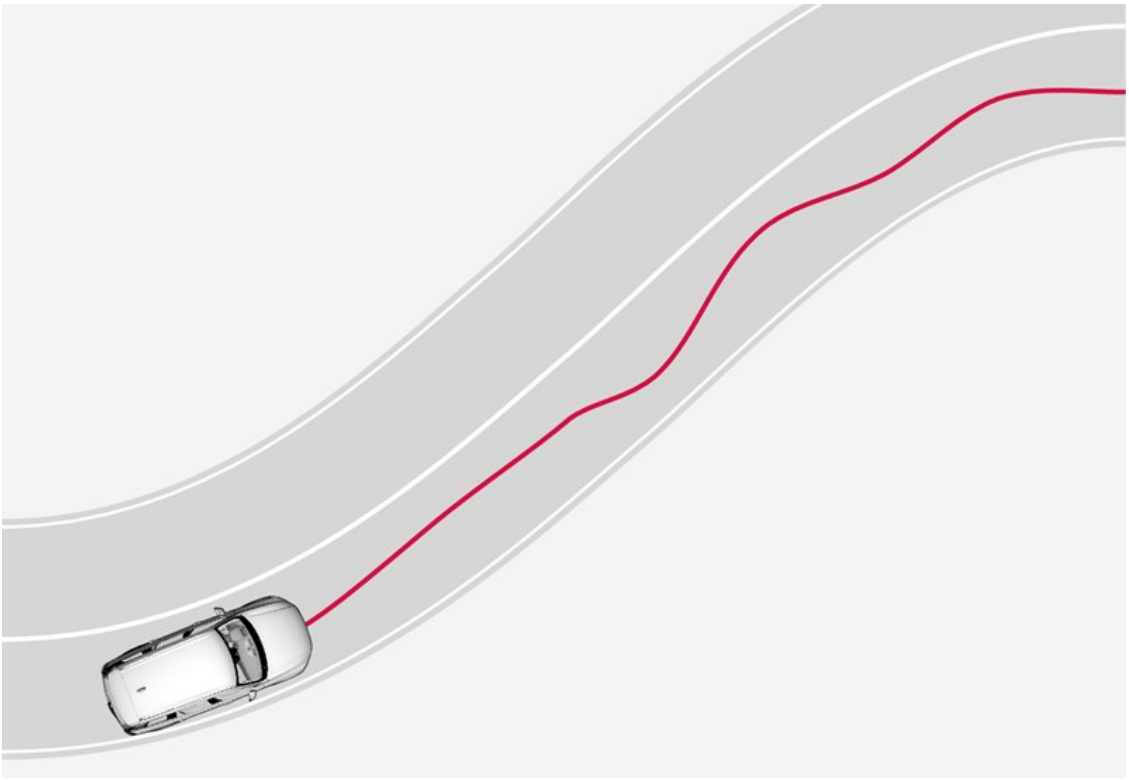
CHALMERS
UNIVERSITY OF TECHNOLOGY



Vehicle Lateral Motion Control with Performance and Safety Guarantees

Master's thesis in Systems, Control and Mechatronics

LEI NI

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's thesis

Vehicle Lateral Motion Control with Performance and Safety Guarantees

LEI NI

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Division of Automatic control, Automation and Mechatronics
Mechatronics group
Chalmers University of Technology
Gothenburg, Sweden 2015

Vehicle Lateral Motion Control with Performance and Safety Guarantees
LEI NI

Examiner: Paolo Falcone, Department of Signals and Systems
Supervisor: Lars Johannesson, Volvo Cars

Master's Thesis
Department of Signals and Systems
Division of Automatic control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Volvo Cars Driver Alert Control [1]

Typeset in LaTeX
Gothenburg, Sweden 2015

Vehicle Lateral Motion Control with Performance and Safety Guarantees
Lei Ni
Department of Signals and Systems
Chalmers University of Technology

Abstract

Highly automated and cooperative vehicles could improve traffic safety and energy efficiency. One of the key functions for autonomous cars is a path-following algorithm satisfying both performance and safety requirements.

Model Predictive Controllers (MPC) has capability of systematically and explicitly taking into system constraints and control requirements. This thesis explores Model Predictive Controllers (MPC) techniques to address vehicle lateral motion control problem. Based on the review of bases of MPC and related results, a persistent feasible MPC is formulated for the path-following problem. Furthermore, an explicit solution to the formulated MPC is generated.

Simulation results show that the explicit controller is able to generate the control input which keeps the vehicle on a desired path and fulfils the performance and safety requirements.

Keywords: Model Predictive Control (MPC), Lateral Motion Control, Control Invariant Set, Autonomous Driving, Explicit MPC

Acknowledgements

I would like to thank Prof. Paolo Falcone as my examiner. Without his solid support and genius guide, I wouldn't have accomplished this work. Moreover, his passion and wisdom always inspire me. All these influences are the great gifts for my later career and life.

Second, many thanks to my supervisor Lars Johannesson to give me such a fantastic topic as thesis. Besides, his explanations of the function requirements, as well as his advice for technique solutions are of great value for me to understand the problem.

My special thanks go to my classmate Stephan Heinrich for the interesting discussions and helpful tips.

Lei Ni, Gothenburg, September 2015

Contents

List of Figures

List of Tables

1
Introduction

1.1 Motivation and Background

Road Safety is an important societal issue. The latest statistics show European Union (EU) efforts to reduce road deaths have achieved considerable success. However, Europe's roads still face a big challenge: In 2013, about 26,000 people died on the roads of the European Union (EU). Besides, for every death on Europe's roads there are an estimated 4 permanently disabling injuries such as damage to the brain or spinal cord, 8 serious injuries and 50 minor injuries. In an attempt to redeem all these loss, European Commission has set an ambition to halve road deaths by 2020 compared to 2011[2].

One of the efforts of making driving safer for drivers together with passengers and pedestrians is the development of active safety systems. The City Safety system by Volvo Car Group is an example: the system uses laser sensor to monitor an area in front of the vehicle; If City Safety determines a collision is unavoidable and the driver does not respond, it activates the vehicle's brakes and switches off the throttle.

An even more ambitious approach towards driving safety is the development of autonomous driving systems. Volvo Cars autopilot system is being tested on the Swedish road. Volvo Cars autonomous driving project, i.e. Drive Me [3], aspires to have 100 autonomous cars used by real-world customers on roads by 2017, and "By 2020, nobody shall be seriously injured or killed in a new Volvo". Newly developed autonomous driving functions, must comply with the functional safety standard [4] to ensure the system safety. This thesis addresses the safety issues related to motion control for autonomous driving applications.

1.2 Goal of this thesis

In ISO 26262 – Functional safety for road vehicles [4], Automotive Safety Integrity Level (ASIL) is defined. The required ASIL standard for certain function can be determined by performing hazard and risk analysis based on the exposure, severity, and controllability of a vehicle. Functionalities with likely potential for severely life-threatening or fatal injury in the event of a malfunction will be classified as ASIL D, which requires the safety goals determined to be achieved with highest level of assurance.

For an autonomous driving car, in many driving scenario (e.g. overtaking on highway), it can cause fatal injuries on road users if the car deviates too much from desired path. As a reflection of high level safety requirement on lateral control, total deviation from the desired path shall be within bounded values. Meanwhile, the comfort performance remains relevant and important as an autonomous car will still carry human passengers. The goal of this thesis is to develop and implement a control strategy which can enable lateral vehicle motion control with guarantee of safety and performance.

1.3 Layout of this thesis

The remainder of this thesis is structured as follows:

Chapter 2 explains and defines the problem this thesis is supposed to solve.

Chapter 3 introduces available research results related to this thesis and reviews relevant theories of Model Predictive Control (MPC).

Chapter 4 presents a MPC design and implementation to solve the problem defined in Chapter 2.

Chapter 5 presents and discusses the simulation results.

