the distance between the two vehicles is at least a safe distance d_s. So, we can set two points at the position on starting and ending overtaking based on predicting the leading vehicle moving by a point-mass model. The point should be set in the original lane before the start of overtaking and after the end of overtaking. During overtaking, the point should be set in the lane next lane. The reference trajectory will be smooth, and it will be easy for the vehicle to follow.

3.2 Vehicle modelling

The vehicle model is a vital concept that should be considered in the field of autonomous vehicles. If we model the vehicle very accurately, the disadvantages are computationally cost and complex model. However, If the model is too simple, the tracking prediction error will be particularly large. So, we want to introduce three basic modelling methods first.

3.2.1 Point-mass model

It is a simple model in order to reduce the computational burden of the control algorithm. So, the model can write as:

$$\dot{\xi}(t) = f\big(\xi(t), u(t)\big) \tag{2}$$

The five states are lateral and longitudinal velocities in the body frame, the yaw rate, lateral and longitudinal vehicle coordinates in the inertial frame. These are denoted respectively as $\xi = [\dot{y}, \dot{x}, \psi, Y, X]'$. We assume that the longitudinal velocity of the vehicle is constant. So the input $u(t)$ is the lateral acceleration,a_y.and the function can show as the following:

$$\ddot{y} = a_y \tag{3a}$$

$$\ddot{x} = 0 \tag{3b}$$

$$\dot{\psi} = \frac{a_y}{\dot{x}} \tag{3c}$$

$$\dot{Y} = \dot{x}\ \sin(\psi) + \dot{y}\cos(\psi) \quad (3d)$$

$$\dot{X} = \dot{x}\ \cos(\psi) - \dot{y}\sin(\psi) \quad (3e)$$

3.2.2 Kinematic bicycle model

The nonlinear continuous-time equations that describe a kinematic bicycle model (Fig .3) are written as:

$$\dot{x} = vcos(\psi + \beta) \quad (4a)$$

$$\dot{y} = vsin(\psi + \beta) \quad (4b)$$

$$\dot{\psi} = \frac{v}{l_r}\sin(\beta) \quad (4c)$$

$$\dot{v} = a \quad (4d)$$

$$\beta = \tan^{-1}(\frac{l_r}{l_r + l_f}\tan(\delta_f)) \quad (4e)$$

The x and y are the coordinates of the centre of mass in an inertial frame (X, Y). v and a are the velocity of the vehicle and the acceleration of the vehicle, respectively. The distance from the centre of the vehicle mass to the rear axles and front axles is l_r and l_f, respectively. β is the angle of the current velocity of the centre of mass with respect to the longitudinal axis of the vehicle. Because in most vehicles, the rear wheel cannot be steered, so the $\delta_r = 0$, and the control input is front steering angle and the acceleration, which is δ_f and a.

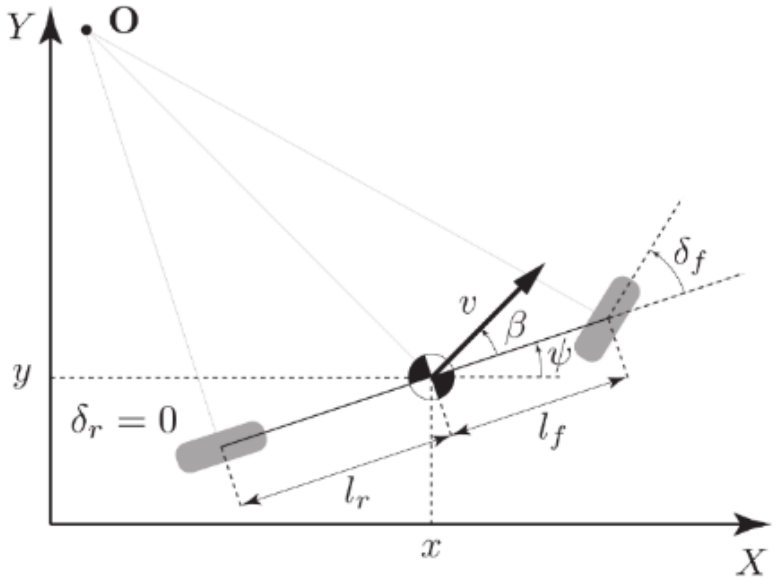

Fig. 3. Kinematic Bicycle Model [3]

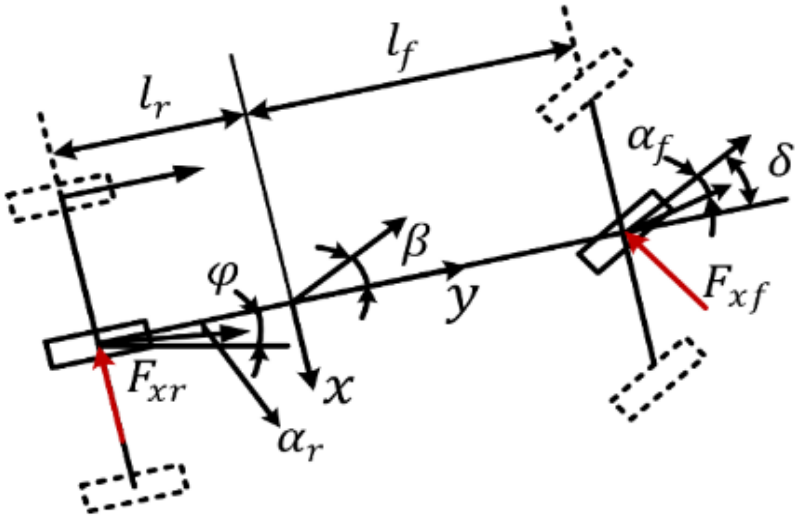

Fig. 4. Dynamic Bicycle Model [12]