Chapter 6 makes an overall conclusion regarding the work done for this thesis and discusses potential future work.

2

Problem Formulation

In this chapter, we first present a desired path based coordinate system. The lateral motion of a vehicle is presented with respect to a desired path. Then, a bicycle model is introduced for the lateral and yaw dynamics and a state space model representing the vehicle dynamics is derived. System constraints including physical constraints and control requirements are presented accordingly. Based on the state space model and system constraints, a path-following problem is formulate.

2.1 Coordinate System

Figure 2.1 shows a desired path based coordinate system. A vehicle body fixed frame is used to represent the vehicle's lateral position and orientation with respect to the desired path:

1) Global frame and body-fixed frame: Denote the global frame as (X, Y). Denote the body fixed frame of the vehicle as (x, y), where x and y are longitudinal and lateral axises of the vehicle, and the point of origin of body-fixed frame is defined as vehicle center of gravity. Therefore, the orientation of the vehicle with respect to the global frame is the orientation of the body-fixed frame w.r.t the global frame, which is denoted as Ψ in Figure 2.1.

2) Desired Position and Orientation: Define the intersection point of y-Axis and the desired path as desired position P. Define a frame (x_p, y_p) attached to the point P, where y_p is tangent to the desire path and x_p is vertical to y_p . The orientation of the frame (x_p, y_p) is the desired orientation of the vehicle, which is denoted as Ψ_{des} in Figure 2.1.

3) Relative Position and Orientation: Define Δy as position distance measured along vehicle lateral axis y, and Define $\Delta\Psi$ as $\Psi - \Psi_{des}$. Therefore, Δy and $\Delta\Psi$ are vehicle's position and orientation, respectively, w.r.t. the desired path.

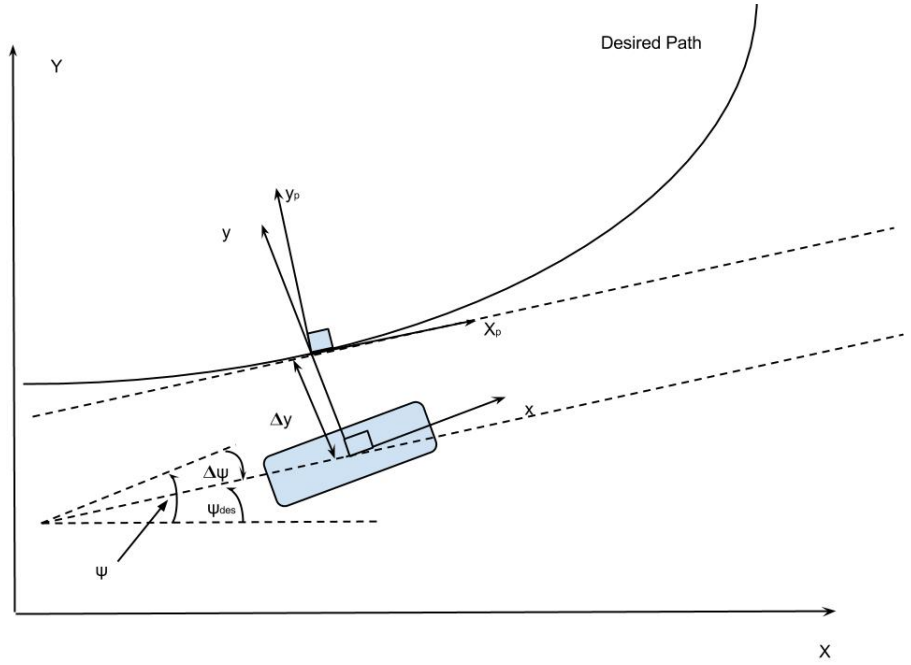


Figure 2.1: Vehicle in a desired path based coordinate system

2.2 Lateral and Yaw Dynamics

Figure 2.2 shows the bicycle model used to simplify the lateral and yaw dynamics model [5]. In figure 2.2, (x, y) denote the body fixed frame of the vehicle. l_f and l_r represent the distances of the vehicle center of gravity from the front and rear axles, respectively. F_{yf} and F_{yr} denote lateral tyre forces at the front and rear axles, respectively.

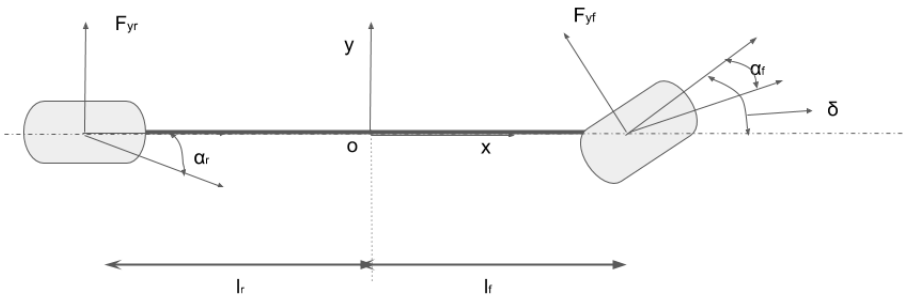


Figure 2.2: Bicycle Model

The vehicle motion w.r.t. the desired path in Figure 2.1 is subject to the lateral and yaw dynamics [5]:

$$m\dot{V}_y = -mV_x + 2[F_{yf} + F_{yr}] \tag{2.1a}$$

$$I_z\ddot{\Psi} = 2[l_f F_{yf} - l_r F_{yr}] \tag{2.1b}$$

$$\dot{\Delta}\Psi = \dot{\Psi}_{des} - \dot{\Psi} \tag{2.1c}$$

$$\dot{\Delta}y = -V_y + V_x\Delta\Psi \tag{2.1d}$$

$$\dot{\Psi}_{des} = V_x\gamma \tag{2.1e}$$

where m and I_z denote the mass and yaw inertia of the vehicle, respectively. V_x and V_y are the longitudinal and lateral velocities of the vehicle. The lateral tyre force F_{yf} and F_{yr} are, in general, nonlinear functions of the vehicle states. In this thesis, the lateral tire forces are calculated as:

$$F_{yi} = -C_i\alpha_i, i \in \{f, r\} \tag{2.2}$$

Where C_i are the tire cornering stiffness coefficients at the two axles and α_i are the tyre slip angles. For small slip angles: C_i can be approximated as,

$$\alpha_f = \frac{V_y + l_f\dot{\Psi}}{V_x} - \delta,\ \alpha_r = \frac{V_y - l_r\dot{\Psi}}{V_x} \tag{2.3}$$

2.3 State Space Model

Combining eq.(2.1), eq.(2.2) and eq.(2.3), we can derive the equation of motion of a bicycle w.r.t. a desired path as,

$$\ddot{y} = -(\frac{C_f + C_r}{V_x m})\dot{y} + (-V_x - \frac{C_f l_f - C_r l_r}{V_x m})\dot{\Psi} + \frac{C_f}{m}\delta_{wheel} \tag{2.4a}$$

$$\ddot{\Psi} = -(\frac{C_f l_f - C_r l_r}{I_z V_x})\dot{y} - (\frac{C_f l_f^2 + C_r l_r^2}{I_z V_x})\dot{\Psi} + \frac{C_f l_f}{I_z}\delta_{wheel} \tag{2.4b}$$

$$\Delta\Psi = \Psi_{des} - \Psi \tag{2.4c}$$

$$\dot{\Delta}\Psi = \dot{\Psi}_{des} - \dot{\Psi} \tag{2.4d}$$

$$\dot{\Delta}y = -\dot{y} + V_x\Delta\Psi \tag{2.4e}$$

$$\dot{\Psi}_{des} = V_x\gamma \tag{2.4f}$$

where the parameters' definition are concluded in Table 2.1

Table 2.1: Parameters' Definition

Parameters	Description	Unit
m	Mass	kg
I_z	Yaw moment of inertia of vehicle	kg m^2
C_r	Effective cornering stiffness, rear tires	N/rad
C_f	Effective cornering stiffness, front tires	N/rad
l_r	Longitudinal distance from c.g. to rear axle	m
l_f	Longitudinal distance from c.g. to front axle	m
γ	Bounded curvature of desired path	1/m

In the defined project for this thesis, steering angle rate is accessible command to the steering actuator system. Therefore, steering rate is treated as input:

$$\dot{\delta}(t) = u(t) \tag{2.5}$$

Meanwhile, non-zero road curvature causes the vehicle to deviate from the desired path. Therefore, road curvature is defined as disturbance:

$$\gamma(t) = \omega(t) \tag{2.6}$$

Define the state vector as $x = [\Delta y \ \dot{y} \ \Delta\Psi \ \dot{\Psi} \ \delta_{wheel}]^T$, input as $u = \dot{\delta}_{wheel}$, and input disturbance as $\omega = \gamma$. Eq.(2.4) can be rewritten as,

$$\dot{x} = A_c(V_x)x + B_c(V_x)u + E_c(V_x)\omega \tag{2.7a}$$

where

$$A_c = \begin{bmatrix} 0 & -1 & V_x & 0 & 0 \\ 0 & -\frac{C_f+C_r}{V_x m} & 0 & -V_x - \frac{C_f l_f - C_r l_r}{V_x m} & \frac{C_f}{m} \\ 0 & 0 & 0 & -1 & 0 \\ 0 & -\frac{C_f l_f - C_r l_r}{I_z V_x} & 0 & -\frac{C_f l_f^2 + C_r l_r^2}{I_z V_x} & \frac{C_f l_f}{I_z} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.7b}$$

$$B_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{2.7c}$$

$$E_c = \begin{bmatrix} 0 & 0 & V_x & 0 & 0 \end{bmatrix}^T \tag{2.7d}$$

System (2.7) is a linear time invariant (LTI) system when V_x is a constant.

2.4 Physical Constraints

A set of physical and design constraints should be considered for states and inputs of state space model (2.7):

a) Steering angle rate, i.e. the control input of the system (2.7), has a physical limitation from steering actuators, which can be expressed as:

$$\dot{\delta}_{min} < \dot{\delta}_{wheel} < \dot{\delta}_{max} \tag{2.8}$$

where $\dot{\delta}_{min} = -\dot{\delta}_{max}$ and $\dot{\delta}_{max}$ is the maximal steering angle change rate of the steering actuator.

b) Steering angle is also subject to physical limitation of steering actuators,

$$\delta_{min} < \delta_{wheel} < \delta_{max} \tag{2.9}$$

where $\delta_{min} = -\delta_{max}$ and δ_{max} is the maximal steering angle of the steering actuator.

c) Curvature of desired path is bounded for this thesis project:

$$\gamma_{min} < \gamma < \gamma_{max} \tag{2.10}$$

where $\gamma_{max} = -\gamma_{min}$ and γ_{max} is the maximal curvature from the desired path. Fig. 2.3 show a desired path for the thesis project.

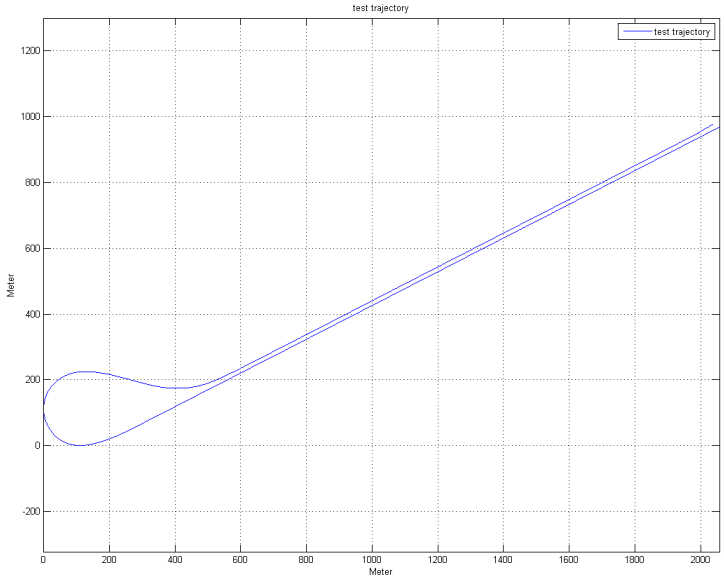


Figure 2.3: Trajectory of a vehicle driving along the test track

2.5 Control Requirements

Besides the constraints of the physical system, the lane keeping algorithm is supposed to fulfil safety requirements and guarantee comfort performance, which can be modelled as system constraints as well.

Safety Requirement: Although safety specification for the lateral control in highly automated drive is still on its early stage, we can see that guaranteeing bounded lateral deviation from the desired path is a fundamental safety requirement.

Lateral deviation bounds can be written as,

$$\Delta y_{min} < \Delta y < \Delta y_{max} \tag{2.11}$$

where $\Delta y_{min} = -\Delta y_{max}$. Δy_{min} and Δy_{max} should be chosen that eq.(2.11) will ensure that the vehicle is able to avoid potential hazards.

Comfort Requirements: Comfort requirements considered in this thesis are lateral velocity and lateral acceleration.

1) Lateral velocity bounds are defined as,

$$\dot{y}_{min} < \dot{y} < \dot{y}_{max} \tag{2.12}$$

where $\dot{y}_{min} = -\dot{y}_{max}$.

2) Requirement on lateral acceleration:
Lateral acceleration can be expressed as,

$$a_y = \dot{\Psi} V_x + \ddot{y} \tag{2.13}$$

Therefore, yaw rate are bounded for comfort purpose,

$$\dot{\Psi}_{min} < \dot{\Psi} < \dot{\Psi}_{max} \tag{2.14}$$

where $\dot{\Psi}_{min} = -\dot{\Psi}_{max}$.

2.6 Problem Statement

The objective of this thesis is solving the following problem:

Design a control algorithm controlling the vehicle such that $\Delta y \to 0$ and $\Delta\Psi \to 0$ asymptotically, while constraints (2.8), (2.9), (2.11), (2.12), (2.14) are satisfied for any curvature with given bounds in (2.10).

3

Literature Review on Model Predictive Control

The problem stated in Section 2.6 can be conveniently formulated as Model Predictive Control (MPC) [7]. Model Predictive Control is a control technology that can systematically take into account the system constraints at design phase. This feature makes MPC a suitable choice for solving the problem stated in Section 2.6.

The objective of this chapter is to overview the bases of Model Predictive Control and the available results related to this thesis work. Firstly, relevant researching work is reviewed. Followed by related work, formulation of standard MPC and a robust MPC are presented. Explicit MPC is introduced as it moves heavy computational burden offline and enable easier deployment of MPC. In particular, persistent feasibility and stability of MPC are discussed. The notations in this chapter are extracted from [7].

3.1 Related Work

MPC has been extensively used for vehicle control applications. In [8], the author used and compared full nonlinear model and online linearized model base MPC for active steering on icy road. In particular, a MPC controller is used to compute the front steering angle in order to follow the trajectory on slippery roads at the highest possible entry speed. [8] is a quite early example of application of MPC to autonomous driving for tracking purpose.

In [9], a robust nonlinear MPC for lane keeping and obstacle avoidance of autonomous and semi-autonomous vehicles is designed. The MPC framework is based on the "Tube-MPC" approach. The framework presents a systematic way of enforcing robust constraint satisfaction under the presence of model mismatch and disturbances during the MPC design stage. A robust invariant set is computed for the error system to bound the maximum deviation of the actual states from the nominal states under a linear control action.

3.2 MPC Formulation

3.2.1 MPC problem

Consider the problem of regulating states to the origin. Denote the state of the discrete-time linear time-invariant system as,

$$x(t+1) = Ax(t) + Bu(t), \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ are the state and input vectors, respectively, subject to the constraints,

$$x(t) \in \mathcal{X},\ u(t) \in \mathcal{U},\ \forall t \geq 0 \tag{3.2}$$

where the sets $\mathcal{X} \subseteq \mathbb{R}^n, \mathcal{U} \subseteq \mathbb{R}^m$ are polytopes containing the origin in their interior.