3.2.3 Dynamic bicycle model

The dynamic bicycle model (Fig .4) has the same part as the kinematic, which is the inertial position coordinates and heading angle. There are differential equations in this case:

$$\ddot{y} = -\dot{\psi}\dot{x} + \frac{2}{m}\left(F_{xf}\cos\delta_f + F_{xf}\right) \tag{5a}$$

$$\ddot{x} = \dot{\psi}\dot{y} + a_x \tag{5b}$$

$$\ddot{\psi} = \frac{2}{I_z}\left(l_f F_{xf} - l_r F_{xr}\right) \tag{5c}$$

$$\dot{Y} = \dot{x}\ \sin(\psi) + \dot{y}\cos(\psi) \tag{5d}$$

$$\dot{X} = \dot{x}\ \cos(\psi) - \dot{y}\sin(\psi) \tag{5e}$$

Where m and I_z is the car's mass and yaw inertia. In coordinate frames aligned with the wheels, the F_{xr} and F_{xf} is the lateral tire forces at the rear and front wheels, respectively.a_x is the longitudinal acceleration of the vehicle.

The F_{xr} and F_{xf} is defined by tire model:

$$F_{xf} = C_f\alpha_f \tag{6a}$$

$$F_{xr} = C_r\alpha_{fr} \tag{6b}$$

C_f is the linearized cornering stiffness of the front wheel. C_r is the stiffness about rear wheel. α_f and α_r are the tire slip angle of the front wheel and rear wheel. This dissertation uses a dynamic model of the ego vehicle to generate the ego vehicle state.

3.3 Model predict control

Model Predictive Control (MPC) approach is the most popular technique to solve autonomous vehicle problems. Although the MPC has some disadvantages, such as the high computation cost in the autonomous vehicle problem, it provides an easy way to add the constraints in the design. Furthermore, the MPC can also predict the future state, which is vital in the design.

3.3.1 Periodic MPC

A basic MPC formulation can divide three parts: cost function, vehicle model, and constraints. In the cost function, the state may be compared with the reference state, and in some papers, the obstacle information will add to the function. In the constraints part, it has the boundary of the state and may have the obstacle information. Only the first control input is applied to the vehicle model at each time step.

A general formulation of the MPC can be written as:

$$min \sum_{k=0}^{N_P-1} J_k\left(X_k, U_k, X_k^{ref}, U_{k-1}\right) \tag{7a}$$

s.t.:

$$X_{k+1} = f(X_k, U_k), \qquad k = 0, \dots, N_P - 1 \tag{7b}$$

$$G(X_k, U_k) \leq g_b, \qquad k = 0, \dots, N_P - 1 \tag{7c}$$

$$X_0 = X_{init} \tag{7d}$$

In formulation (7), U_k is the input in the MPC. X_k is the state at k, and X_0 is the state when $k = 0$. X_{init} is the current state updated online at every sampling step. , X_k^{ref} is the state provided by the reference value. The function J_k is the cost function and the function G comprises of all constraints with g_b being the bound value. N_P is the prediction horizon. In formulation (7b), the predictive model will be the linearized kinematic model.

3.3.1.1 Cost function

The cost function is based on the system state and the input. In order to generate a flexible and smooth trajectory, the cost function is written as:

$$J_k\left(X_k, U_k, X_k^{ref}, U_{k-1}\right) = \left\|X_k - X_k^{ref}\right\|_Q^2 + \|U_k\|_R^2 + \|U_k - U_{k-1}\|_S^2 \qquad k = 0, \dots, N_P - 1 \tag{8}$$

where Q, R and S are weighting matrices of appropriate dimensions. The cost function considers the deviations of the state X_k ,the state reference state X_k^{ref} ,the control input U_k and the change about the control input which is $U_k - U_{k-1}$.

3.3.1.2 Constraints

The width of the road constrains the lateral position of the vehicle.:

$$y_{road,min} < y < y_{road,max} \tag{9}$$

$y_{road,min}$ and $y_{road,max}$ represent the limits of the road. There is always a speed limit on the road. So, there is also a constraint about velocity:

$$v < v_{road,max} \tag{10}$$

Second, the variation of the steering angle δ_f and the acceleration a are constrained, as:

$$-\delta_{f,lim} < \delta_f < \delta_{f,lim} \tag{11}$$

$$-\Delta\delta_{f,lim} < \Delta\delta_f < \Delta\delta_{f,lim} \tag{12}$$

$$-a_{lim} < a < a_{lim} \tag{13}$$

Where $\delta_{f,lim}$ is the steering angle limits, $\Delta\delta_{f,lim}$ is limitation of the rate of change of steering angle and a_{lim} is the acceleration limits , which are determined by the physical properties and comfort of passengers.

3.3.1.3 Predictive model

The initial kinematic model is nonlinear, it can be linearized by seeking its Jacobian matrix first. The Jacobian matrix of kinematic model can be defined as A' and B'. So the linearized model can write as:

$$\dot{X} = f(X, U) \approx A'X + B'U \tag{14}$$

And we get a discrete-time model with Forward Euler Discretization with sampling time d_t:

$$X_{k+1} = X_k + f(X_k, U_k)d_t \tag{15}$$

We can use first degree Taylor expansion around $\bar{X}$ and $\bar{U}$ which is the state estimate by ego vehicle:

$$X_{k+1} = X_k + (f(\bar{X}, \bar{U}) + A'X_k + B'U_k - A'\bar{X} - B'\bar{U})d_t \tag{16}$$

So, we can get:

$$X_{k+1} = AX_k + BU_k + C = f(X_k, U_k) \tag{17}$$

Where:

$$A = (I + d_t A') \tag{18a}$$

$$B = d_t B' \tag{18b}$$

$$C = (f(\bar{X}, \bar{U}) - A'\bar{X} - B'\bar{U})d_t \tag{18c}$$

3.3.2 Event-triggered MPC

In periodic MPC, the U_k is regarded as the control input, and $U_{k+1}, U_{k+2}, U_{k+3} \ldots\ldots U_{k+N_P-1}$ are discarded in next MPC runs. It will waste computing resources. So, we propose an event-triggered MPC which uses the generated control inputs more efficiently. We define the error as the absolute percentage error between current ego vehicle states and predicted states generated by the MPC. Meanwhile, keeping the error within a reasonable range is a vital task. So, before proceeding to the next MPC step, there is another condition :

$$\left|\frac{X_k - X_{e,k}}{X_k}\right| \times 100\% \leq \varepsilon_{con}, \quad k = 1, \ldots, N_P - 1 \tag{19}$$

The condition (19) means that the predictive state X_k need to compare with the real ego vehicle

current state $X_{e,k}$. If the difference is less than the limit ε_{con}, the, U_{k+1} will be the next control input and skip the MPC computations until the absolute percentage error is more than the limit ε_{con}. Compared with periodic MPC, the event-triggered MPC has an additional judgment condition, as shown in Fig. 5 . It can help MPC to use the calculated optimal input by MPC more efficiently.

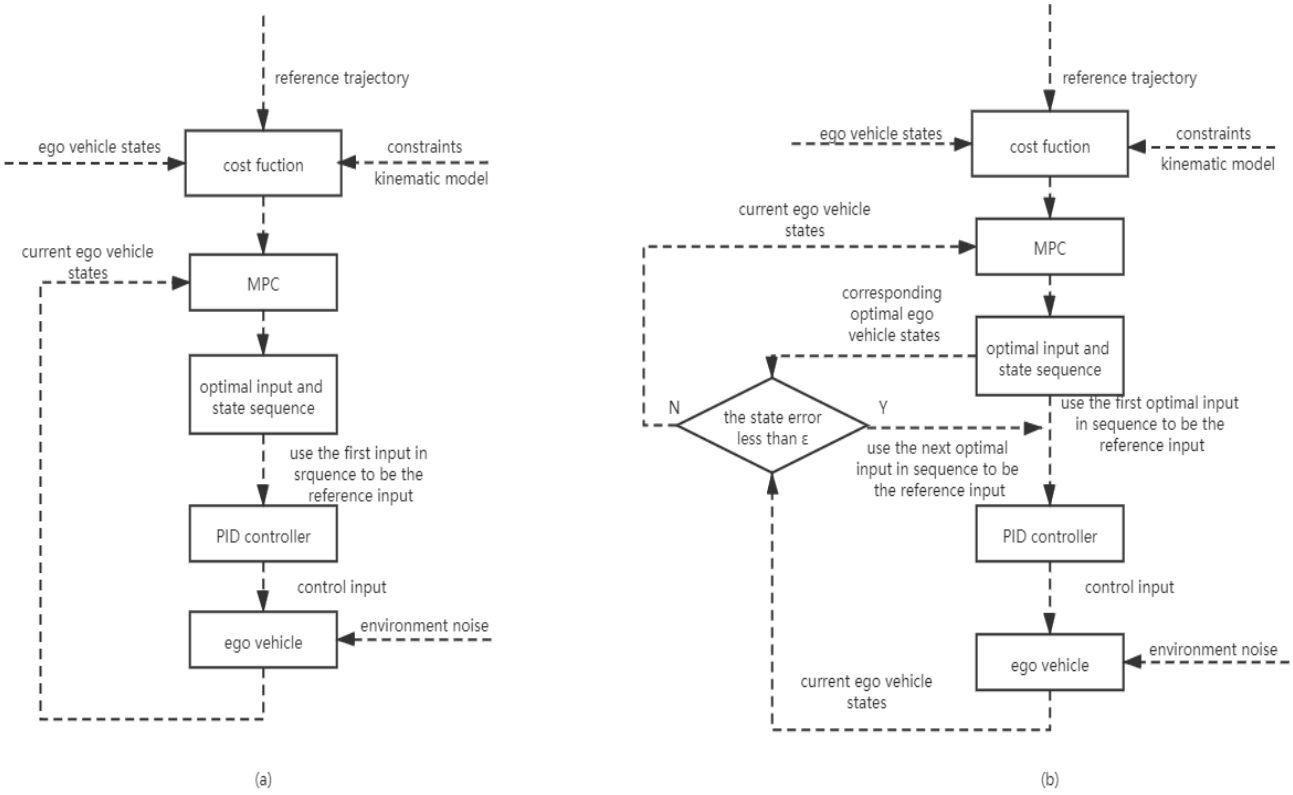

Fig. 5. (a) Motion control using a periodic MPC (b) Motion control using an event-triggered MPC

CHAPTER4 EXPERIMENTAL RESULTS

The control architecture we proposed, shown in Fig.2, has three main parts which are motion planning, MPC, and low-level controller. We consider two methods for motion planning: reference trajectory generated by A* and generate by minimum-jerk method. For each method, the performance of the event-triggered MPC is compared with the performance of periodic MPC. The ego vehicle model parameters used in simulation are summarized in Table I and the parameters of MPC controller are shown in Table II. The threshold parament ε_{con} in event-triggered MPC is setting to ε_{con} is 5% for lateral position and yaw. For other states, this parameter is 10%. This is because the lateral position and yaw angle are the most important in overtaking problems. The MPC formulation is solved by CVXPY solver [27].