Assume that the measurements or estimation of the states $x(t)$ is available at the current time t, and the finite time optimal control problem:

$$\begin{aligned} J_t^* = \min_{U_{t\to t+N|t}} \quad & J_t(x(t), U_{t\to t+N|t}) = p(x_{t+N|t}) + \sum_{k=0}^{N-1} q(x_{t+k|t}, u_{t+k|t}) \\ \text{subj. to} \quad & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t}, k = 0, ..., N-1 \\ & x_{t+k|t} \in \mathcal{X}, x_{t+k|t} \in \mathcal{U}, k = 0, ..., N-1 \\ & x_{t+N|t} \in \mathcal{X}_f, \\ & x_{t|t} = x(t) \end{aligned} \tag{3.3}$$

is solved at time t, where $U_{t\to t+N|t} = u_{t|t}, ..., u_{t+N-1|t}$ and $x_{t+k|t}$ denotes the state vector at time $t+k$ predicted at time t.

The predictions are obtained by starting from the current state $x(t)$ and applying the input sequence $u_{t|t}, ..., u_{t+N-1|t}$ to the system $x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t}$.

Let $U^*_{t\to t+N-1|t} = \{u^*_{t|t} u^*_{t+N-1|t}\}$ be the solution of the problem in (3.3) at time t and $J_t^*(x(t))$ be the corresponding value of cost function. Then select the first element of $U^*_{t\to t+N-1|t}$ as the input of the system (3.1). Repeat solving the optimization problem (3.3) and applying the first element of $U^*_{t\to t+N-1|t}$ at every time instance, then a MPC controller is obtained. This strategy is also known as receding horizon control where control problem is solved over a future horizon (see Fig.3.1).

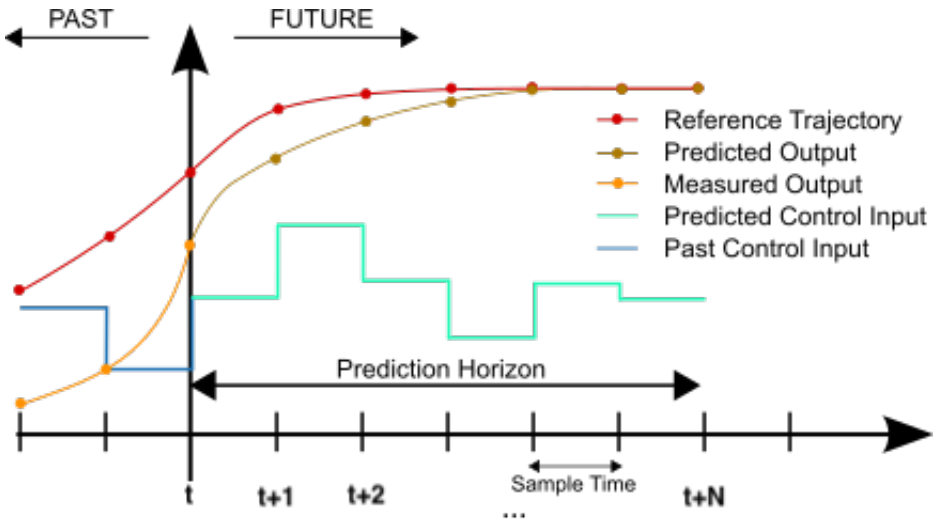


Figure 3.1: The receding horizon concept of MPC [6]

Consider optimization problem (3.3). If the cost function is quadratic and time invariant, the solution is a time invariant function of the initial state $x_0 = x(t)$. Denote the input vector $u_0, \cdots u_{N-1}$ as U, then the notations could be simplified as,

$$\begin{aligned} J^* = \min_U \quad & J(U) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{subj. to} \quad & x_{k+1} = Ax_k + Bu_k, k = 0, ..., N-1 \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U}, k = 0, ..., N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(t) \end{aligned} \tag{3.4a}$$

The MPC control law is,

$$u(t) = u_0^*(x(t)) \tag{3.4b}$$

where $u_0^*(x(t))$ is the first element of $U^*(x(t))$.

3.2.2 Explicit Solution to MPC

The optimal control problem (3.4a) can be solved with an online solver. An online solver adds much computation burden to the embedded system equipped in a car and makes MPC prohibitive for automotive application. An explicit solution to (3.4a) allows one to perform the heavy computation off-line [15], which is preferred. By substituting $x_k = A^k x_+ \sum_{j=0}^{k-1} A^j B u_{k-1-j}$ as vector of parameter, the optimization problem(3.4a) can be re-written as a Multiparametric Quadratic Programming (mpQP) problem:

$$\begin{aligned} V^*(x_0) = & \frac{1}{2} x_0' Y x_0 + \min_U (\frac{1}{2} u' H u + x_0' F u) \\ \text{subj. to} \quad & Gu \leqslant W + Sx \end{aligned} \tag{3.5}$$

Theorem 3.1. *For x_0 inside a polyhedral set, over which the QP problem (3.5) is feasible, solution to (3.5) is piecewise affine and continuous.*

Since mpQP problem (3.5) is identical to the MPC problem (3.4a), we conveniently conclude that,

Theorem 3.2. *For x_0 inside a polyhedral set, over which the optimization problem (3.4a) is feasible, solution to the problem (3.4a) is piecewise affine and continuous.*

Thus, the MPC problem defined by (3.4a) can be calculated explicitly, and the control law (3.4b) can be written in form of:

$$u(x_0) = \begin{cases} F_1 x + g_1 \text{ if } H_1 x_0 \leqslant k_1 \\ \quad \dots \\ \quad \dots \\ F_M x + g_M \text{ if } H_M x_0 \leqslant k_M \end{cases} \tag{3.6}$$

3.3 Robust MPC with Bounded Input Disturbance

The nominal MPC assumes that the system to be controlled and the model used for the prediction and optimization are the same. However, in the real world, uncertainty introduced by model mismatch, nonlinearity and disturbance is unavoidable and severe. Robust MPC schemes have been proposed and surveyed in [12]. The author in [12] formulated general robust MPC problem and discuss different approaches for control synthesis. Here we introduce the problem definition of a robust MPC where system is subject to bounded input disturbance.

Consider the problem of regulating to the origin, the state of the discrete-time linear time-invariant system:

$$x(t+1) = Ax(t) + Bu(t) + \omega(t), \tag{3.7}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $\omega(t) \in \mathbb{R}^{n_\omega}$ are the state, input and additive disturbance vectors, respectively, subject to the constraints:

$$x(t) \in \mathcal{X},\ u(t) \in \mathcal{U},\ \omega(t) \in \mathcal{W},\ \forall t \geq 0 \tag{3.8}$$

where the sets $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^m$, $\mathcal{W} \subseteq \mathbb{R}^{n_\omega}$ are polyhedra.

Correspondingly, in order to formulate a MPC controller with tolerant to the additive noise, the problem (3.4a) will be re-formulated to satisfy any disturbance ω as,

$$\begin{aligned} J^* = \min_U \quad & J(U) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{subj. to} \quad & x_{k+1} = Ax_k + Bu_k + \omega_k, k = 0, ..., N-1 \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U}, \forall \omega_k \in \mathcal{W}, k = 0, ..., N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(t) \end{aligned} \tag{3.9}$$

3.4 Persistent Feasibility

A direct consequence of the limited prediction of the MPC controller (finite time horizon) is that the controller might drive the state of the system to a subspace of the state space where there is no solution. It means that feasibility at the current time does not necessarily imply feasibility for all future times. It is desirable to design a *persistent feasible* [7] MPC such that feasibility for all future times is guaranteed if the controller is feasible at time O.