Table I. Parameters of the ego vehicle

Parameters of the ego vehicle		[Units]
Distance from centre of mass to rear axles	$l_r = 2.5$	$[m]$
Distance from centre of mass to front axles	$l_f = 2.5$	$[m]$
Vehicle mass	$m = 2020$	$[Kg]$
Rear cornering stiffness	$C_r = -3200$	$[N/Rad]$
Front cornering stiffness	$C_f = -3400$	$[N/Rad]$
Inertia	$I_z = 2850$	$[Kg/m^2]$

Table II. Parameters of MPC controller

Parameters of MPC Controller		[Units]
Road boundary	$y_{road,max} = 4$	$[m]$
Road boundary	$y_{road,min} = -4$	$[m]$
Velocity limits	$v_{road,max} = 15$	$[m/\mathrm{s}]$
acceleration limits	$a_{lim} = 1.3$	$[m/s^2]$
Steering angle limits	$\delta_{lim} = 30$	[°]
Rate of change of steering angle limits	$\Delta\delta_{lim} = 30$	[°/s]
Weighting matrix	$Q = \mathrm{diag}(1.0,1.0,0.5,0.5)$	
Weighting matrix	$R = \mathrm{diag}(0.01,0.01)$	
Weighting matrix	$S = \mathrm{diag}(0.1,0.1)$	
Horizon Length	$N_p = 5$	

4.1 Scenario 1: A* method

In this scenario, the reference trajectory is generated by a real-time A* method. The ego vehicle's initial velocity is 5m/s and the leading vehicle's velocity is 4m/s which is constant, and the safe distance is 45 meters. Fig.6 shows a trajectory of the ego vehicle where the ego vehicle can overtake the leading vehicle within the road boundary.

According to Table. III, as the sampling time gradually increases, the required number of MPC runs gradually become less. The significant changes are in the yaw angle and lateral position. Based on the total calculation time, the event-triggered MPC requires less time for computing the optimal trajectory. When the sampling time is 0.5s, the event-triggered MPC can only save 7 MPC runs which are almost 1.5s in total. Compared with periodic MPC, as shown Table III, the event-triggered MPC can save up to half the number of MPC runs when the sample time is 0.1s. However, the average calculation time per step in periodic is always less than event-triggered MPC. It is because the percentage of error in event-triggered MPC is bigger, so it needs more time for calculations.

The mean percentage error will become larger if the sampling time becomes large, as shown in Fig.7. As the sampling time increases, the average percentage error of lateral position increases from 1.03% to 3.61% in periodic MPC and from 1.19% to 3.97% in event-triggered MPC. The average absolute percentage error in yaw angle changes from 2.44% to 3.94% in periodic MPC and in event-triggered MPC, it changes from 3.28% to 5.62%.

When we focus on the mean absolute percentage error with a sampling time of 0.2 seconds shown as Fig. 8, large errors often occur in two phases. The first phase is in the early stages of overtaking, that is, when the vehicle is about to turn to another lane. The second phase is in the latter part of the overtaking when the vehicle returns to its original lane. Fig. 8 shows that the large error of the yaw angle tends to appear earlier than the lateral position.

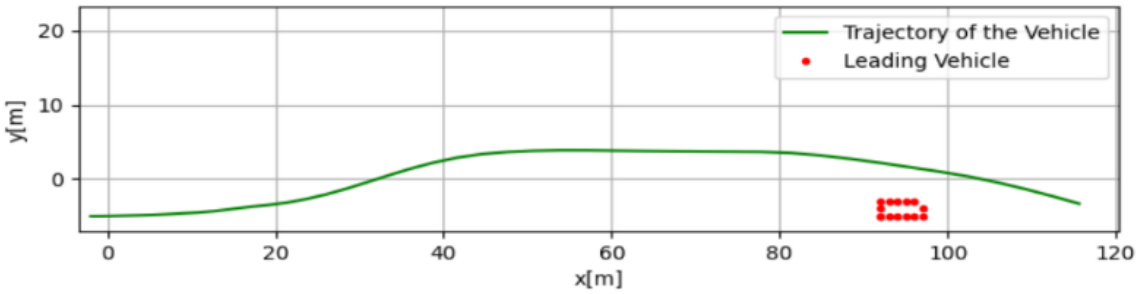

Fig. 6. Ego vehicle trajectory

The number of MPC runs reduced because the sampling time increased, which means the frequency of correcting errors is reduced, so the error of the system state before each MPC run will increase. The results show good performance of the periodic MPC controller and event-

triggered MPC controller to track the reference path and also proves that we need to reduce the threshold of the lateral position and yaw angle to ensure that the error is within a controllable range. Event-triggered MPC can save up to 50% of the calculation cost and reduce the calculation time as much as possible when the error is within the acceptable range.

Table III. The number of MPC runs and calculation time of Periodic MPC and Event-triggered MPC

Sample time d_t (s)	Periodic MPC			Event-triggered MPC		
	Number of MPC runs	Average calculation time per MPC step(s)	Total calculation time(s)	Number of MPC runs	Average calculation time per MPC step(s)	Total calculation time(s))
0.1	127	0.09	11.62	63	0.09	5.98
0.2	58	0.14	8.11	31	0.15	4.72
0.3	38	0.21	7.87	21	0.27	5.66
0.4	30	0.27	7.99	20	0.34	6.81
0.5	24	0.33	7.81	17	0.36	6.26

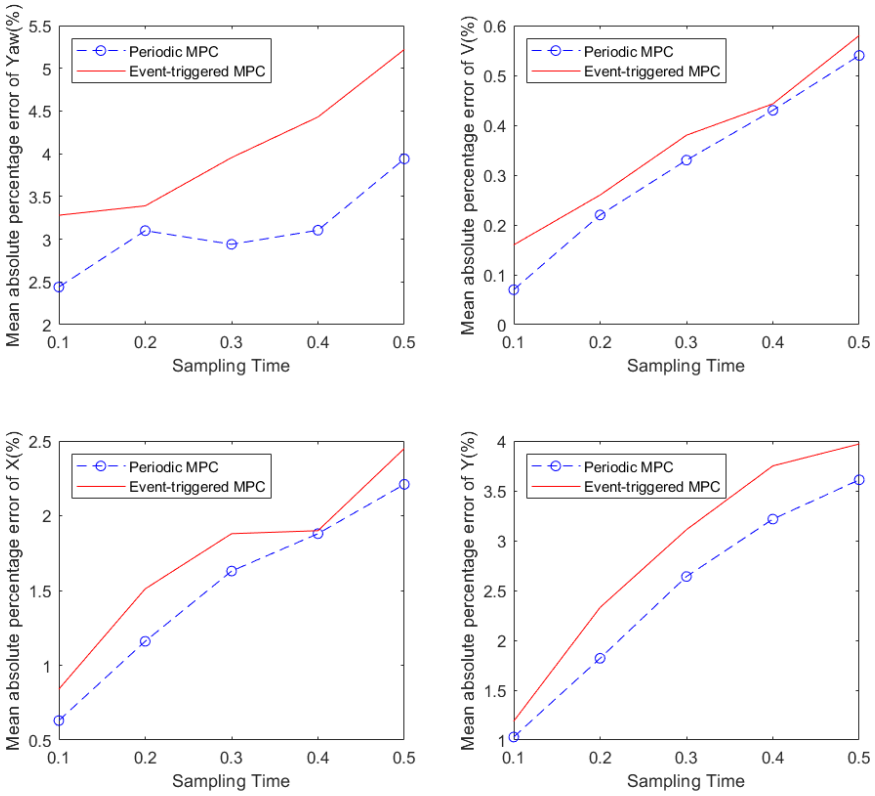

Fig. 7. Mean absolute percentage error of the ego vehicle state

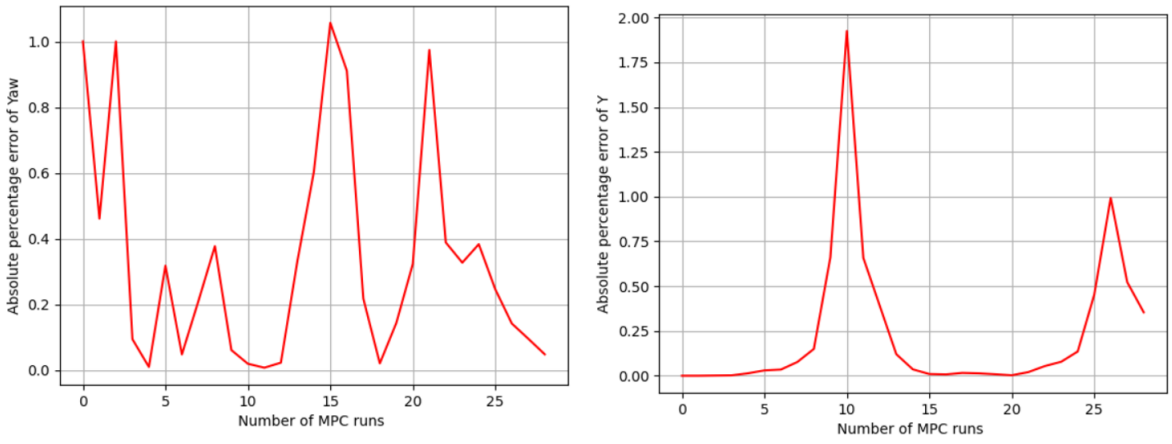

Fig. 8. Absolute percentage error of yaw angle and lateral position (event-triggered MPC, $d_t = 0.2s$)

4.2 Scenario 2: minimum-jerk method

4.2.1 Dry road

In this scenario, the reference trajectory is given by the real-time minimum-jerk method based on an optimization problem. The initial conditions are the same as scenario one. Due to the characteristics of this method, the reference trajectory, different from A* method, is smooth.

Fig.9 indicates the ego can track the smooth trajectory successfully. A comparison between Table III and Table IV shows the difference between A* and the minimum-jerk method. When the vehicle uses event-triggered MPC, the number of MPC runs has not decreased, but the total calculation time has decreased. When the sampling time is 0.1 seconds, the average calculation times of A* and minimum-jerk method are similar. But with the gradual increase of the sampling time, the time consumed by each step begins to gradually decrease, which shows that a smooth trajectory can increase the computation speed.

Fig.10 shows the absolute percentage of error for all four states. The difference between the average error of lateral position and speed is not large, and the average error of the heading angle is smaller when the A* algorithm is used. On the contrary, the average error of the longitudinal position is bigger when using the A* algorithm, especially when the sampling time is greater than 0.3 seconds.

When a vehicle follows a smooth trajectory, the vehicle needs to adjust the steering angle at any time, so the error of the yaw angle can quickly increase. However, the smooth trajectory is easy for MPC to control the vehicle compared with discount segment generate by A* method, because the vehicle system is a nonholonomic system. It is the reason that the absolute percentage error of the lateral position is much less when the reference trajectory is smooth.

The smooth trajectory can save more calculation cost when the ego vehicle uses event-triggered MPC according to the simulation because the absolute percentage error will be reduced especially in two phases what mentioned before. Also, the vehicle still can overtake the leading vehicle safely.