3.4.1 Robust Control Invariant Set

Before present conditions for persistent feasibility, we introduce a few definitions on set invariance theory.

Consider a discrete system:

$$x_{k+1} = f(x_k, u_k, \omega_k), \tag{3.10}$$

where $x_k \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $\omega_k \in \mathbb{R}^{n_\omega}$ are the state, input and disturbance vectors, respectively, subject to the constraints:

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \omega_k \in \mathcal{W},\ \forall k \in \mathbb{Z}_{0+} \tag{3.11}$$

A robust control invariant set is a set of states for which there exists a control law such that the system (3.10) never violates constraints (3.11) for any admissible sequence of disturbances.

Definition 3.1. *(Robust Control Invariant (RCI) Set:)*

A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be a robust control invariant set for (3.10) subject to constraints in (3.11) if:

$$x_k \in \mathcal{C} \Rightarrow \exists u(t) \in \mathcal{U} : f(x_k, u_k, \omega_k) \subseteq \mathcal{C}, \forall \omega_k \in \mathcal{W},\ \forall k \in \mathbb{Z}_{0+} \tag{3.12}$$

Definition 3.2. *(Maximal Robust Control Invariant Set:)*

The set $\mathcal{C}^\infty$ is said to be maximal robust control invariant if it is a robust control invariant set and contains all the other robust control invariant sets in $\mathcal{X}$.

3.4.2 Persistent Feasible MPC

Theorem 3.3. *Consider the problem (3.9) with $N \geq 1$. By setting values of $\mathcal{X}_f$ as a robust control invariant set $\mathcal{C}$ for system (3.7) then persistent feasibility of the MPC law for all cost functions is enforced [7].*

According to the theorem above, problem (3.9) is persistent feasible if select $\mathcal{X}_f$ as a robust control invariant set $\mathcal{C}$. Furthermore, we can select $\mathcal{X}_f$ as the maximal robust control invariant set $\mathcal{C}^\infty$ to enforce the persistent feasibility of (3.9),

$$\begin{aligned} J_0^* = \min_{U_0} \quad & J_0(x_0, U_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{subj. to} \quad & x_{k+1} = Ax_k + Bu_k + \omega_k, k = 0, ..., N-1 \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U},\ \forall \omega_k \in \mathcal{W},\ k = 0, ..., N-1 \\ & x_N \in \mathcal{C}^\infty \\ & x_0 = x(t) \end{aligned} \tag{3.13}$$

3.4.3 Calculation of RCI

The calculation of maximal robust control invariant sets relies on the **Pre**-set operator.

Definition 3.3. *(Pre set operater* $\mathbf{Pre}_f(\mathcal{S}, \mathcal{W})$*)*

$$\mathbf{Pre}_f(\mathcal{S}, \mathcal{W}) \triangleq \{x \in \mathcal{X} :\ \exists u \in \mathcal{U}\ f(x, u, \omega) \subseteq \mathcal{S}, \forall \omega \in \mathcal{W}\}, \tag{3.14}$$

which computes the set of states of (3.10) that can be robustly driven to the target set $\mathcal{S} \in \mathbb{R}^n$ in one step.

The procedure to compute the maximal robust control invariant set for system (3.10) subject to constraints (3.11) based on the operator (3.14) is summarised by the following algorithm.

Algorithm 1 Computation of maximal robust control invariant set $\mathcal{C}^\infty$

INPUT: $\mathcal{X}$, $\mathcal{W}$, $f(x_k, u_k, \omega_k)$
OUTPUT: $\mathcal{C}^\infty$

1: **procedure** RobustInvariantSet
2: $\quad \Omega_0 \leftarrow \mathcal{X}$
3: *loop*:
4: $\quad \Omega_{k+1} \leftarrow \mathbf{Pre}(\Omega_k, \mathcal{W}) \cap \Omega_k$
5: $\quad$ **if** $\Omega_{k+1} = \Omega_k$ **then**
6: $\quad\quad \mathcal{C}^\infty \leftarrow \Omega_k$
7: $\quad$ **else**
8: $\quad\quad k \leftarrow k + 1$, **goto** *loop*.
$\quad$ **return** $\mathcal{C}^\infty$

Algorithm 1 utilizes **Pre**-set operator to calculate $k-$step backward reachable set Ω_k for the dynamic system subject to certain constraints. When Ω_k remains unchanged, i.e., invariant, the algorithm is terminated.

In order to achieve the termination of the Algorithm 1 within finite steps, termination criteria, i.e. step 5 of the algorithm needs to be modified [11]. Algorithm 2 presents one method of the modification of termination criteria.

Denote the standard $n-$dimensional volume of a polyhedron as $\mathbb{V}$. Denote an outer approximation of maximal robust control invariant set $\mathcal{C}^\infty$ as $\mathcal{C}^N$. Termination of the algorithm above could be executed at N_{th} step when $\mathbb{V}$ of $N-$step backward reachable set Ω_N approaches a horizontal asymptote which indicates the value of $\mathbb{V}$ of $\mathcal{C}^\infty$. The modified algorithm could be expressed as,

Algorithm 2 Computation of invariant set approximation $\mathcal{C}^N$

INPUT: $\mathcal{X}$, $\mathcal{W}$, $f(x_k, u_k, \omega_k)$
OUTPUT: $\mathcal{C}^N$

1: **procedure** RobustInvariantSetApproximation
2: $\quad\Omega_0 \leftarrow \mathcal{X}$
3: *loop*:
4: $\quad\Omega_{k+1} \leftarrow \mathbf{Pre}(\Omega_k, \mathcal{W}) \cap \Omega_k$
5: $\quad$**if** $\frac{\mathbb{V}_{k+1}-\mathbb{V}_k}{\mathbb{V}_k} < \epsilon$ **then**
6: $\quad\quad\mathcal{C}^N \leftarrow \Omega_k$
7: $\quad$**else**
8: $\quad\quad k \leftarrow k+1$, **goto** *loop*.
$\quad$**return** $\mathcal{C}^N$

where ϵ is a small ratio and used for the determination of whether $\mathbb{V}$ of Ω_k approaches $\mathbb{V}$ of $\mathcal{C}^\infty$. If $\mathbb{V}$ of Ω_k approaches $\mathbb{V}$ of $\mathcal{C}^\infty$, an outer approximation of robust control invariant set is calculated. Notice that the outer approximation of robust control invariant set may not be sufficient in practice sense. However, due to the high computation demand of set calculation and the time limitation of the thesis project, we would like to apply the outer approximation and explore the control schemes in following parts.

3.5 Stability

MPC does not, in general, guarantee stability of the closed-loop system. Stability must be enforced by properly formulating both nominal MPC problem (3.4a) and the robust feasible MPC problem (3.13). Stability can be analyzed by constructing Lyapunov function for the closed-loop system.

The value function is naturally a candidate Lyapunov function chosen for stability analysis of model predictive control [13]. The author in [13] gives sufficient conditions on the terminal cost and/or terminal constraints which guarantee stability of MPC in general case.

One method for MPC problem (3.4a) to achieve closed-loop stability is by proper choice of terminal cost $x^T P x$. If the matrix P is obtained by solving discrete Algebraic Riccatic Equation (ARE) (3.15) and the prediction horizon length is chosen properly, (3.4a) will yield a controller that is stabilizing.

$$P = A^T(P - PB(B^TPB + R)^{-1})B^TP)A + Q \tag{3.15}$$

However, this approach does not guarantee stability in presence of disturbance (e.g. 3.13). Author in [14] reviewed the previous researches regarding stability of MPC problem subject to uncertainty. A min-max scheme enforces closed-loop stability for robust MPC with bounded additive disturbance, and therefore can be used to guarantee stability of MPC formulated from (3.13). Due to the time and scope limit, this thesis will not explore min-max scheme. Only close-loop stability condition for the nominal system, i.e. the case where $\omega = 0$ in (3.13) is enforced. One potential method to enforce close-loop stability with our approach is to introduce road curvature γ as an extra state and $\dot{\gamma}$ as bounded input disturbance, which will not be covered by this thesis due to time limit.