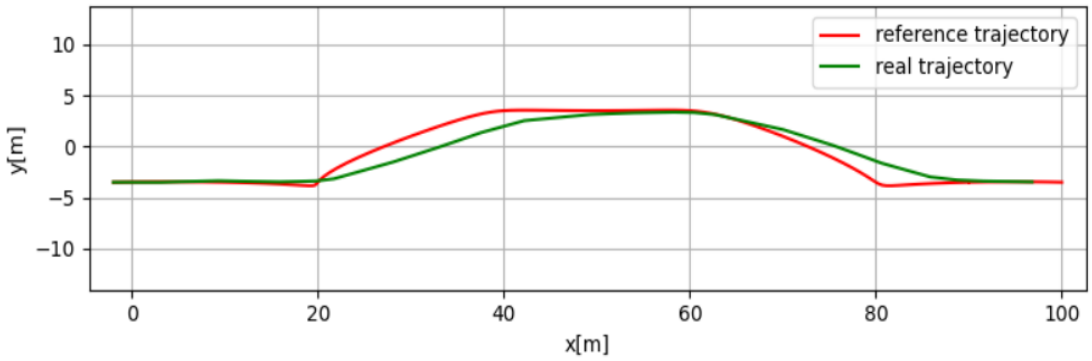

Fig. 9. Reference trajectory and ego vehicle trajectory

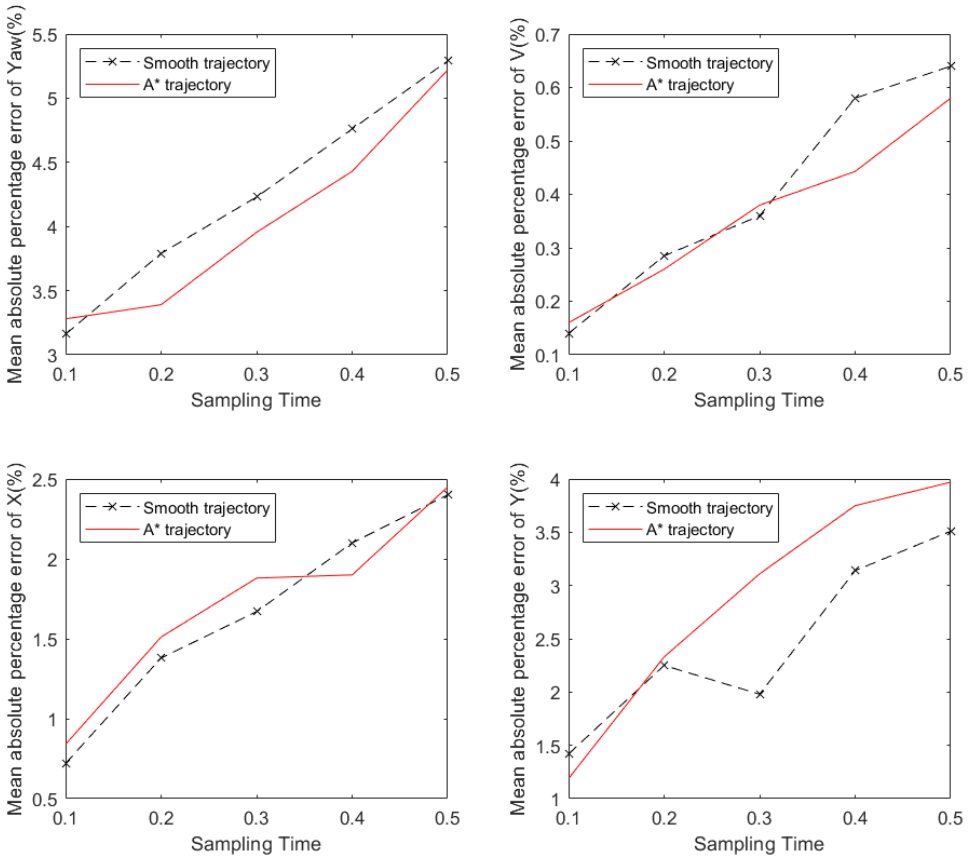

Fig. 10. Mean absolute percentage error of the ego vehicle state in different scenarios (event-triggered MPC)

Table IV. The number of MPC runs and calculation time of Event-triggered MPC with smooth reference trajectory

Sample time d_t (s)	Event-triggered MPC		
	Number of MPC runs	Average calculation time per MPC step(s)	Total calculation time(s)
0.1	52	0.10	5.19
0.2	33	0.13	4.46
0.3	29	0.19	5.60
0.4	20	0.24	4.75
0.5	18	0.31	5.43

4.2.2 Icy road

Different from ordinary dry roads, icy roads have more uncertainty. This is mainly because when the car is driving on a frozen road, the friction coefficient of the road is greatly reduced, so the acceleration of the car and the lateral force of the tire will be strongly affected. In this part, we assume that the initial conditions are the same as the situation when the road is dry and the friction coefficient of a frozen road is 0.4 times that of a dry road. Therefore, when the ego vehicle is modelled by dynamic model, the acceleration and tire sideslip force that acts on the vehicle are adjusted to:

$$F_{xf,ice} = 0.4F_{xf} \tag{20a}$$

$$F_{xr,ice} = 0.4F_{xr} \tag{20b}$$

$$a_{ice} = 0.4a \tag{20c}$$

Although the overtaking was completed on the frozen road shown as Fig.11, by comparing Table IV and Table V, we can see that the number of MPC runs and the calculation time of each run have increased. The total calculation time is one to two seconds longer than that on dry roads.

It can be concluded from Fig. 12 that the average absolute percentage error of each state has improved compared with driving on the dry road, and the longer the sampling time, the bigger the error. We can also see that when the sampling time is 0.5 seconds, the mean absolute percentage error of the steering angle is close to 6%, and the mean absolute percentage error of the lateral position is even closer to 10%. The mean absolute percentage error of the longitudinal position has also increased by 1% at each sampling time. The smallest error change in the error is in the velocity of the ego vehicle because it is difficult to accelerate and decelerate due to the icy road, but it does not affect the vehicle to maintain the original speed.

As shown in previous experiments, when the error is larger, the calculation time required for event-triggered MPC is also longer, and the number of MPC runs that can be skipped less. The

source of the increased error is bad weather. On icy roads, due to the significant reduction in friction, the input of the vehicle system will be affected, so the control of the vehicle is more difficult.

It can be seen from Fig .12 that when the sampling time is bigger than 0.4 seconds, the lateral error begins to become unacceptable. Moreover, increasing the sampling time cannot save the calculation cost, as shown in Table V. Therefore, when encountering cold weather, event-triggered MPC needs to reduce the sampling time to obtain better performance.

Table V. The number of MPC runs and calculation time of Event-triggered MPC on icy road

	Event-triggered MPC		
Sample time d_t (s)	Number of MPC runs	Average calculation time per MPC step(s)	Total calculation time(s)
0.1	70	0.10	6.98
0.2	41	0.14	5.74
0.3	29	0.23	6.62
0.4	21	0.36	7.76
0.5	18	0.39	7.02

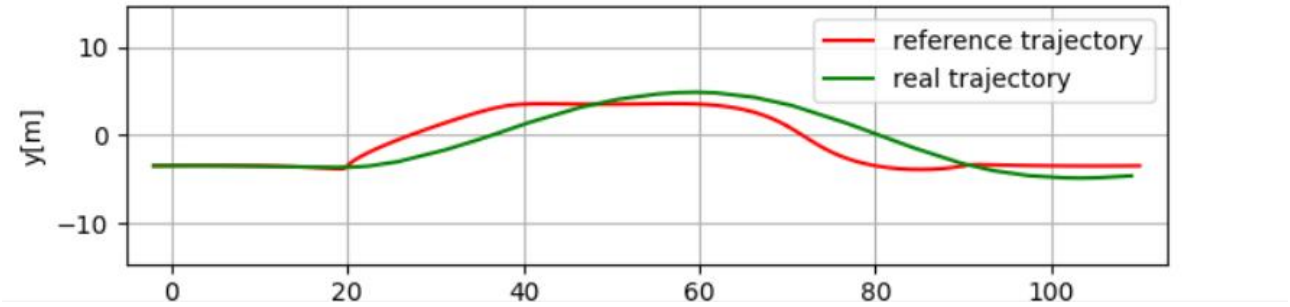

Fig. 11. Reference trajectory and ego vehicle trajectory on icy road

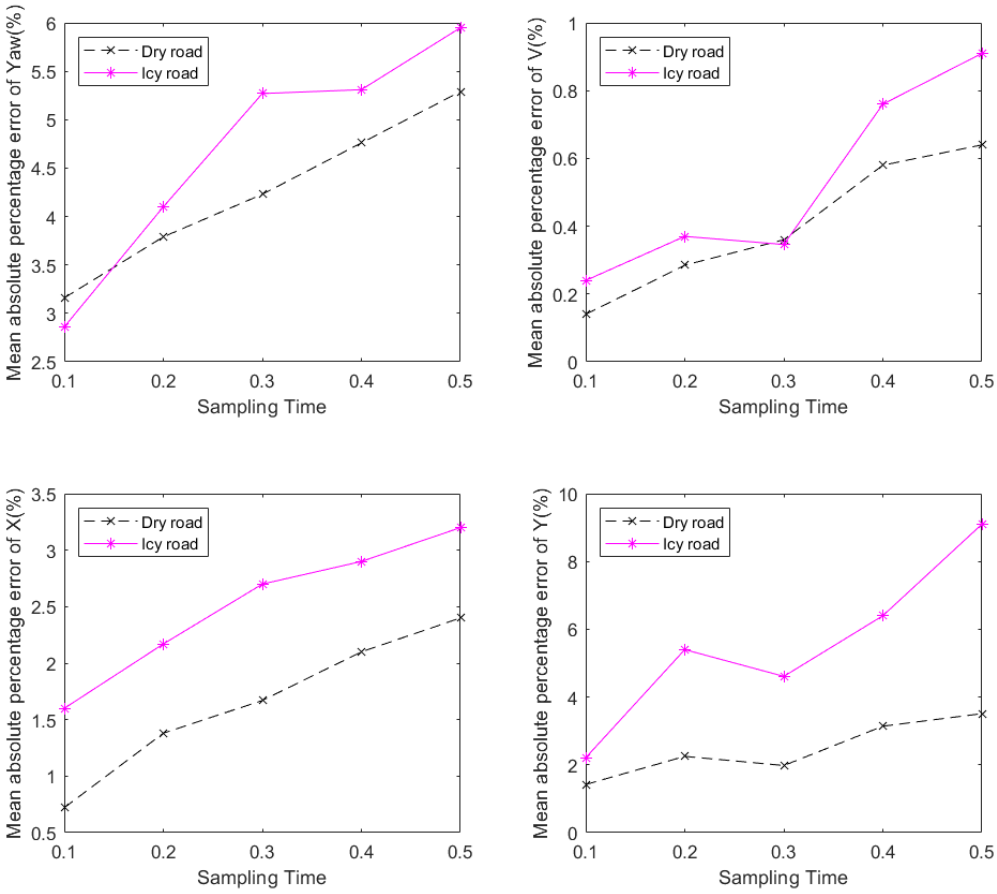

Fig. 12. Mean absolute percentage error of the ego vehicle state in different types of roads (event-triggered MPC)

CHAPTER 5 CONCLUSIONS AND FUTURE DIRECTIONS

In this dissertation, we have developed an overtaking control architecture for autonomous cars using Model Predictive Control. We also proposed an event-triggered MPC, which can save 50% computing resources. Under this method, the error thresholds ε_{con} of yaw angle and lateral position are proved to be stricter than other states to control errors. At the same time, the smooth trajectory is proven to save more computational costs in the overtaking problem. When the road becomes icy the sample time of event-triggered MPC need to be set as a small number.

The smooth trajectory generated method needs to set the future positions in advance and is not real-time. So, a fast real-time smooth collision-free trajectory generation method is one of the future research directions. At the same time, looking for a better control threshold for each state to achieve a balance between computing resources and errors is also a topic of future research.