4

Control Implementation

Considering that the goal of this thesis is to develop and implement a control strategy which can enable lateral vehicle motion control with guarantee of safety and performance, we would like to design and implement a MPC controller by rewriting problem stated in Section.2.6 into an MPC problem (3.13), which consequently achieves the goal of this thesis. Furthermore, to reduce the computational burden of real time embedded system, an explicit MPC controller is generated.

In this Chapter, we will first introduce the software tools used for control implementation in this thesis. Secondly, we step-by-step present mathematical expressions and code implementation of a MPC problem in form of (3.13).

4.1 Matlab and Multi-Parametric Toolbox

MATLAB is a powerful tool for numerical calculation. With its capability of matrix manipulations, it is very convenient to define state-space models for LTI system and design state-space control and estimation. The Multi-Parametric Toolbox (MPT) [16] is an open source, Matlab-based toolbox for parametric optimization, computational geometry and model predictive control.

The computational geometry functions of MPT are capable of calculating **Pre**-set defined in (3.14) and polytope's properties including volume $\mathbb{V}$. With **Pre**-set and polytope's volume $\mathbb{V}$ available, algorithm 2 for invariant set calculation can be realised. Meanwhile, MPT supports the c code generation for explicit MPC controller in form of (3.6) by solving mpQP problem (3.5). Thus, Matlab and MPT are chosen as implementation and code generation tool for this thesis.

4.2 Controller Implementation

In order to implement a controller by solving the robust and persistent feasible MPC problem (3.13), the discrete state space model (3.7), polytopal constraints (3.8) and maximal RCI Set $\mathcal{C}^\infty$ defined in Definition.3.2 need to be pre-calculated.

4.2.1 Discrete State Space Model

Denote T_s as the sampling time, and let $x(k) = x(t)$. Assume V_x is constant, system (2.7) is turned into a linear time invariant (LTI) system. By applying zero-order

hold (ZOH) to the obtained LTI system, the discrete model can be present as in form of (3.7). Matlab function *c2d* can be conveniently used for discretization,

Matlab Code 4.1: Model discretization

```
% system model x(k+1) = A*x(k) + B*u(k) +B_r*r(k) =  A*x
    (k) + [B B_r]*[u(k) r(k)]^T
sysd = c2d(sysc, sampleTime, 'zoh');
```

Denote B_γ as the discretized E_c in eq.(2.7d) with ZOH method, then $\omega = B_\gamma \gamma \in \mathbb{R}$ for system (3.7).

4.2.2 Polytopal Constraints

Physical constraints (2.8, 2.9, 2.10), and control requirements (2.11, 2.12, 2.14), can be rewritten in form of (3.8), where

$$\mathcal{X} = \left\{ x = \begin{bmatrix} \Delta y \\ \dot{y} \\ \Delta\Psi \\ \dot{\Psi} \\ \delta_{wheel} \end{bmatrix} \in \mathbb{R}^5 : \begin{array}{c} \Delta y_{min} < \Delta y < \Delta y_{max} \\ \dot{y}_{min} < \dot{y} < \dot{y}_{max} \\ \Delta\Psi_{min} < \Delta\Psi < \Delta\Psi_{max} \\ \dot{\Psi}_{min} < \dot{\Psi} < \dot{\Psi}_{max} \\ \delta_{wheel_{min}} < \delta_{wheel} < \delta_{wheel_{max}} \end{array} \right\} \tag{4.1a}$$

$$\mathcal{U} = \{u = \dot{\delta}_{wheel} \in \mathbb{R} : \dot{\delta}_{wheel_{min}} < u < \dot{\delta}_{wheel_{max}}\} \tag{4.1b}$$

$$\mathcal{W} = \{\omega = B_\gamma \gamma \in \mathbb{R} : B_\gamma \gamma_{min} < \omega < B_\gamma \gamma_{max}\} \tag{4.1c}$$

where B_γ is discretized from E_c in (2.7d) with ZOH method in Section 4.2.1.

In MPT Toolbox, polytopal constraints $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{W}$ are constructed as,

Matlab Code 4.2: $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{W}$

```
% system constraints
model.x.min  = ...
    [sys_bounds.state.delta_y_lower;...
    sys_bounds.state.dot_y_lower;...
    sys_bounds.state.delta_Psi_lower;...
    sys_bounds.state.dot_Psi_lower;...
    sys_bounds.state.steering_angle_lower];
model.x.max = ...
    [sys_bounds.state.delta_y_upper;...
    sys_bounds.state.dot_y_upper;...
    sys_bounds.state.delta_Psi_upper;...
    sys_bounds.state.dot_Psi_upper;...
    sys_bounds.state.steering_angle_upper];

```

```
model.u.min = sys_bounds.input.steering_rate_lower;
model.u.max = sys_bounds.input.steering_rate_upper;

oemga_upper = B_r*sys_bounds.disturbance.gamma_upper;
oemga_lower = B_r*sys_bounds.disturbance.gamma_lower;

% state constraints
X = Polyhedron('lb',model.x.min,'ub',model.x.max);
% input constraints
U = Polyhedron('lb',model.u.min,'ub',model.u.max);
% noise constraints
W = Polyhedron('lb',oemga_lower,'ub',oemga_upper);
```

4.2.3 Maximal RCI Set

In Algorithm 2, calulation of maxiaml RCI set relies on **Pre**-set. **Pre**-set function can be realized in MPT toolsbox,

Matlab Code 4.3: **Pre**-set function

```
    % backward reachable set
    R = model.reachableSet('X', S, 'U', U,...
        'direction', 'backward');

    % intersect with the state constraints
    P = R.intersect(X0).minHRep();
```

In Algorithm. 2, volume $\mathbb{V}$ of Ω_k is used to formulate termination criteria, i.e. step 5 of the algorithm. The termination criteria is implemented as,

Matlab Code 4.4: Termination criteria of Algorithm. 2

```
    % iretation variables for debug and algorithm
       termination purpose
    P.computeVRep();
    volume = P.volume;

    % volume_pre is volume from last iretation
    delta_volume_ratio = (volume_pre - volume)/volume_pre;
    if delta_volume_ratio  < threashold_volume_ratio
```

4.2.4 Explicit MPC

MPT can be used for simulation of MPC by solving MPC problem (3.13) online. Furthermore, with matrices Q, R selected, MPT has a built-in command to calculate P by solving algebraic Riccati equation (ARE), which enforces the asymptotic stability of the nominal MPC. It is also convenient to set calculated Maximal RCI

set approximation $\mathcal{C}^N$ as terminal constraint set $\mathcal{X}_f$ with MPT. The following code shows how MPC problem (3.13) is constructed in MPT with Q, R, P and $\mathcal{C}^N$ pre-calculated:

Matlab Code 4.5: MPC Formulation

```
% Controller design model x(k+1) = A*x(k) + B*u(k)
model = LTISystem('A',A,'B',B);
model.x.penalty = Q;
model.u.penalty = R;
% add terminal set and terminal penalty
model.x.with('terminalSet');
model.x.terminalSet = robustInvSet;
model.x.with('terminalPenalty');
% Terminal penalty P is calculated by solving algebraic
    Riccati equation (ARE)
model.x.terminalPenalty = model.LQRPenalty();

% N is prediction horizon
onlineController = MPCController(model, N);
```

where **robustInvSet** represents $\mathcal{C}^N$ in the code, and **LQRPenalty()** calculates P by solving algebraic Riccati equation (ARE).

Furthermore, MPT has the ability of solving Multiparametric Quadratic Programming (3.5), and generating explict controler in form of (3.6) in c code, with one line command,

Matlab Code 4.6: Generate Explicit MPC

```
%generate explict controler
explicitController = onlineController.toExplicit();
```

5

Simulation Results

The MPC controller generated from Chapter.4 is tested in simulations. In this chapter, numerical results of calculation of robust control invariant set and control simulation are presented.

5.1 Paramters and Constraints

Table 5.1: Parameters' Values

Paramters	Description	Value
m	Mass	2164 kg
I_z	Yaw moment of inertia of vehicle	4373 kg m^2
C_r	Effective cornering stiffness, rear tires	122380 N/rad
C_f	Effective cornering stiffness, front tires	150540 N/rad
l_r	Longitudinal distance from c.g. to rear axle	1.6456 m
l_f	Longitudinal distance from c.g. to front axle	1.3384 m

Table.5.1 lists the values of vehicle parameters used in model (2.7). Meanwhile, in this chapter, we consider the driving scenario where the vehicle is at constant speed $V_x = 50(km/h)$ and sampling time is $T_s = 0.025s$. Matrices A_c, B_c and E_c in model (2.7) can be constructed with the parameters mentioned above, and discrete state space model in form of (3.7) is generated by the code presented in Section.4.2.1.

$\mathcal{X}, \mathcal{U}$ and $\mathcal{W}$ defined in (4.1a) - (4.1c) are constructed with the following values:

$$\begin{cases} \Delta y_{max} = -\Delta y_{min} = 0.2(m) \\ \dot{y}_{max} = -\dot{y}_{min} = 0.4(m/s) \\ \Delta\Psi_{max} = -\Delta\Psi_{min} = 30(degree) = 0.5236(rad) \\ \dot{\Psi}_{max} = -\dot{\Psi}_{min} = 15(degree/s) = 0.2618(rad/s) \\ \delta_{wheel_{max}} = -\delta_{wheel_{min}} = 30(degree) = 0.526(rad) \end{cases} \tag{5.1a}$$

$$\dot{\delta}_{wheel_{max}} = -\dot{\delta}_{wheel_{min}} = 40(degree/s) = 0.6981(rad/s) \tag{5.1b}$$

$$\gamma_{max} = -\gamma_{min} = 0.012 \tag{5.1c}$$

5.2 Maximal RCI Set Calculation

The implementation of Algorithm 2 according to Section.4.2.3 is used for the calculation of robust control invariant set. Choose $\epsilon = 0.05$ for termination criteria. Fig.5.1 shows the evolution of volume, Chebyshev radius and Chebyshev center norm of Ω_k at the iterations of Algorithm 2.

As volume of Ω_k approaches to a horizontal asymptote as steady state, Chebyshev radius and Chebyshev center norm, respectively, approach to horizontal asymptotes. Therefore, we conclude that Ω_k is approaching to $\mathcal{C}^\infty$ and our algorithm is working as expected. The numerical calculation returns $\mathcal{C}^N = \Omega_N$, where $N = 11$. $\mathcal{C}^N$ is used as an approximation of maximal robust control invariant set $\mathcal{C}^\infty$.

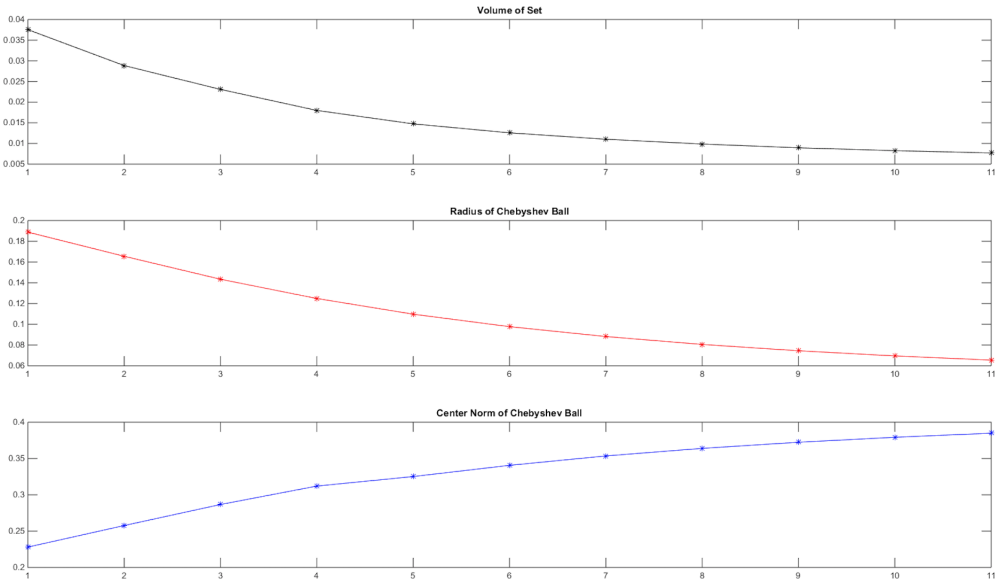


Figure 5.1: Evolution of volume, chebyshev radius and chebyshev center norm of Ω_k

5.3 Control Simulation

Consider the MPC implementation in Code 4.5.
Choose prediction horizon,

$$N = 3 \tag{5.2}$$

Choose Q, R matrix as:

$$Q = \begin{bmatrix} 19130 & 0 & 0 & 0 & 0 \\ 0 & 610 & -8440 & 0 & 0 \\ 0 & -8440 & 117240 & 0 & 0 \\ 0 & 0 & 0 & 0 & 176570 \end{bmatrix}, \quad R = 1 \tag{5.3}$$

Although P is calculated by solving algebraic Riccati equation (ARE) to enforce the asymptotic stability of the nominal MPC, simulation presented in following part shows that robust MPC is stable as well in our case.

Fig.5.2 shows the curvature of desired path in our simulation case. While the blue curve stands for the desired curvature, the red lines represent the boundaries of curvature.

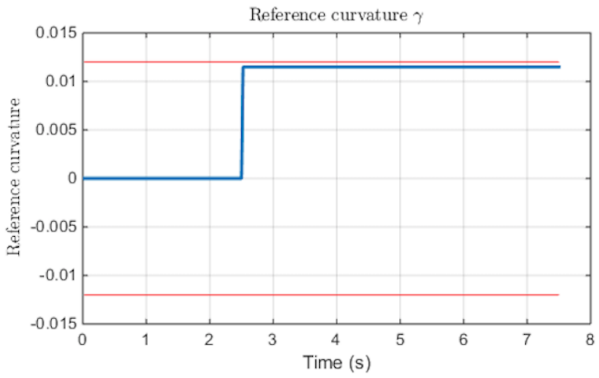


Figure 5.2: Control Simulation: Desired Curvature γ

Fig.5.3 presents the simulation result of generated controller with curvature defined in Fig.5.2.

Fig.5.3a and 5.3b show, respectively, steering angle rate $\dot{\delta}$ as input and steering angle δ. Both $\dot{\delta}$ and δ are within their own boundaries presented by the red lines in the Figures. Boundaries for $\dot{\delta}$ and δ are defined by (2.8) and (2.9) respectively.

In particular, Fig.5.3c and Fig.5.3d show, respectively, the lateral deviation Δy and orientation error $\Delta\Psi$ with respect to the desired path defined in Fig.5.2. In Fig.5.3d, we can see that $\Delta\Psi$ reaches zero as steady state when non-zero curvature γ is present. In Fig.5.3c, Δy is not zero when non-zero curvature γ is present. Nevertheless, the constraints for Δy and $\Delta\Psi$ are both satisfied. As the constraints for Δy are fulfilled, safety control requirement (2.11) is satisfied.

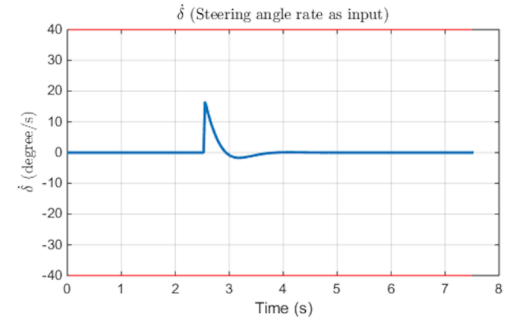
Fig.5.3e and 5.3f are, respectively, $\dot{y}$ and $\dot{\Psi}$. As both $\dot{y}$ and $\dot{\Psi}$ are within red boundaries, comfort requirements (2.12) and (2.14) are guaranteed.