REFERENCES

[1] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968,

[2] M. A. Abbas, R. Milman and J. M. Eklund, "Obstacle Avoidance in Real Time With Nonlinear Model Predictive Control of Autonomous Vehicles," in Canadian Journal of Electrical and Computer Engineering, vol. 40, no. 1, pp. 12-22, winter 2017.

[3] J. Kong, M. Pfeiffer, G. Schildbach and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea (South), 2015, pp. 1094-1099.

[4] Yongsoon Yoon, Jongho Shin, H. Jin Kim, Yongwoon Park, Shankar Sastry, "Model-predictive active steering and obstacle avoidance for autonomous ground vehicles," in Control Engineering Practice,Volume 17, Issue 7,2009,pp. 741-750.

[5] J. Liu, P. Jayakumar, J. L. Stein and T. Ersal, "Combined Speed and Steering Control in High-Speed Autonomous Ground Vehicles for Obstacle Avoidance Using Model Predictive Control," in IEEE Transactions on Vehicular Technology, vol. 66, no. 10, pp. 8746-8763, Oct. 2017.

[6] Gao, Y, Lin, T, Borrelli, F, Tseng, E, & Hrovat, D. "Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads." *Proceedings of the ASME 2010 Dynamic Systems and Control Conference*. *ASME 2010 Dynamic Systems and Control Conference, Volume 1*. Cambridge, Massachusetts, USA. September 12–15, 2010. pp. 265-272. ASME.

[7] B. Németh, T. Hegedűs and P. Gáspár, "Model Predictive Control Design for Overtaking Maneuvers for Multi-Vehicle Scenarios," 2019 18th European Control Conference (ECC), Naples, Italy, 2019, pp. 744-749,.

[8] X. Zhang, A. Liniger and F. Borrelli, "Optimization-Based Collision Avoidance," in IEEE Transactions on Control Systems Technology, vol. 29, no. 3, pp. 972-983, May 2021.

[9] X. Zhang, A. Liniger, A. Sakai and F. Borrelli, "Autonomous Parking Using Optimization-Based Collision Avoidance," 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 2018, pp. 4327-4332,.

[10] Nishant Chowdhri, Laura Ferranti, Felipe Santafé Iribarren, Barys Shyrokau, "Integrated nonlinear model predictive control for automated driving "Control Engineering Practice,Volume 106,2021,104654,ISSN 0967-0661.

[11] U. Rosolia, S. De Bruyne and A. G. Alleyne, "Autonomous Vehicle Control: A Non-convex Approach for Obstacle Avoidance," in IEEE Transactions on Control Systems Technology, vol. 25, no. 2, pp. 469-484, March 2017.

[12] J. Ji, A. Khajepour, W. W. Melek and Y. Huang, "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints," in IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 952-964, Feb. 2017,.

[13] T. Shamir, "How should an autonomous vehicle overtake a slower moving vehicle: design and analysis of an optimal trajectory," in IEEE Transactions on Automatic Control, vol. 49, no. 4, pp. 607-610, April 2004.

[14] M. BHANA, “Trajectory generation for safe overtaking maneuver inautonomous vehicles : Evaluated in lane merging scenario utilizing a trajectory planner,” dissertation, 2018.

[15] L. Truong, “Evaluation of Prediction Models in Trajectory Planning for Overtaking Maneuvers.,” dissertation, 2019.

[16] P. Petrov and F. Nashashibi, “Modeling and Nonlinear Adaptive Control for Autonomous Vehicle Overtaking,” in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 4, pp. 1643-1656, Aug. 2014.

[17] V. S. Chipade, Q. Shen, L. Huang, N. Ozay, S. Z. Yong and D. Panagou, “Safe Autonomous Overtaking with Intention Estimation,” 2019 18th European Control Conference (ECC), 2019, pp. 2050-2057.

[18] J. Karlsson, N. Murgovski and J. Sjöberg, “Temporal vs. spatial formulation of autonomous overtaking algorithms,” 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 1029-1034.

[19] M. Obayashi, K. Uto and G. Takano, “Appropriate overtaking motion generating method using predictive control with suitable car dynamics,” 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 4992-4997.

[20] A. Buyval, A. Gabdulin, R. Mustafin and I. Shimchik, “Deriving overtaking strategy from nonlinear model predictive control for a race car,” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 2623-2628.

[21] R. Lattarulo, D. He and J. Pérez, “A Linear Model Predictive Planning Approach for Overtaking Manoeuvres Under Possible Collision Circumstances,” 2018 IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1340-1345.

[22] S. Dixit et al., “Trajectory Planning for Autonomous High-Speed Overtaking in Structured Environments Using Robust MPC,” in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 6, pp. 2310-2323, June 2020.

[23] N. Murgovski and J. Sjöberg, “Predictive cruise control with autonomous overtaking,” 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 644-649.

[24] Noureldin, Aboelmaged Németh, Balázs Gáspár, PéterHegedűs Tamás" Optimal Control of Overtaking Maneuver for Intelligent Vehicles"in Journal of Advanced Transportation,2018.

[25] Támas Hegedűs, Balázs Németh, Péter Gáspár,Graph-based Multi-Vehicle Overtaking Strategy for Autonomous Vehicles,IFAC-PapersOnLine,Volume 52, Issue 5,2019,Pages 372-377,ISSN 2405-8963.

[26] D. Adelberger, M. Wang and L. del Re, “Decision making through stochastic maneuver validation for overtaking on country roads,” 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 3487-3493.

[27] Steven Diamond and Stephen Boyd,” CVXPY: A Python-embedded modeling language for convex optimization”,Journal of Machine Learning Research,2016,17,83,pages 1--5