To sum up, the simulation shows that, at speed $V = 50km/h$, the implemented controller is able to steer the vehicle such that $\Delta y \to 0$ and $\Delta\Psi \to 0$ asymptotically. Further more, constraints (2.8), (2.9), (2.11), (2.12), (2.14) are satisfied for the curvature with given bounds in (2.10).

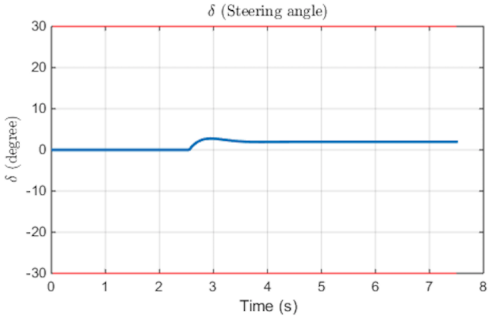
Recall the problem statement in section 2.6. As additional simulations show that the design method in this thesis generates controllers at different speeds are able to control the vehicle to follow the desired path while satisfying the physical constraints and control requirements derived from safety and comfort needs. We can conclude that the goal of the thesis is achieved.

However, in most simulation cases, the boundaries are not hit by the input or states. In other words, the constraints (4.1a), (4.1b) and (4.1c) are not active for most test

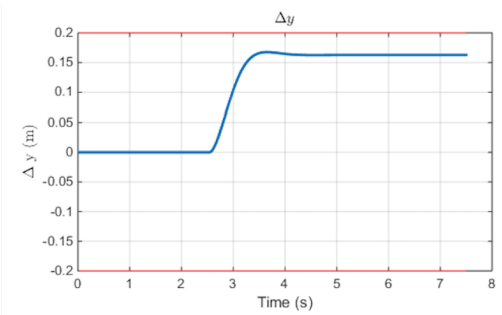
cases. With limited computational power for the calculation of RCI set during this thesis work, we were not able to explore enough test cases with different tuning, vehicle parameters and control requirements. Therefore, one of potential future work could be to re-formulate the requirements or models such that the sets (4.1a), (4.1b) and (4.1c) are activated more often during tests.



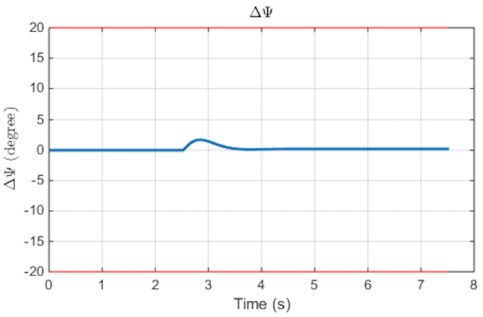
(a) Control Input $\dot{\delta}$



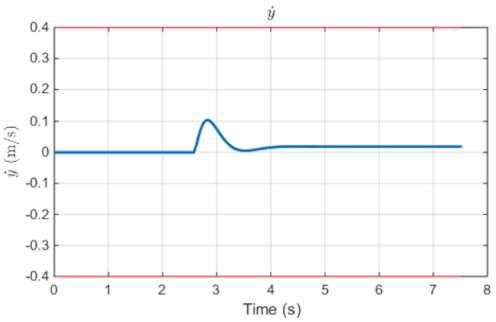
(b) Steering Angle δ



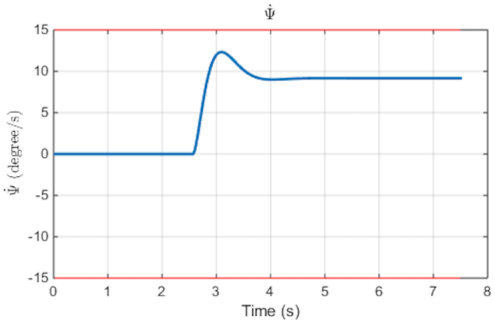
(c) Lateral Deviation Δy



(d) Orientation Error $\Delta\Psi$



(e) $\dot{y}$



(f) $\dot{\Psi}$

Figure 5.3: Control Simulation: Input and States (The red lines represent the boundaries of each variable)

6
Conclusion

Within the work of this thesis, we designed and implemented a robust MPC for lateral vehicle motion control. The safety and comfort performance requirements are formulated as constraints, which are enforced under MPC problem formulation. Relevant topics for MPC are discussed and explored. In particular, persistent feasibility and explicit solution to MPC are discussed and achieved. Preliminary simulation results of the implemented controller are presented. The controller is able to minimise the lateral deviation w.r.t. a desired path and guarantee safety and performance requirements.

Meanwhile, as discussed through the thesis, there are a few research gaps remain to be explored. And the exploration of these gaps are naturally potential future work:

1) As discussed in Section 3.4.3, the applied Algortihm 2 does not necessarily give an inner approximation of maximal robust invariant set. Further termination conditions should be formulated such that of the Chebishev's ball Ω_{k+1} is enclosed in Ω_k to get an inner approximation of maximal RCI set.

2) Section 3.5 mentioned that by applying P obtained by solving discrete Algebraic Riccatic Equation (ARE) and proper prediction horizon length, nominal MPC will be stabilizing. However, the MPC scheme used in this thesis is a robust MPC instead of nominal MPC. Although the simulation results show that our controller is also stabilizing, the theory gap needs to be filled with more academic work.

3) Chapter 5 presented the simulation results of which the constraints sets are not active. Therefore, requirements or models must be re-formulated such that the constraints sets are activated in more driving scenarios. With constraints sets activated, the benefit of MPC as a constrained optimal control is maximized.

To sum up, with exploring MPC for lateral vehicle motion control, this thesis work built some basis for further investigation on lateral vehicle motion control with guarantee of safety and performance.

Bibliography

[1] Volvo Car XC90 Driver Alert Control. Retrieved August 14, 2015.

[2] European Commission Statistics – accidents data. Retrieved August 14, 2015.

[3] Volvo Car Group initiates world unique Swedish pilot project with self-driving cars on public roads. Retrieved September 08, 2015

[4] ISO/DIS 26262, ISO-26262 Road Vehicles - Functional Safety, International Organization for Standardization, 2011.

[5] Rajamani, R. (2011). Vehicle dynamics and control. Springer Science & Business Media.

[6] Model predictive control-WIKIPEDIA. Retrieved January 19, 2020.

[7] F. Borrelli, A. Bemporad, M. Morari (2015). Predictive Control for linear and hybrid systems. To be published .

[8] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., & Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. Control Systems Technology, IEEE Transactions on, 15(3), 566-580.

[9] Gao, Y. (2014). Model Predictive Control for Autonomous and Semiautonomous Vehicles (Doctoral dissertation, University of California, Berkeley).

[10] Di Cairano, S., & Borrelli, F. (2013, December). Constrained tracking with guaranteed error bounds. In Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on (pp. 3800-3805). IEEE.

[11] Kolmanovsky, I., & Gilbert, E. G. (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. Mathematical problems in engineering, 4(4), 317-367.

[12] Bemporad, A., & Morari, M. (1999). Robust model predictive control: A survey. In Robustness in identification and control (pp. 207-226). Springer London.

[13] Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. Automatica, 36(6), 789-814.

[14] Paulson, J. A., Streif, S., & Mesbah, A. (2014). Stability for Receding-horizon Stochastic Model Predictive Control. arXiv preprint arXiv:1410.5083.

[15] Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. Automatica, 38(1), 3-20.

[16] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In Proc. of the European Control Conference, pages 502–510, Zurich, Switzerland, July 17–19 2013.